AUTOMATIC LAYOUT OF METRO MAPS USING MULTICRITERIA OPTIMIZATION

A THESIS SUBMITTED TO

THE UNIVERSITY OF KENT

IN THE SUBJECT OF COMPUTER SCIENCE

FOR THE DEGREE

OF DOCTOR OF PHILOSOPHY.

Draft on July 30, 2007

By

Jonathan M. Stott

July 2007

Contents

List of Tables **viii**

List of Figures **xiii**

Dedication **xiv**

Abstract **xv**

Acknowledgements **xvi**

1 Introduction **1**

1.1 Metro Maps: A Brief History of Design 1

1.2 Motivation 4

1.3 Contributions of this Research 5

1.4 Thesis Outline 5

1.5 Publications 6

2 Background **8**

2.1 Characteristics of Metro Map Diagrams 8

2.1.1 Evolution of the Diagrammatic Design 10

2.1.2 Lines 14

2.1.3 Labelling 14

2.1.4 Topographic Metadata 17

2.1.5 Stations, Termini and Interchanges 20
2.2 Graphs and Graph Drawing 22
2.2.1 Force-Directed Graph Drawing 22
2.2.2 Orthogonal Graph Drawing 24
2.2.3 Graph Drawing Aesthetics 27
2.2.4 Optimisation Methods for Drawing Graphs 28
2.3 Schematic Diagrams 29
2.4 Cartography 31
2.4.1 Cartographic Generalisation 31
2.5 Labelling 31
2.6 Applications of the Metro Map Metaphor 32
2.7 Existing Research 33
2.7.1 Force Directed Metro Map Layout 33
2.7.2 Mixed-Integer Program for Metro Map Layout 35
2.7.3 Path Simplification for Metro Map Layout 35
2.7.4 Centrality-based Scaling 35

3 Metro Map Layout **36**
3.1 Aims 36
3.2 Definitions 37
3.3 Hill Climbing Multicriteria Optimisation 38
3.4 Initial Embedding 39
3.5 Node Movement 41
3.6 Node Movement Criteria 43
3.6.1 Angular Resolution Criterion, c_1 43
3.6.2 Balanced Edge Length Criterion, c_2 45
3.6.3 Edge Crossings Criterion, c_3 47
3.6.4 Edge Length Criterion, c_4 47

3.6.5 Line Straightness Criterion, c_5 50
3.6.6 Octilinearity Criterion, c_6 51
3.6.7 Criteria Weightings 52
3.7 Node Movement Rules 55
3.7.1 Restrict Movement to Boundary of Drawing Area 55
3.7.2 Enforcement of Geographic Relationships 56
3.7.3 Handling Node and Edge Occlusions 57
3.7.4 Preservation of Edge Ordering 59
3.8 Moving Nodes 60
3.8.1 Cooling 61
3.9 Labelling 62
3.9.1 Labelling Criteria 63
3.9.2 Label Occlusion Criteria 64
3.9.3 Label Position Criterion 65
3.9.4 Label Position Consistency Criterion 67
3.9.5 Node Proximity Criterion 68
3.9.6 Perpendicular Tick Criterion 69
3.9.7 Label Criteria Weightings 71

4 Clustering and Partitioning **73**
4.1 Clustering Overlength Edges 74
4.1.1 Why Cluster Overlength Edges? 74
4.1.2 Identifying Clusters Based On Overlength Edges 76
4.1.3 Multiple Overlength Edges 78
4.2 Clustering Non-straight Lines 80
4.2.1 Why Cluster Non-straight Lines? 80
4.2.2 Identifying Clusters Based on Non-straight Lines 82
4.3 Partitioning 82

4.3.1 Finding Faces 85
4.3.2 Deriving the Dual Graph 86
4.3.3 Diminishing the Dual Graph 86
4.3.4 Finding Partitions 86

5 Results and Discussion 96

6 Empirical Evaluation 97

6.1 Introduction 97
6.2 Hypotheses 97
6.3 Methodology 104
6.4 Results 108
6.5 Analysis 108
6.6 Conclusion 108

7 Applications of the Metro Map Paradigm 109

7.1 Flexibility and Extensibility 109
7.2 Visualisation of Project Plans 109
7.2.1 Timescale Criterion 110
7.2.2 Initial Embedding 110
7.2.3 Implementation 111
7.3 Website Navigation Visualisation 112
7.4 Other Applications 118

8 Conclusion 119

8.1 Summary of Contributions 119
8.2 Further Work 119

Bibliography 120

A Examples of Method Output **131**

B Empirical Evaluation **138**

B.1 Questions and Maps Used 138

B.1.1 Atlanta 138

B.1.2 Bucharest 141

B.1.3 Mexico City 144

B.1.4 Stockholm 146

B.1.5 Toronto 149

B.1.6 Washington D.C. 151

B.2 Associated Material 153

B.2.1 Preliminary Script 153

B.2.2 Postliminary Script 154

B.2.3 Concluding Handout 155

B.3 Results 155

B.3.1 Route Planning Tasks 155

B.3.2 Design Preferences 196

B.3.3 Questionnaire Feedback 199

List of Tables

3.1 Examples of how the edge length criterion varies for different edge lengths 49
3.2 Examples of octilinearity criterion calculations 53
3.3 Typical node movement criteria weightings 54
3.4 Label Occlusions 65
3.5 Label Position Weightings 67
3.6 Typical label position criteria weightings 72
4.1 Breadth-first search of the graph in Figure 4.3 78
4.2 Breadth-first search of the graph in Figure 4.3 taking multiple overlength edges into account 79
6.1 Characteristics of metro maps used in the empirical evaluation . . 99
6.2 Number of subjects in each experiment group 104
B.1 Atlanta questions 138
B.2 Bucharest questions 141
B.3 Mexico City questions 144
B.4 Stockholm questions 146
B.5 Toronto questions 149
B.6 Washington questions 151
B.7 156
B.8 Design Preferences: Geographic Map 196

B.9 Design Preferences: Normalised Published Map 197
B.10 Design Preferences: Automatically-Drawn Map 198
B.11 Question 1 199
B.12 Question 2 200
B.13 Question 3 202
B.14 Question 4 204

List of Figures

1.1 London Underground map published in 1926. 2
1.2 London Underground map published in 1933. 3
1.3 Extract from the London Underground map published in 2005. . . 3
2.1 Extract of part of an 1829 map of the Liverpool to Manchester Railway 10
2.2 Map of London's Metropolitan Railway from 1874 11
2.3 Late nineteenth-century map of Berlin's railways 11
2.4 Map of London's Metropolitan Railway from 1896 12
2.5 Map of the London and North Eastern Railway Great Eastern suburban network from 1929 13
2.6 Examples of metro map lines on published maps 15
2.7 Examples of metro map labelling on published maps 18
2.8 Examples of metro map topographic metadata on published maps 19
2.9 Examples of metro maps using ticks to represent stations on published maps 20
2.10 Examples of devices used for distinguishing interchange stations on published metro maps 21
2.11 Examples of metro maps using different devices for representing stations on published maps 21
2.12 Randomised embedding of K_6 23

2.13 Embedding of K_6 using the basic spring embedder 23
2.14 An orthogonal graph 24
2.15 A planar orthogonal graph 25
2.16 Three route maps rendered to show the same route 30
2.17 Open source route map 33
3.1 Metro map features 37
3.2 Grid used for embedding the metro map graph 38
3.3 Moving nodes to grid intersections 42
3.4 Angular Resolution 43
3.5 Angular resolution calculation 44
3.6 Angular resolution calculation 44
3.7 Balanced edge lengths 45
3.8 Example used to illustrate the edge length criterion 49
3.9 Problem with diagonal edge lengths and the edge length criterion 50
3.10 Line Straightness Examples 51
3.11 Example of the calculation of the line straightness criterion 51
3.12 Example of the calculation of the octilinearity criterion 52
3.13 Simple example of node movement criteria for a complete graph . 54
3.14 Restricting node movements within a drawing area boundary . . . 56
3.15 Geographic Relationships 57
3.16 Node and Edge Occlusions 58
3.17 Preservation of edge ordering 59
3.18 The node movement radius 61
3.19 Label Positions 62
3.20 Splitting a long, many-word label over multiple lines 63
3.21 Illustration of the Label Occlusion Criteria 65
3.22 Label Position Consistency 68

3.23 Ambiguous Labelling 68

3.24 Label Tick Examples 69

3.25 Calculating the Perpendicular Tick Criterion 70

3.26 Simple example of label criteria for a complete graph 72

4.1 Clustering Overlength Edges 75

4.2 Example of clustering with a single overlength edge 77

4.3 Example of failed clustering with a single overlength edge 77

4.4 Clustering Multiple Overlength Edges 78

4.5 Example of failed clustering with overlength edges 80

4.6 Finding non-straight lines 81

4.7 Clustering nodes to find non-straight lines 83

4.8 Partitioning a graph into two 83

4.9 Finding Faces 85

4.10 Example of a planar graph, G, and the complete dual graph, G^* . 87

4.11 Example of a cycle in the dual graph equating to a cut in the planar graph represented by the dual graph 87

4.12 Finding the exit edge 90

4.13 Finding the exit edge in a triangular cycle 91

4.14 Non-terminating cutting algorithm 94

4.15 Modified cutting algorithm to remove the risk of an infinite loop in the cutting algorithm 95

6.1 Mexico City geographic map 100

6.2 Mexico City normalised published map 101

6.3 Mexico City automatically-drawn map 102

6.4 Mexico City *Mapa de la Red* official metro map 103

6.5 Screenshot of the software application used to conduct the empirical evaluation. 105

7.1 Example of a website that uses both breadcrumb trails and a hierarchical list of contents 112
7.2 Example of a website site map 113
7.3 A website navigation map using the metro map metaphor using a spring embedder to produce an initial embedding 115
7.4 A website navigation map using the metro map metaphor using a hierarchical layout algorithm to produce an initial embedding . . 117
A.1 Atlanta MARTA map. 132
A.2 Bucharest map. 133
A.3 Mexico City map. 134
A.4 Stockholm map. 135
A.5 Toronto map. 136
A.6 Washington D.C. map. 137
B.1 Atlanta MARTA geographic map. 139
B.2 Atlanta MARTA normalised published map. 139
B.3 Atlanta MARTA automatically-drawn map. 140
B.4 Bucharest geographic map. 142
B.5 Bucharest normalised published map. 142
B.6 Bucharest automatically-drawn map. 143
B.7 Mexico City geographic map. 144
B.8 Mexico City normalised published map. 145
B.9 Mexico City automatically-drawn map. 145
B.10 Stockholm geographic map. 147
B.11 Stockholm normalised published map. 147
B.12 Stockholm automatically-drawn map. 148
B.13 Toronto geographic map. 149
B.14 Toronto normalised published map. 150

B.15 Toronto automatically-drawn map. 150
B.16 Washington D.C. geographic map. 151
B.17 Washington D.C. normalised published map. 152
B.18 Washington D.C. automatically-drawn map. 152

Dedication

Dedication...

Abstract

Abstract goes here...

Acknowledgements

I would like to thank...

Chapter 1

Introduction

Metro maps [Ove03] can be seen worldwide and have possibly become the most memorised cartographic items in the world. Ever since construction of the first railways began in the early 19th century, there has been a need to map the networks. This is particularly the case with metro networks, where people need to plan short journeys across a city or metropolitan area. Before long, the metro networks were rapidly expanding and the traditional topographic maps were quickly becoming cluttered and difficult to read.

1.1 Metro Maps: A Brief History of Design

Early diagrammatic maps started to appear at the start of the 20th century. In 1900 a Dutchman by the name of Henrik Willem Mesdag produced a poster to advertise the ferry service between Harwich in England and the Hook of Holland [Wil47]. Early diagrammatic maps were usually seen in advertising and promotional posters where the aim of using a simplified map of the network might have been to emphasize its ease of use. Most diagrammatic maps around this time showed only small networks or sections of larger networks.

The London Underground diagram [TfL05], designed by Harry Beck and first

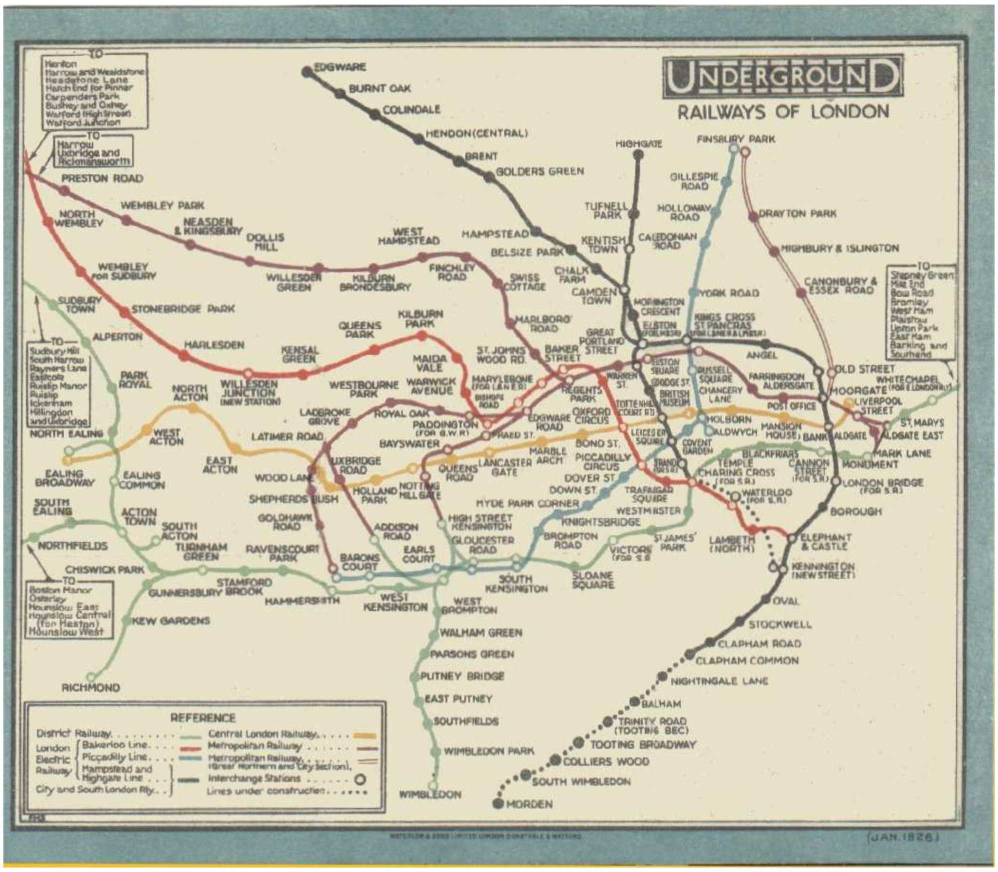


Figure 1.1: London Underground map published in 1926, designed by Fred Stingemore.

published in 1933 [Gar94], marked a significant departure from the more traditional geographic maps and built on the ideas introduced by earlier simple diagrammatic maps. Beck introduced more formalised design rules which have been followed right through to this day [Rob05]. Before 1933, a geographic map of the London Underground was used such as the example designed by Fred Stingemore and published in 1926 (Figure 1.1). Beck's first map of the London Underground was published in 1933 (Figure 1.2). As a testament to the strength of the design, modern London Underground maps still exhibit most of the design rules introduced by Beck, as shown in the 2005 map designed by Clockwork (Figure ??).

The diagram works by straightening meandering lines with line segments drawn either horizontally, vertically or diagonally at 45 and by using a non-linear scale

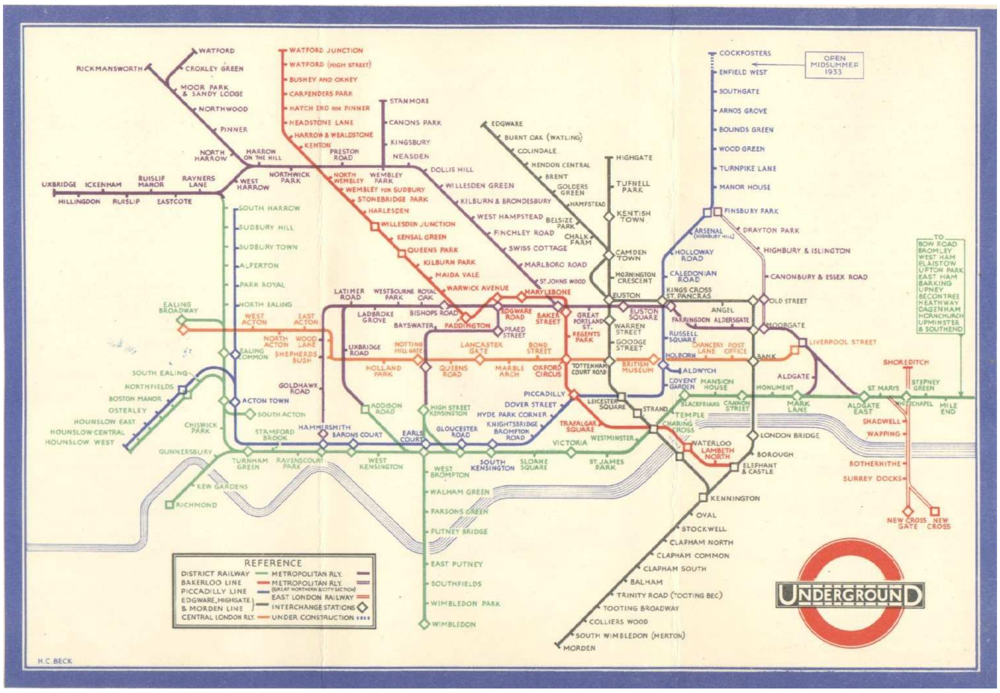


Figure 1.2: London Underground map published in 1933, designed by Harry Beck.

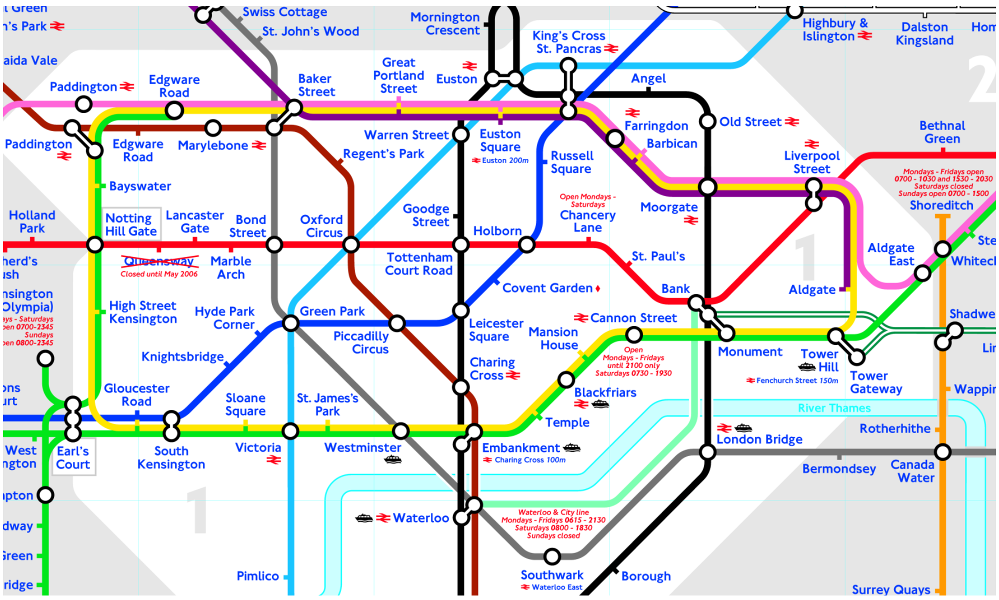


Figure 1.3: Extract from the London Underground map published in 2005 by Clockwork. Reproduced with permission from Transport for London.

so that the central area of the diagram is shown at a larger scale than the extremities. The effect is to produce a diagram that proved to be extremely clear and concise and has even been embraced as an iconic image of London. Following on from the success of the London Underground map, most metro networks now have a schematic map using a similar concept to. Schematic maps can be seen for other public transport networks such as over ground railways and bus routes. As a result of the prevalence of such maps. many people, particularly in the developed world, are familiar with them, and are confident in using them.

It might appear at first glance that metro maps are simple to design. However, this is almost always not the case and it can require a cartographer with lots of skill and design knowledge to be able to produce effective maps. There are many aspects which together contribute together to produce effective maps: the position of stations and the lines between stations, the width of lines; the symbols used to represent stations (circles, dots or ticks are most often used, but not exclusively so); the font and size of text used for labelling; the amount of geographic context and metadata to include (such as roads, rivers or coastlines); and the size, shape and resolution (for computer displays) of the medium being used to display the maps. Even what might seem to be trivial changes–slightly increasing the size of labels, for example–can drastically alter the overall appearance of a map. The challenge faced by metro map designers is to balance these issues so that the map is as easy to use as possible by people travelling on the metro system.

1.2 Motivation

★ What is our motivation for this research?

1.3 Contributions of this Research

The main contributions of this research are:

- 1
- 2
- 3
- 4

1.4 Thesis Outline

The structure of this thesis is as follows: [★ Update this section to reflect current structure]

Chapter 1 (this chapter) introduces the metro map as a tool for visualisation and outlines our motivation and contributions to research.

Chapter 2 provides detail on the areas that this research draws upon such as graph theory, graph drawing, cartography, map labelling and schematic diagrams.

Chapter 3 presents a detailed examination of closely related research including previous research on drawing metro maps.

Chapter 4 describes the process by which we draw metro maps using multi-criteria optimisation. It details the process involved in selecting optimum positions for nodes including the various criteria and rules that we have implemented.

Chapter 5 addresses the problem of moving many nodes in a graph at the same time by presenting manners in which clusters of nodes can be identified and also how the graph can be partitioned.

Chapter 6 presents our method for labelling the maps using another multicriteria optimiser.

Chapter 7 describes how we evaluated our results using an empirical experiment. The design, conduct and analysis of experimental results are detailed.

Chapter 8 provides a summary of applications for the metro map metaphor beyond public transport networks including the modifications to our method that would be required.

Chapter 9 summarises our conclusions and presents directions for further research.

Appendix A provides a detailed discussion of an example of our metro map drawing method.

Appendix B shows a number of maps drawn with our method as well as the criteria weightings required for each map.

Appendix C contains detailed information on our empirical experiment including the actual questions, the material presented to subjects and the complete set of experimental results.

1.5 Publications

Four publications have resulted from this research:

- J. M. Stott and P. Rodgers. Metro Map Layout Using Multicriteria Optimization. In *Proceedings 8th International Conference on Information Visualisation (IV04)*, pp. 355-362. IEEE, July 2004.

- R.A. Burkhard, M. Meier, P. Rodgers, M.T.J. Smis, and J. Stott. Knowledge visualization: A comparative study between project tube maps and gantt charts. In K. Tochtermann and H. Maurer, editors, *Proceedings of the 5th International Conference on Knowledge Management*, pp. 388-395. Know-Center, Austria, June 2005.

- J. M. Stott, P. Rodgers, R. Burkhard, M. Meier, and M. Smis. Automatic layout of project plans using a metro map metaphor. In *Proceedings of the 9th International Conference on Information Visualisation (IV05)*, pp. 203-206. IEEE Computer Society, July 2005.

- J. M. Stott and P. Rodgers. Automatic metro map design techniques. In *Proceedings of the 22nd International Cartographic Conference*, p. 10. International Cartographic Association, July 2005.

Chapter 2

Background

This chapter introduces the various background material which is relevant to my area of research. It begins with a discussion of the characteristics of metro maps, followed with a description of a number of different methods for graph drawing and laying out schematic diagrams. Aesthetics of graphs is very important, so a number of papers that consider graph aesthetics from an empirical point of view are described. Map labelling and the applications of the metro map metaphor to visualisation of abstract concepts are also covered.

2.1 Characteristics of Metro Map Diagrams

Metro map diagrams have been around almost as long as the networks themselves. Section 1.1 introduced some of the earliest examples of the metro map diagram as opposed to a map. The first task is to define the characteristics that distinguish a map from a diagram. The Oxford English Dictionary defines a *diagram* and *map* as follows:

> **diagram** An illustrative figure which, without representing the exact appearance of an object, gives an outline or general scheme of it,

so as to exhibit the shape and relations of its various parts.

map A drawing or other representation of the earth's surface or a part of it made on a flat surface, showing the distribution of physical or geographical features, with each point in the representation corresponding to an actual geographical position according to a fixed scale or projection.

So the purpose of the diagram is to remove the reliance on the actual geographic position of a point and also the need to rely on a fixed scale. This allows a much greater freedom to visually enhance the diagram in order to satsify particular requirements.

The requirements of metro map diagrams evolved during the early years of public transport systems in urban areas such as London. Some of the requirements would be altruistic and be intended to help the passenger solve route-planning tasks easily. A passenger might need a map for a number of different route planning tasks including "how do I get from station A to station B"; "how many stops is it until I have to get off"; "where do I have to change trains"; "what route do I need to take after I have changed trains"; "what are the alternative routes to get to my destination". Commercial pressure might also have dictated change or become a greater reason to change diagrams. For example, diagrams that appear clean and simple give the impression that a transport network is simple to use; straight lines give an impression of faster and more direct services than might actually be the case; distortion of scale and equal spacing of stations on a line can give the impression of shorter journeys in many cases (particularly from outlying regions of urban networks).

Throughout this thesis, the term *metro map* is used to indicate a diagrammatic map. The next section describes how the diagrammatic style for metro maps evolved.

Figure 2.1: Extract of part of an 1829 map of the Liverpool to Manchester Railway, drawn by G. Hennet [Hen29].

2.1.1 Evolution of the Diagrammatic Design

It is important to understand the evolution of the diagrammatic style for metro maps in order to appreciate the features which characterise the diagrams over equivalent maps. The railway revolution in Britain during the mid 19th century saw the construction of nearly 10,000km of new railways and in many cases the British Government's Board of Ordnance (the precursor of the Ordnance Survey) could not keep up with surveying the new tracks. For example, the very earliest railways were very often just etched onto existing geographic maps such as the example of the Liverpool to Manchester Railway shown in a map from 1829 (Figure 2.1) [Hen29]. Unfortunately, the burgeoning networks soon encountered a kind of map spaghetti and it became impossible to provide useful maps in this way. The very nature of densely populated areas implies a very dense railway network with stations much closer together than in more sparsely populated rural areas. Specialised maps soon appeared which did away with almost all surface topography, including one of London's Metropolitan Railway in 1874 (Figure 2.2) and one of Berlin's railways (Figure 2.3).

However, these maps didn't last very long in cities such as London, Berlin and New York as the networks continued to expand and even these "simpler" maps soon became cluttered. The need arose to make more space in the maps where

Figure 2.2: Map of London's Metropolitan Railway, published in 1874. This map illustrates an early example of the removal of much of the surface topography to produce a clearer map. [★ p8 of Metro Maps of the World]

Figure 2.3: Late nineteenth-century map of Berlin's railways. This map has all surface topology removed to enhance the clarity of the railway. [★ p8 of Metro Maps of the World]

Figure 2.4: Map of London's Metropolitan Railway, published in 1896. This map has large distortions in the scale—stations in the top-left corner of the map are up to six miles apart while some stations in the bottom-right corner are just a few hundred yards apart. [★ p9 of Metro Maps of the World]

needed by distorting the scale. This was also essential with maps covering both rural areas (where stations could be miles apart) and urban areas (where stations could be just a few hundred yards apart). If the map were to remain to scale and still be large enough for stations in the urban area to still be legible, the map would have to be excessively large. An early example of a diagram using distorted scale in this manner is that of the Metropolitan Railway, published in 1896 (Figure 2.4) where stations in the top-left corner of the map are actually much further apart than stations in the bottom-right corner. In a talk at the London Transport Museum on 6 May 2003, K. Garland, the author of *Mr. Beck's Underground Map*, described this selective distortion of scale the "convex effect" and suggests that Harry Beck was was the first to use this device in a large diagram of a complete network in his 1933 London Underground map: George Dow almost certainly predates this use in his 1929 LNER diagrams (Figure 2.5).

The next step in the evolution of the diagrammatic form of the metro map came with the introduction of the practice of simplifying lines by removing most

Figure 2.5: Map of the London and North Eastern Railway Great Eastern suburban network, published in 1929 and drawn by George Dow.

of the deviations a line makes and replacing it with a straight line on the diagram. This process is also known as shape generalisation. Early examples of maps which began to use this feature date back to the mid-1920s. This feature was adopted by George Dow in his London and North Eastern Railway diagrams of 1929 (Figure 2.5) and positively seized upon by Beck for his seminal 1933 London Underground digram (Figure 1.2). The great step made by both Dow and Beck was to make use of parallel lines and lines of common angle; Beck extended this concept to use lines entirely composed of horizontal, vertical and 45° diagonal components. Between 1929 and 1933, Dow was to produce a number of other "Dowagrams" of rail networks in south-east England, all conforming to the same design guidelines.

After Beck's 1933 London Underground map, many other maps appeared which followed similar forms. However, there was very little innovation and new ideas tended to be restricted to using different symbols for stations or other minor cosmetic tweaks. It is surely testament to the forms introduced by both Dow and Beck that we still use maps with the same features more than 70 years later.

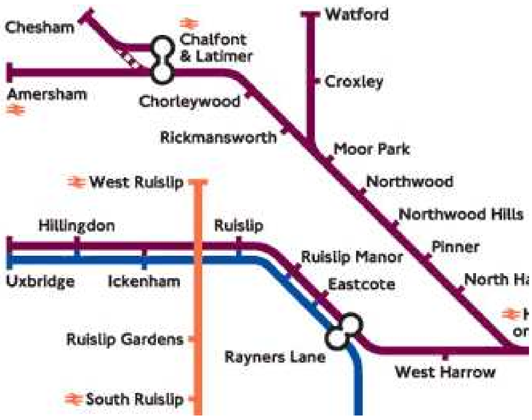
2.1.2 Lines

Depiction of lines on metro maps involves a number of characteristic features. Probably the most prominent feature of many metro maps is the frequent use of lines of common angle (angle generalisation). For example, the use of solely horizontal, vertical and 45° diagonals causes lines to be parallel with other lines. This tends to be very effective owing to the eight-fold symmetry of the horizontal, vertical and diagonal lines. The introduction of lines of common angle also forces the use of shape generalisation where the meandering path of a line has to be straightened to fit the common angles in use. Another prominent feature is the convex effect of increased scale at the centre of the map with decreasing scale towards the extremities of the map (scale generalisation) coupled with regular spacing of stations along a line. Lines are also usually distinguished on the map by using different colours. The colours of lines should normally be chosen such that lines that run together or intersect have great enough contrast in order to be able to tell them apart.

Examples of the features of metro map lines can be seen in six map excerpts in Figure 2.6. Examples (a), (b), (c) and (d) all show shape, scale and angle generalisation with regular spacing between stations and horizontal, vertical and 45° diagonal lines. Example (e) is similar except that it uses 35^circ diagonals instead of 45° diagonals. Example (f) is unusual in that it doesn't exhibit much use of angle or shape generalisation and the spacing between stations is much less regular than in the other examples. The last example is much closer to the geographic layout of the metro map than any of the other five examples.

2.1.3 Labelling

Labelling of station names is clearly an essential part of metro maps. It also introduces many challenges to make sure that labels are clear and unambiguous.



(a) London [TfL05]

(b) Berlin [BVG07]

(c) Munich [MVG07]

(d) Washington D.C. [WMA06]

(e) Madrid [MdM07]

(f) New York [MTA07]

Figure 2.6: Examples of metro map lines on published maps. Examples (a), (b), (c) and (d) illustrate horizontal, vertical and 45° diagonal lines with even spacing between stations. Example (e) shows a map that uses 35° diagonals and example (f) uses a geographic layout.

Many factors contribute to the way in which labels are applied to the maps and the labels also have a significant contribution to the way in which the maps are drawn. The font and size of lettering has a direct impact on the space required for each label: the size is usually dictated by the use that is intended for the map (for example, a large-print version of a map for visually impaired users or a version for a poster in a station would likely require larger font sizes). Most maps use sans serif fonts with proper case lettering[1].

A common feature of station labels is that they tend to be placed along a single side of a line when that line is straight for any length. A possible justification for this is that the labels then form a straight list along one side of the line, making it easier for the user to scan the labels to find the station that they want. Examples (a) and (c) in Figure 2.7 are good examples of this feature.

Station labels tend to be mostly horizontal[2], even to the point where a designer would prefer to use vertical or diagonal lines to avoid the need to use diagonal labels [Ros04]. Horizontal lines cause a problem when using horizontal labels as there usually isn't enough space for all the labels to be on the same side of the line (at least without making the line excessively long). Diagonal labels are quite often used for horizontal lines where they are needed, but occasionally, the labels will alternate from one side of the line to the other as in the labels for Ickenham, Hillingdon and Uxbridge in Figure 6(a). An example of diagonal labelling can be seen in Figure 7(f).

The biggest challenge when placing labels is to ensure that there is enough space for them without any cases where a label is drawn on top of (occludes) a line or a station. It is also desirable that the label is positioned so that it belongs unambiguously to just a single station. The example in Figure 8(f) shows a map

[1]Interestingly, proper case lettering on maps appeared quite late on, perhaps as late as 1936 in an LNER map.

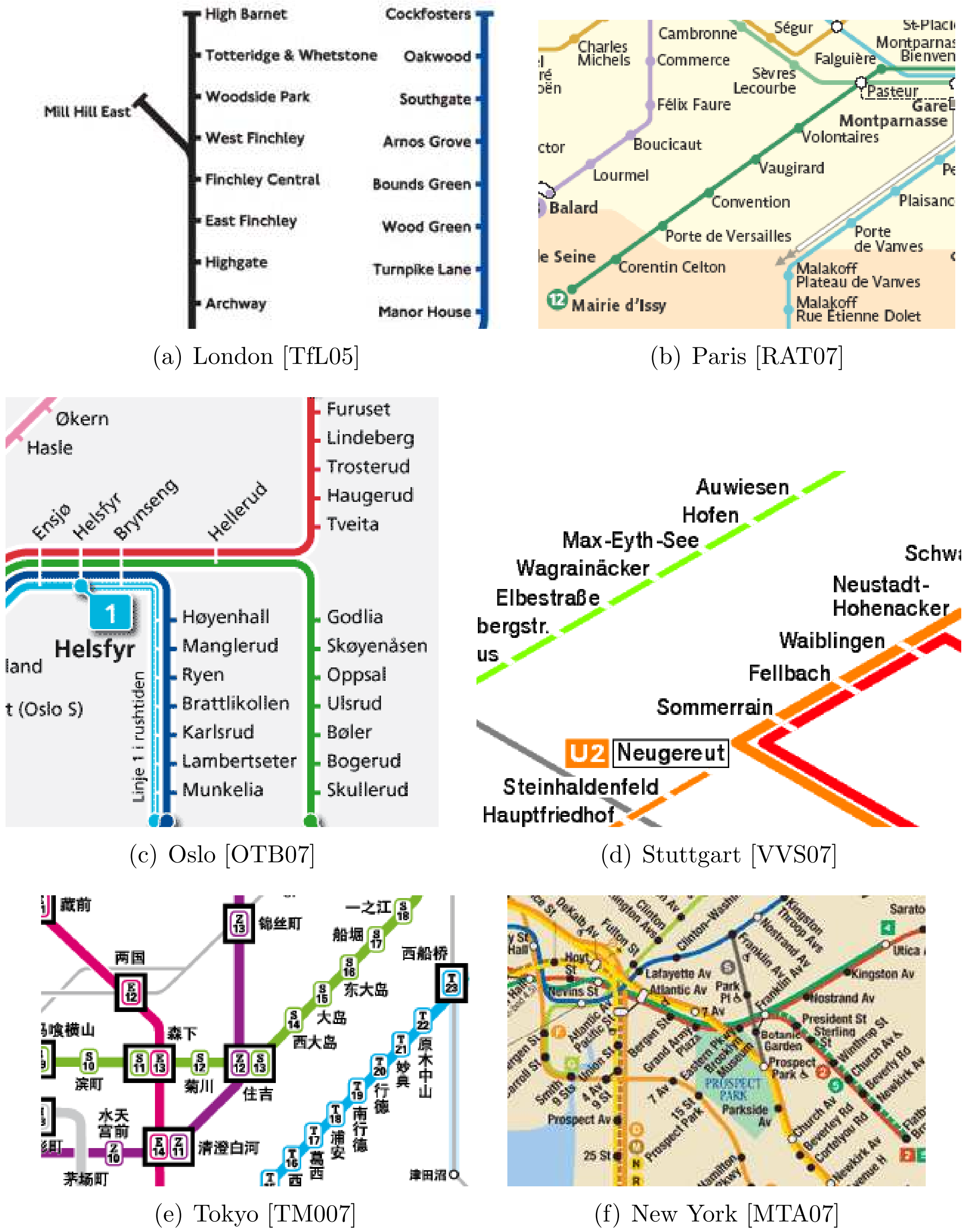
[2]The first map to use only horizontal station labels was probably a 1908 map of the District Line in London.

where labels have been allowed to be drawn on top of lines and some of the labels could even be claimed to be ambiguous.

2.1.4 Topographic Metadata

Topographic metadata are features such as roads, rivers, landmarks, parks and shorelines that do not serve any purpose for showing the interconnections of the public transport network. Having said this, metadata can perform the task of contexualising the transport network, for example to show which stations are on which side of a river that flows through a city. Topographic metadata is very commonly removed entirely or at least nearly completely from most published metro maps today and those maps that do feature metadata tend to be fairly of fairly small networks. The general trend is for less metadata to be present as the size and complexity of the transport network that the map is of increases. Examples of maps with no metadata at all and illustrated in Figure 2.8 include Madrid (a), Mexico City (b) and Stockholm (c).

Nearly all metro maps exhibit some form of topographic metadata, but in most cases it is usually not very prominent on the map. Coastal maps or maps of places with a major river nearly always show the river on the map (such as examples (d), (e) and (f) in Figure 2.8). In many cases the metadata is also styled to take on the characteristics of the map itself, including using the same lines of common angle and shape generalisation of coastlines. However, the inclusion of geographic metadata like coastlines and rivers can make the task of applying scale generalisation to the map difficult without distorting the metadata features beyond recognition.

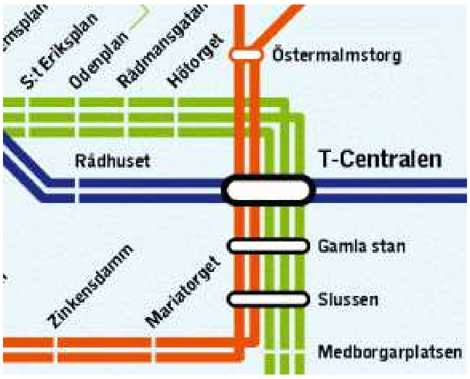


(a) London [TfL05]
(b) Paris [RAT07]
(c) Oslo [OTB07]
(d) Stuttgart [VVS07]
(e) Tokyo [TM007]
(f) New York [MTA07]

Figure 2.7: Examples of metro map labelling on published maps. Examples (a), (b), (c) and (d) all show horizontal station labels consistently on the same side of a line. Example (e) places information about the station inside the device used to represent the station and example (f) uses diagonal labels of various orientations.

(a) Madrid [MdM07]

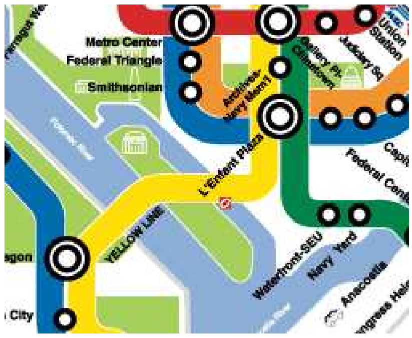
(b) Mexico City [Mex06]



(c) Stockholm [SL006]

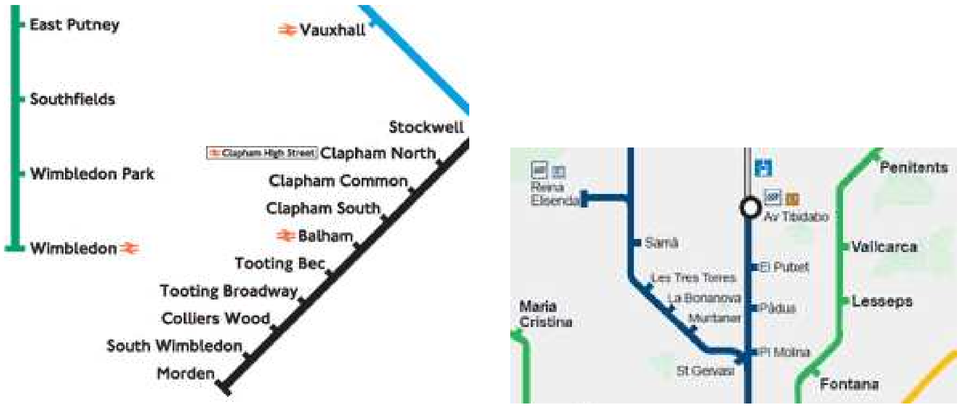
(d) Barcelona [TMB07]

(e) Boston [MBT07]



(f) Washington D.C. [WMA06]

Figure 2.8: Examples of metro map topographic metadata on published maps. Examples (a), (b) and (c) have no topographic metadata at all. Examples (d), (e) and (f) show examples of topographic metadata including coastline, parks, rivers and landmarks.

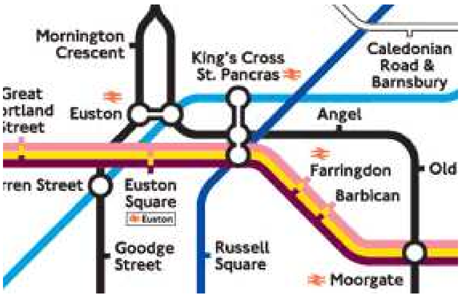


(a) London [TfL05] (b) Barcelona [TMB07]

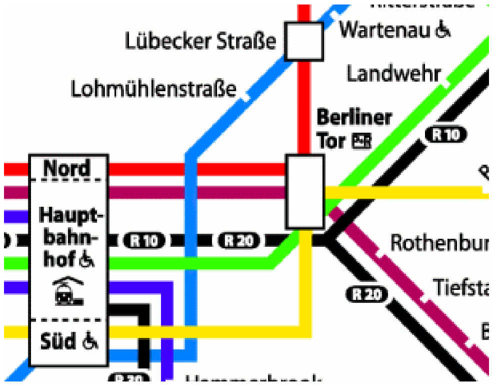
Figure 2.9: Examples of metro maps using ticks to represent stations on published maps.

2.1.5 Stations, Termini and Interchanges

The representation on a map of a station is also an important characteristic of metro maps. It does, however, have less of an impact on the overall layout of the map. There are a number of different ways of representing stations, often using devices such as dots, rings, ticks, squares and diamonds. In many cases, terminus stations and interchange stations are indicated using differently. In the examples from London (Figure 9(a)) and Barcelona (Figure 9(b)), ticks are used against the lines to indicate stations. Interchanges can very often be quite complex, such as those at Euston or King's Cross St. Pancras in London (Figure 10(a)) where the interchange is indicated with a ring. The rings are quite often bridged together which implicitly indicates a connection between lines. The example from Hamburg (Figure 10(b)) uses rectangles of differing sizes to join together lines where interchange is possible, most notably for the *Hauptbahnhof*. Examples of different devices used to represent stations include dots within the line (Lisbon, Figure 11(a)) and slices taken through the line (Stockholm, Figure 11(b)).



(a) London [TfL05]



(b) Hamburg [Hoc07]

Figure 2.10: Examples of devices used for distinguishing interchange stations on published metro maps.

(a) Lisbon [ML007]

(b) Stockholm [SL006]

Figure 2.11: Examples of metro maps using different devices for representing stations on published maps.

2.2 Graphs and Graph Drawing

[★ Introduction to graph drawing [dBETT94, dBETT99].]

Graph drawing is a generally difficult problem—many aspects are NP-complete [GJ83, MO85]. Therefore, acceptable heuristics are generally required to find good drawings of graphs.

2.2.1 Force-Directed Graph Drawing

Natural physical models appear to offer an excellent basis for a number of heuristics for the layout of graphs. Examples include a mechanical model of springs and electrical forces and a physical model based on simulated annealing. Section 2.7.1 introduces the previous work on the automatic layout of metro maps which uses a force directed approach.

The spring embedder [Ead84] uses a model of springs and electrical forces. In this case, nodes are represented as rings and edges as springs attached to the rings. The force of the spring causes connected nodes to attract each other (the force is calculated in terms of the logarithm of the distance between the nodes). A repulsive force is also applied using an inverse square law. This ensures that non-adjacent nodes are kept apart. The forces should allow for a reasonable separation of the nodes while edges are kept to be of roughly similar length.

The spring embedder typically starts with a random embedding of the graph and a number of iterations are applied until some equilibrium is reached. For larger graphs, a greater number of iterations is typically required. Figure 2.12 and Figure 2.13 show how the spring embedder lays out the complete graph with six nodes, K_6.

Fruchterman and Rheingold [FR94] extend Eades' algorithm by basing the force calculations on an optimal distance, k, which depends on the number of

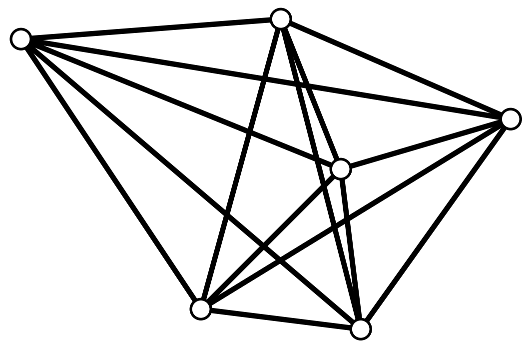


Figure 2.12: Randomised embedding of the complete graph with six nodes, K_6.

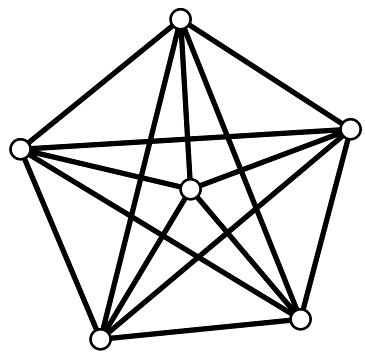


Figure 2.13: Embedding of K_6 using the basic spring embedder.

nodes and the area of the drawing area.

Kamada and Kawai show another approach to force directed graph drawing [KK89]. Their approach uses the relationship between the graph theoretic distance and the geometric distance between nodes to produce good embeddings. The algorithm works particularly well for symmetric graphs and is relatively good at minimising edge crossings. They use Floyd's shortest paths algorithm [Flo62] to find the minimum graph theoretical distances which runs in $O(n^3)$ time (faster shortest path algorithms are available); the other parts of their algorithm run in either linear or constant time.

There are many other force directed algorithms, including an algorithm based on magnetic springs [SM94, SM95], an adaptive approach (which is generally faster) [FLM95] and a method that ensures that edge crossings properties are preserved [Ber99].

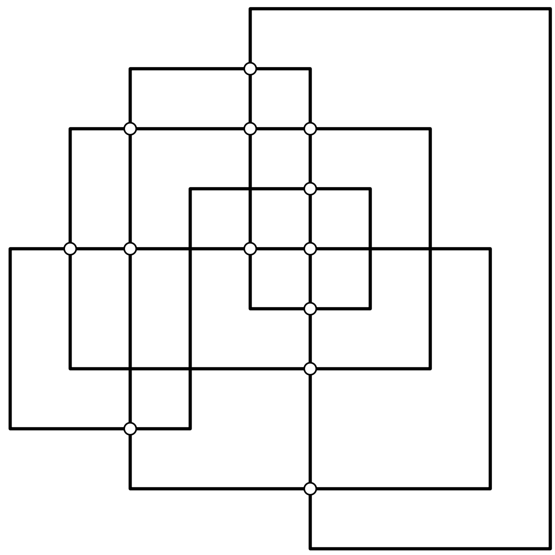


Figure 2.14: An orthogonal graph.

2.2.2 Orthogonal Graph Drawing

Orthogonal graph drawing is mainly concerned with drawing graphs where edges are restricted to horizontal and vertical lines. In many approaches, polylines (lines with more than one straight segment) are allowed where a line with a single segment cannot be drawn orthogonally (see Figure 2.14). As such, it is particularly interesting with regards to metro map layout where a predominantly orthogonal embedding is required. Orthogonal diagrams also have applications outside of graph drawing, particularly in the field of VLSI (very large scale integration) design and in diagrams used for information systems design such as entity relationship diagrams. Many orthogonal graph drawing algorithms are called dynamic—that is they construct an embedding from the bottom up rather than considering the whole graph in a top-down approach.

A number of approaches are based on the Kandinsky model which dictates that a graph should be drawn with nodes represented as finite-sized rectangles and edges should be drawn using only horizontal and vertical components. Tamassia [Tam87] presents an algorithm for embedding a planar graph on a grid in such

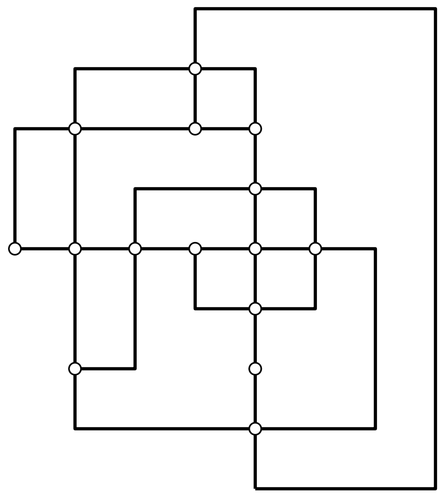


Figure 2.15: A planar orthogonal graph.

a way as to minimise the number of bends. A planar graph is a graph which can be drawn without edge crossings as shown in Figure 2.15. Highly connected graphs are not catered for—the maximum degree of a node for an orthogonal drawing using this algorithm is four. (The degree of a node is the number of edges which are incident to the node. For example, the degree of node 1 in Figure 2.14 above is 4.) The algorithm runs in $O(n^2 \log n)$ time where n is the number of nodes in the graph. The algorithm is based on network flow techniques where the flow is related to the number of bends on an edge and the capacity of nodes (how many more edges could be added incident to the node without exceeding the maximum of four incident edges). The aim of the algorithm is to find the minimum cost flow for the graph which should be that with the fewest edge bends. Later in the paper, an extension to k-gonal graphs is given. A k-gonal graph is one where edges are formed of polylines with segments at multiples of $180/k$ degrees. An orthogonal graph is therefore one where $k = 2$. This is a relatively simple extension of their algorithm and one which would prove useful for metro map layout as metro maps tend to be 4-gonal. (The maximum degree graph that Tamassia's algorithm handles is therefore equal to $2k$.)

Fößmeier and Kaufmann [FK95] extend Tamassia's algorithm to handle graphs with degree greater than $2k$. They do this be extending Tamassia's representation. They show an algorithm which first creates a nearly-orthogonal representation in $O(n^2 \log n)$ time. The nearly-orthogonal representation is then used to create an embedding on the grid using a similar compaction algorithm to Tamassia. To be able to draw high degree nodes, large nodes are represented by $8s$ smaller nodes (where s is the length of the side of the node). Larger nodes are preferred over increasing the number of bends (smaller nodes would be possible but only by increasing the number of bends in the graph).

Another algorithm is given by Papakostas and Tollis in [PT97]. This algorithm also considers graphs with nodes of degree greater than four, but is more general as it also takes into account simple non-planar graphs. They use three algorithms to size the nodes (taking into account the degree of the node), number and group the nodes into a particular order for processing and the placement of the nodes such that the increased size of the nodes is taken into account. Nodes are paired together so that edges between pairs of nodes can share a row or column. The algorithm is able to produce an embedding of a graph in $O(e)$ time where e is the number of edges in the graph. Maximum bounds on the size of the graph can be given in terms of the number of edges and there will not be more than e bends as each edge has at most one bend. However, the resulting embeddings tend to have a relatively high number of edge crossings and a planar embedding for a plane graph may not necessarily be found.

Brandes et al. [BEKW02] use a non-dynamic algorithm for drawing graphs where a previous embedding (possibly a rough sketch) is known. The Kandinsky model is again used. The graph is planarised by inserting dummy nodes at edge crossings. The use the concept of "readability"—the total number of bends in the graph—and "stability"—by how much the angles of the final embedding differ to the sketch. Their algorithm is claimed to run in $O(n^2 \log n)$ time which is very fast. They claim to maintain stability further by attaching nodes at the extremity of the graph to a bounding box surrounding the graph. After the algorithm has completed, the dummy nodes are removed and a compaction step is used to minimise the area of the graph. This paper is of particular interest as it particularly interested in embedding schematic diagrams.

An interesting paper by Six, Kakoulis and Tollis [SKT00] deals with post-processing of orthogonal embeddings of graphs. This is done by considering a number of cases where improvements could be made, such as by straightening out U-turns, removing superfluous bends, moving degree two nodes to improve their placement, removing self-crossings of edges, dealing with stranded nodes and reducing excess area. In general, significant reductions of the area of the graph, the number of edge bends and crossings and the length of edges are produced, resulting in more compact and aesthetically pleasing embeddings.

2.2.3 Graph Drawing Aesthetics

Many graph drawing algorithms make common assumptions that certain aesthetic criteria of the graph have a detrimental effect on the readability of the graph. These criteria include edge crossings, edge length, angular resolution of incident edges, proximity of one node or edge to another node or edge and node/edge occlusion [DM90]. Little work has been done in order to quantify and justify these aesthetic criteria.

The first attempt at an empirical study into graph aesthetics was by Purchase, Cohen and James [PCJ95]. In this work they considered the symmetry of the graph, edge crossings and edge bends, but do not consider other metrics such as node distribution or edge length. They conclude by saying that attempting to minimise edge crossings and edge bends makes a significant positive effect on the aesthetic quality of a graph layout. A later paper by Ware, Purchase, Colpoys and

McGill [WPCM02] contributes to this work by measuring the aesthetic quality of a graph based on the time taken for shortest paths to be found. They conclude that the continuation of edges (where edges pass through nodes as straight as possible) contributes to the quality of the graph when it is used for finding shortest paths.

2.2.4 Optimisation Methods for Drawing Graphs

[★ [SM94, SM95] ¡– why?]

A number of other graph drawing methods based on optimisation exist including a heuristic optimisation approach [Tun94], an approach using genetic algorithms [BBS97] and the use of simulated annealing [DH96]. Optimisation is a process whereby incremental improvements are made to a graph. Each successive iteration should produce a more optimal graph (in most cases—some algorithms occasionally allow less optimal graphs in order to avoid local minima).

Tunkelang [Tun94] takes the approach of creating an aesthetic cost function which is then minimized using a local optimisation procedure. The algorithm is flexible by nature of the way in which the aesthetic cost function is modular. The cost function used by Tunkelang is based on three aesthetic criteria, namely: uniform edge lengths, even distribution of nodes and a minimal number of edge crossings. A naïve implementation of the cost function runs in $O\left(e^2\right)$ time where e is the number of edges in the graph. This can be improved to a linear function by not recalculating the cost function from scratch each time. The local optimisation procedure involves examining the locality of a node to see whether a better position can be found. The algorithm is capable of producing acceptable embeddings for moderately sized graphs but struggles with large and dense graphs.

Simulated annealing is another approach to drawing graphs. It is based on physical processes of the way that liquids cool into a crystalline form (annealing).

This uses the analogy that the minimum energy state of the system (the crystal state) is equivalent to the minimum energy of a simulated annealing system. Davidson and Harel demonstrate an algorithm which is based on simulated annealing [DH96]. As with force directed drawing algorithms, they start with a random embedding of the graph and iteratively generate an improved embedding. The simulated annealing process produces an erratic improvement with some chance of subsequent iterations being apparently worse than the previous iteration. This has the advantage that it is possible to escape from local minima in the search space. Results are reasonably comparable to that of force directed methods, but the running time of the algorithm is generally poor, especially for large graphs (over sixty nodes).

2.3 Schematic Diagrams

There have been a few papers which are concerned with the schematisation of route maps. These normally involve a process of cartographic generalisation such that the essential information for following a route is preserved (such as the intersections and approximate direction) while exaggerating distances to make the schematic cleaner and easier to read (see Figure 2.16). This is particularly useful for route-planning applications and especially so with today's modern in-car satellite navigation devices where the aim is to communicate to the driver a route as quickly as possible.

Agrawala and Stolte describe the LineDrive system to generalise and schematise route maps using three types of generalisation [AS01]. The aspects of the map which are generalised are:

- Length of roads. Shorter roads are drawn longer and longer roads are drawn shorter such that routes involving roads with lengths that differ by several

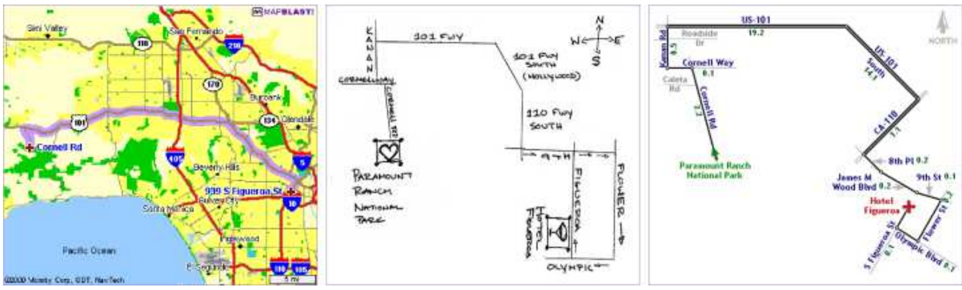


Figure 2.16: Three route maps rendered to show the same route [AS01]. The left-hand image uses a standard geographic map; the middle image is a schematic sketch while the right-hand image is a computer-generated schematic.

orders of magnitude can be drawn on a compact schematic map.

- Angle generalisation. Very acute angles at intersections are made more obtuse and roads are aligned with the horizontal or vertical axis.
- Shape generalisation. The exact meandering of a road is not important, so the shape of the road is straightened out.

Special care is taken to ensure that the topology of the map is preserved in the schematic, so that intersections, turn direction and the overall route shape are all maintained.

Another approach uses an algorithm for generalising the shapes of curves in a schematic diagram by eliminating unnecessary curves (in a similar way to the shape generalisation part used by Agrawala and Stolte) [BLR00]. Casakin et al. [CBKF01] provide a taxonomy of various aspects of schematisation of route maps (particularly intersections) and use their taxonomy to provide an empirical assessment of schematised graphs. Yates and Humphreys [YH98] give a discussion of various aspects of schematic diagrams and show a prototype (which uses a heuristic provided as a sample applet in the Java 1.1.6 SDK).

[★ Line generalisation [AV00, BCC+06, BCD+02, CC96, CD03, CDH+05, DP73, ET94, Goo95, GNS07, GHMS93, HS98, II86a, II86b, II88, MO88, Ney99, Tou85]]

★ intra-edge crossings [BNUW07].]

2.4 Cartography

★ Introduction to cartography. Explain why cartography is relevant to my research (kind of obvious really, but just to clarify why I have a section on it here!).

2.4.1 Cartographic Generalisation

★ Perhaps something about map generalisation. Reference to cartography research that uses generalisation to take a complex or highly detailed map and to simplify the map.

2.5 Labelling

[★ Section on labelling maps. Describe point-, line- and area-feature labelling but emphasize point-feature labelling as that is most relevant to my work.]

In many fields, such as cartography, labelling a map or diagram is an essential part of drawing the diagram. Typically, when labelling maps and diagrams, three types of labels are needed: node labels (which label point features, such as cities or stations on a railway), edge labels (which label such things as roads, railways or rivers) and area labels (for labelling oceans or countries).

Point feature label placement is discussed in detail by a survey by Christensen, Marks and Shieber [CMS95]. Typically, labels are placed in one of a finite number of positions (the labelling space) surrounding a point feature (or with a near infinite number of positions by sliding a label around the point feature as described in [vKSW98]). An order of preference is usually specified as to which positions a label should take up (for example, one could prefer a label to appear to the right of a point feature as opposed to the left).

A number of approaches exist in order to label point features such as performing an exhaustive search, using a greedy algorithm or using a discrete gradient descent algorithm (essentially an optimisation method).

Kakoulis and Tollis [KT98] show a method for labelling graphical features (of graphs). Their approach firstly reduces the search space for labels by seeing where a label would overlap another node or edge. The subset of potential label positions is then reduced further by detecting overlapping labels. Their results show that their algorithm produces a more efficient labelling for orthogonal diagrams than previous attempts. They also say that the size of labels is a significant factor in being able to produce a successful labelling of a graph.

2.6 Applications of the Metro Map Metaphor

[★ Section defining information and knowledge visualisation. Specifically, what are the differences. Illustrate the two types of visualisation with examples. Go on to explain knowledge visualisation in more detail, in particular areas which are relevant to my research.]

Other applications of the metro map metaphor are covered in more detail in Chapter 7.

[★ O'Reilly open source route map [ORe03] (Figure 2.17)]

[★ A subway map of cancer pathways [HW02] - also available at http://www.nature.com/nr

[★ Organising web-based learning resources [BGH02]]

The metro map is a good metaphor for visualisation of many other concepts. Sandvad et al. [SGSK01] use a metro map metaphor as a basis for the visualisation of non-geographic information. Nesbitt [Nes04] also uses the metaphor as a means for visualising abstract concepts such as the recommended reading of a thesis

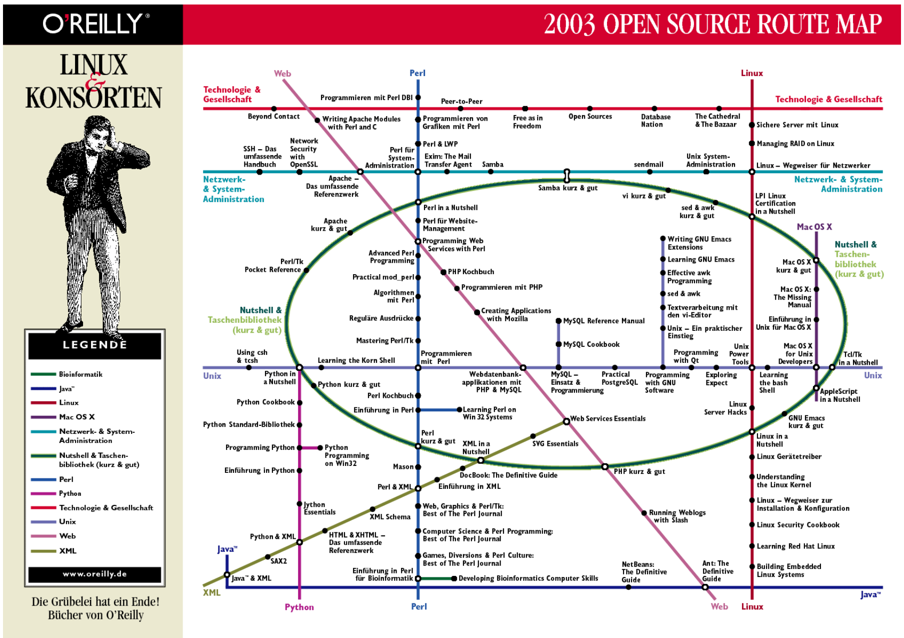


Figure 2.17: Open source route map [ORe03].

where lines represent key trains of thought and stations represent individual ideas.

Lauther and Stübinger [LS01] present a demonstration of software which is capable of laying out schematic diagrams using a force-directed approach with the aim of visualising cable plans schematically.

2.7 Existing Research

2.7.1 Force Directed Metro Map Layout

There is one approach to laying out metro maps using a force directed heuristic by Hong et al. [HMdN04, HMdN06]. In their paper, they put forward five different layout methods which use combinations of spring-based algorithms. The algorithms used are the GEM algorithm [FLM95], the PrEd algorithm [Ber99] and a magnetic spring algorithm [SM95]. Some of their methods include a preprocessing step which involves simplifying the graph by removing all nodes with only two incident edges. They also include a final step of labelling the graph which uses a combinatorial approach to try to achieve a labelling with as few overlaps as possible.

They analyse their methods with regards to a set of five criteria:

- that each line should be drawn as straight as possible
- that there should be as few edge crossings as possible
- that labels should not overlap
- that edges should be drawn orthogonally or at 45°
- that each line should be drawn with a unique colour

They claim that their results satisfy the first four of these criteria (they do not consider the fifth criteria in any great detail).

As their methods were progressively refined to the application, they produced increasingly better graphs. However, the results were hampered as the geographic topology of the maps was not at all considered. This is because they started with a random embedding of the graph, as is common with many other force-directed graph drawing algorithms. They argue that people using metro systems are not concerned with the real topology of the system, but this is clearly not the case when one is using a “northbound” train and the map shows the line going from top to bottom (it seems reasonable to assume that people perceive north as being “up” and south as being “down”.

The results they present generally satisfy their first four criteria. Their best method is their fifth method which seems to produce the most aesthetically pleasing graph. However, the graphs suffer from a number of flaws, most notably the irregular spacing of nodes—some are very close together (so close that you can not discern any edge between them) and others are very far apart. They also do not consider drawing whole train lines; each pair of connected nodes are connected by only a single edge when in many real-world examples, many edges (different train lines) might need to be drawn between nodes. Their labelling step produces an acceptable labelling, but many metro maps tend not to use diagonal labels as horizontal labels are most likely easier to read. The main problem is that their resulting graphs generally have very few features (if any) in common with existing metro maps. This is evidently a problem if the maps were to be used as a replacement for the existing maps—people would most likely have issues if the mental map of the map is too greatly changed.

2.7.2 Mixed-Integer Program for Metro Map Layout

[★ Nollenburg [N0̈5]: description of his approach and an objective evaluation of its strengths and weaknesses.]

2.7.3 Path Simplification for Metro Map Layout

[★ Merrick and Gudmundsson [MG07]]

2.7.4 Centrality-based Scaling

[★ Merrick and Gudmundsson [MG06]]

Chapter 3

Metro Map Layout

This chapter describes our metro map layout algorithm which draws on the observations made of existing metro maps discussed in Section 2.1. We describe how the features that we intend to model can be broken down into a set of criteria and how those criteria can be combined using a multicriteria optimisation method. This chapter is only concerned with moving individual nodes, but there are occasions when the movement of several nodes simultaneously is required. Moving clusters of nodes is described in Chapter 4.

3.1 Aims

There are a a large number of characteristics that are apparent on existing metro maps. These characteristics were discussed in detail in Section 2.1. We will take the following of these characteristics into consideration as part of our layout method:

- **Lines.** Lines will be drawn to take advantage of parallel lines and lines of common angle using angle generalisation—more specifically, lines should be drawn horizontally, vertically or with 45° diagonals. In using lines of

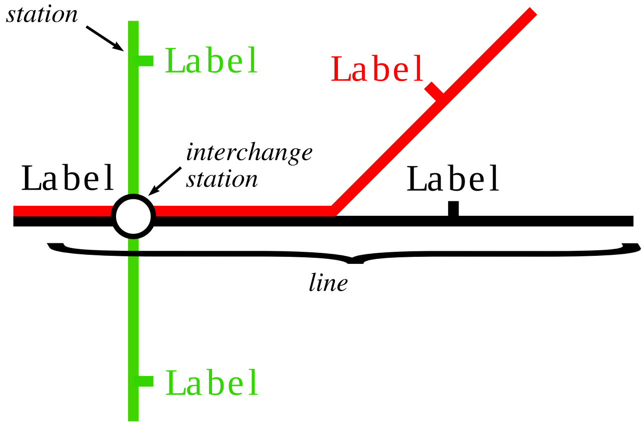


Figure 3.1: Metro map features.

common angle, the shape of lines will also be implicitly generalised. Lines should have stations spaced at regular intervals and the scale of the map should be generalised to allow for this.

- **Labelling.** The maps will have station labels and these labels will have an effect on the overall layout of the map.

3.2 Definitions

We use a graph as an abstract representation of a metro map. In this case, a graph, G, is a set of nodes, V, with connections between pairs of nodes represented by a set of edges, E. When drawing metro maps, we use the nodes to represent stations on the network and edges to represent a single connection between two stations. In some cases, there may be several edges connecting two nodes where two or more lines run together. We also use the term *line* to represent a subset of edges that form a particular line on the network (such as the Central or Northern Lines on the London Underground map). These features are illustrated in Figure 3.1.

We decided to use a graph model to represent metro maps due to the programmatic flexibility which makes tasks such as finding neighboring nodes or incident edges relatively straightforward.

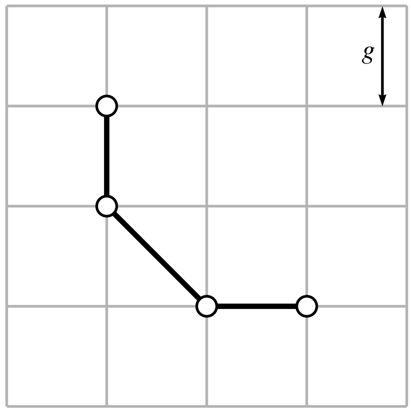


Figure 3.2: Grid used for embedding the metro map graph where g represents the grid spacing.

The graph is embedded on an integer square grid, as shown in Figure 3.2. This means that nodes can only be centred on grid intersections, but there is no explicit requirement for edges to follow grid lines. The spacing between adjacent intersections in the grid is denoted by g. Discretizing the search space in this manner allows us to dramatically reduce the number of potential locations for nodes. Another significant advantage is that producing orthogonal graphs is much easier as nodes are more likely to be in line with one other.

3.3 Hill Climbing Multicriteria Optimisation

Our process is built around a hill climbing multicriteria optimization method that combines a number of criteria that are judged to affect the quality of the graph and some method to change the layout of the graph. We are only ever making improvements to the graph with each change in the layout. We considered a number of other optimization methods such as simulated annealing and genetic algorithms. However, we found that the simpler method using multicriteria optimization was more appropriate. Simulated annealing adds an element of non-determinism in order to escape from local minima in the search space. However, for this to beeffective, a large number of iterations are required–certainly many more than the five to ten iterations that were needed for multicriteria optimization. It was quicker and easier to cater for specific local minima as and when they became apparent. Genetic algorithms converge much more slowly than a hill climber or simulated annealing and would have significantly greater requirements for memory and computation owing to the need to maintain a population to evolve new generations of solutions.

Multicriteria optimization is an iterative process. For each iteration an attempt is made to move each node, followed by clusters of nodes, followed by each node label. The number of iterations can either be fixed beforehand or the process can be stopped if no further improvement can be made. We always use a fixed number of iterations, the number of which is indicated for the examples presented in Chapter 5. Care is made to ensure that the method is deterministic. This allows us to see how subtle changes in the criteria weightings affect the finished maps without randomness becoming a factor.

3.4 Initial Embedding

An important consideration which greatly affects the finished maps is that of the initial layout of the map. There are essentially two possible ways of generating an initial layout for the map. We could start with a completely random layout or with some other topologically correct layout such as the actual geographic layout of the map.

Starting with a random layout is going to make producing a map that corresponds in any way to the geographic layout very difficult. There is no explicit concept of the topology of the map making it impossible to say whether a certain line should be oriented in a particular direction (north-to-south or east-to-west) or to place nodes that are close together in reality close together on the final map. Even if the topology of the map was known and we could reason about the relative positions of nodes (that is to say that one node should be north of another or that a group of nodes are close together), it would be very likely that a large number of iterations and massive fluctuation in the starting layout would be required. It therefore becomes very difficult to predict and reason about potential movements of nodes.

Starting with an actual geographic layout is much more likely to produce better results than a random layout. The entire method then becomes a process of iterative refinement—fewer and smaller node movements would theoretically be required to produce finished maps of acceptable quality than starting with a random layout. Finding the geographic locations of stations in order to position the nodes becomes a time-consuming process, but precise accuracy is not necessarily required. A simple freehand sketch based on knowledge of the geography and topography of the metro system might suffice, particularly for simple maps.

One implicit advantage of metro maps is that they are nearly always planar. In this context, that means that edges are unlikely to intersect other than at nodes—actual intersections usually mean something specific to the geography of the map such as where one line physically passes over or under another line. In fact, it may be essential that this intersection is preserved in the final map. Therefore, an optional step is recommended where each intersection between lines is replaced by a dummy node. The dummy node is treated just like any other node in the graph, except that it does not have a label and is not drawn on the finished map. This ensures that the topography of the edge crossing is retained in the finished layout.

We will be starting with a geographic layout (or a close alternative using a topologically-correct sketch of the map). However, we need to make sure that nodes are first centred on grid intersections, so a way to snap the nodes to the grid is needed. Calculating the nearest grid intersection to a node is simple but care must be taken to ensure that more than one node does not share the same grid intersection. In the case of contention for a particular intersection, the node being snapped should be moved to the nearest grid intersection that is vacant. If the spacing between grid intersections, g, is too large then particularly dense parts of graphs (especially areas where the average length of an edge is less than $0.5g$) may make it difficult to find points close to the starting point for the node. In these case, the value of g should be reduced. Figure 3.3 illustrates how the process of moving nodes to grid intersections works for a simple example.

Figure 3.3 shows an example of how nodes are centred on grid intersections. The graph on the left is the initial layout and the graph on the right is the new layout after nodes (A, B, etc.) have been moved to their new positions on grid intersections (A', B', etc.). In this case two nodes (B and G) could potentially both move to the same grid intersection, but to avoid this, G' is positioned on the next nearest grid intersection.

3.5 Node Movement

Movement of nodes is achieved through the calculation of several criteria which are judged to affect the aesthetic quality of the map. We have implemented a total of six different node movement criteria:

- **Angular Resolution Criterion.** Maximises the angular resolution of incident edges at each node.

- **Balanced Edge Length Criterion.** Attempts to ensure that the length of edges incident to a particular node are approximately equal.

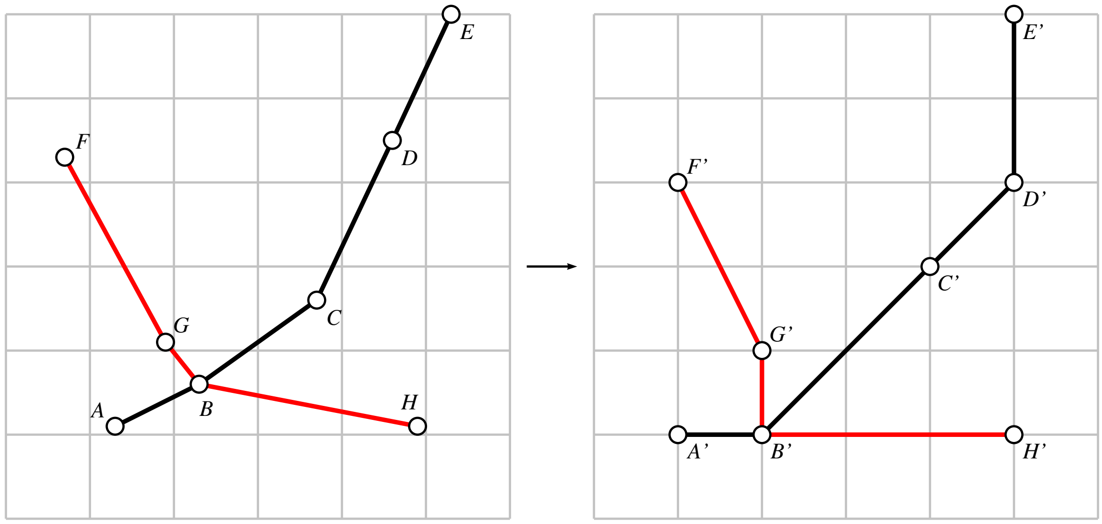


Figure 3.3: Moving nodes to grid intersections. The left-hand graph is the initial layout; the right-hand graph is the same graph but with nodes moved to grid intersections. In the cases where there is already a node occupying a grid intersection, the next nearest grid intersection is used—this is the case with nodes B and G in this example.

- **Edge Crossings Criterion.** Attempts to minimise unnecessary edge crossings.
- **Edge Length Criterion.** Attempts to ensure that edge lengths across the whole map are approximately equal.
- **Line Straightness Criterion.** Prefers edges that form part of a line to be parallel either side of each node that the line passes through.
- **Octilinearity Criterion.** Attempts to ensure that each edge is drawn horizontally, vertically or diagonally at 45°.

As well as these six criteria, we have implemented four node movement rules which are strictly enforced during the layout process:

- **Bounding Area Restriction Rule.** Restrict the movement of nodes to be within a certain bounding area.

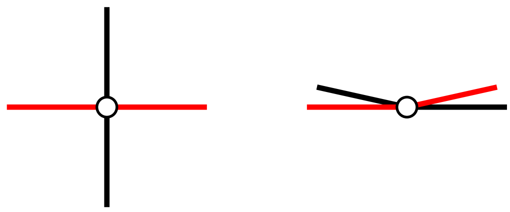


Figure 3.4: Examples of optimal angular resolution (left) and poor angular resolution (right).

- **Geographic Relationships Rule.** Enforce the geographic relationships between pairs of nodes (for example, that one node should be north of another node).
- **Occlusions Rule.** Avoid the introduction of occlusions of other edges and nodes.
- **Edge Ordering Rule.** Preserve the ordering of edges incident to a node.

These rules and criteria, as well as the strategy by which they are used to determine how nodes are to be moved are discussed in more detail in the following sections.

3.6 Node Movement Criteria

3.6.1 Angular Resolution Criterion, c_1

In some metro maps there are occasions where many lines pass through a single node creating the situation where that node has many incident edges. If the edges are drawn such that there is only a small angle between any two adjacent edges then it can become difficult to distinguish between them (particularly if the edges are similarly coloured). Figure 3.4 illustrates this point—the left-hand example shows four incident edges which are arranged so that the angle between each adjacent pair of edges is maximised; the right-hand example has only small angles

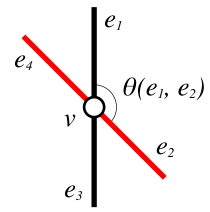


Figure 3.5: Calculating the angular resolution criterion.

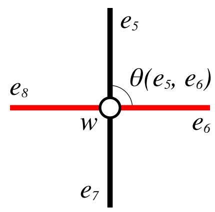


Figure 3.6: Calculating the angular resolution criterion.

between two pairs of edges but very large angles between the other two pairs of edges. The angular resolution criterion, c_1, ensures that there is as large an angle as possible between adjacent edges incident to a node. The criterion is calculated using Equation 3.1:

$$c_1 = \sum_{v \in V} \sum_{\{e_1, e_2\} \in E_v} \left| \frac{2\pi}{\rho(v)} - \theta(e_1, e_2) \right| \tag{3.1}$$

where $\rho(v)$ is the degree of the node v and $\theta(e_1, e_2)$ is the angle between two adjacent edges e_1 and e_2 incident to v.

Figure 3.5 shows an example of calculating the angular resolution criterion for a node v. In this example, $\rho(v)$ is 4, so the ideal spacing between each adjacent pair of edges is $\frac{2\pi}{\rho(v)}$ or $\frac{\pi}{2}$. The angle θ between edges e_1 and e_2 is $\frac{3\pi}{4}$, so for that particular pair of edges, the absolute difference between the actual and ideal angles is $\left|\frac{2\pi}{4} - \frac{3\pi}{4}\right| = \frac{\pi}{4}$. For the other pairs of edges, the difference is also $\frac{\pi}{4}$, so for the area around v, $c_1 = \frac{4\pi}{4} = \pi$.

Figure 3.6 shows another example for a node, w. This time, the lines pass through w at right angles. Again, $\rho(w) = 4$ therefore the ideal spacing between adjacent pairs of edges is $\frac{\pi}{2}$. $\theta(e_5, e_6) = \frac{\pi}{2}$, so the difference between the ideal

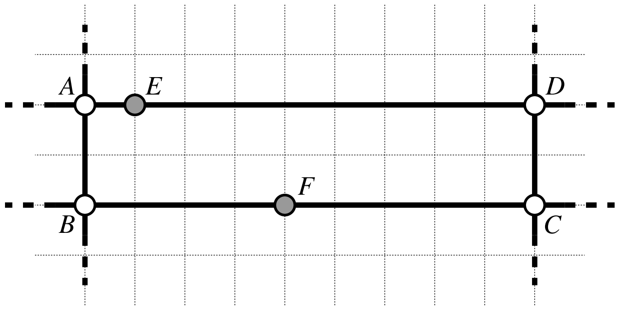


Figure 3.7: Balanced edge lengths.

angle and the actual angle is $\frac{\pi}{2} - \frac{\pi}{2} = 0$. The angles between each of the other three pairs of adjacent edges around w are equal, so the total value of c_1 for the area around w is 0.

3.6.2 Balanced Edge Length Criterion, c_2

One of the characteristics of metro maps is that there are many nodes with two incident edges (degree two). This is very often the case when a line passes through a sequence of several stations. Unfortunately, the edge length criterion as discussed previously does not handle this particular feature of the maps effectively. The balanced edge length criterion overcomes this particular limitation.

Figure 3.7 shows an example whereby there are two nodes with degree two. If we are only considering the edge length criterion for these two nodes, we can show that the criterion evaluates to the same value for both nodes. First assume that the preferred edge length, l, is $1g$. In the case of node E, the edge length criterion evaluates to

$$\left(\frac{|e_{AE}|}{lg} - 1\right) + \left(\frac{|e_{ED}|}{lg} - 1\right) = \left(\frac{1}{1} - 1\right) + \left(\frac{8}{1} - 1\right) = 7 \tag{3.2}$$

and for node F

$$\left(\frac{|e_{BF}|}{lg} - 1\right) + \left(\frac{|e_{FC}|}{lg} - 1\right) = \left(\frac{4}{1} - 1\right) + \left(\frac{5}{1} - 1\right) = 7 \tag{3.3}$$

Clearly all the potential locations for E between A and D will result in the same value for the edge length criterion. However, we want to ensure that the edge lengths are approximately equal–this was after all the original intention of the edge length criterion. In these cases, the balanced edge length criterion can help by penalising nodes with degree two that have incident edges with unbalanced lengths.

Calculation of the balanced edge length criterion, c_2, is simply the sum of the absolute difference between the lengths of the two incident edges of every degree-two node in the graph as shown in Equation 3.4:

$$c_2 = \sum_{v \in V, \rho(v)=2} |e_1| - |e_2| \tag{3.4}$$

Using the nodes E and F from Figure 3.7 as an example, the balanced edge length criterion for E is

$$(1 - 8) = 7 \tag{3.5}$$

and for F is

$$(4 - 5) = 1 \tag{3.6}$$

where e_1 and e_2 are the incident edges of node v which has degree $\rho(v) = 2$.

The optimal position is therefore where F is (or indeed the grid intersection immediately to the right of F). If the lengths of the two incident edges to a node are equal, then the balanced edge criterion for that node evaluates to zero: the edge lengths are balanced.

3.6.3 Edge Crossings Criterion, c_3

In the case of metro maps, edge crossings imply some kind of topographic feature such as two unconnected lines crossing. If an edge crossing is intentional, then a dummy node can be inserted at the crossing point and the map drawn with that node in place. The dummy node can then be removed after the map is drawn or simply left in place and not rendered. However, unintentional edge crossings can affect the readability of the map so that the map is not as easy to comprehend [PCJ95].

Calculation of the edge crossings criterion, c_3, is straightforward. Each edge is checked to see whether any other edges cross it: c_3 is then the total number of crossings in the graph.

Special care has to be taken in the case where more than one line runs in parallel between two nodes. In this case, even though several edges are actually being crossed, only a single crossing is counted for the purposes of this criterion.

Two efficiency enhancements can be made for this criterion. Firstly, as we are using a rule that stops new edge crossings being introduced (see Section 3.7.3), we only need check edges that we already know cross other edges. Secondly, If the graph is already planar (so that there are no edge crossings) or can be made planar (by the introduction of dummy nodes), then there is clearly no need to calculate this criterion. (The weighting for the criterion can be set to zero to indicate that we do not want to calculate the value for that criterion.)

3.6.4 Edge Length Criterion, c_4

A common feature of metro maps is that stations should be spaced evenly along lines and that the spacing should be should be reasonably consistent across the entire map. This comes about because the map is drawn to a irregular scale such that the scale decreases towards the extremities of the map. This scale generalisation was explored in Section 2.1.2.

The edge length criterion, c_4, is based on some multiple, l, of the grid spacing g. The purpose of the criterion is to penalize edges that are longer than or shorter than lg. The edge length criterion is calculated using Equation 3.7:

$$c_4 = \sum_{e \in E} \left| \frac{|e|}{lg} - 1 \right| \tag{3.7}$$

where $|e|$ is the length of edge e. l should always be greater than or equal to 1 as it is impossible to have two nodes less than one grid spacing apart. In the case where $|e| = lg$, the length of edge e is exactly the length that we prefer and the criterion for e evaluates to zero. If $|e| < lg$ or $|e| > lg$ then the value of c_4 for that edge will be greater than zero.

To illustrate how the edge length criterion works, consider the example shown in Figure 3.8. There are three edges in this small graph, AB, BC and BD, with lengths 2, $\sqrt{18}$ and 6 respectively. If we assume that $g = 1$, we can see how c_4 would vary for different values of l by looking at the example values in Table 3.1. Notice how the criterion evaluates to give significantly higher values for edges which are longer than lg than those that are shorter. This has the effect of providing greater pressure on the graph to compress than to expand. This makes sense when we remember that the value of g and l should be chosen to suit the densest parts of the starting layout of the graph, meaning that most edges will be longer than lg.

While this works fine for horizontal or vertical edges, diagonal edges pose an interesting problem, as might have been noticed for edge BC in Figure 3.8. Where diagonal edges are concerned, it is very likely that there does not exist a position on the grid where the edge length criterion can be zero. Figure 3.9 shows a trivial example of such a case. If l is 1, then the edge AB can be drawn to have length of exactly $1g$, whereas the diagonal edge AC can only have a minimum length of

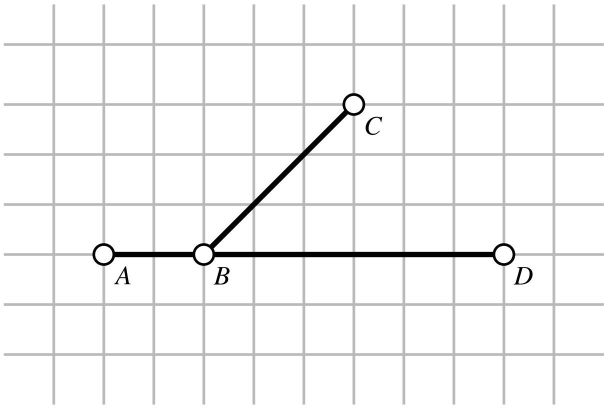


Figure 3.8: Example used to illustrate the edge length criterion.

Table 3.1: Examples of how the edge length criterion varies with different edge lengths. The edges AB, BC and BD are shown on Figure 3.8 and have lengths 2, $\sqrt{18}$ and 6 respectively. In calculating c_4 for each of these edge lengths, we have assumed that $g = 1$. c_4^{XY} means the calculation of c_4 for just the part of the graph between nodes X and Y.

l	c_4^{AB}	c_4^{BC}	c_4^{BD}	c_4
1	1	3.24	5	9.24
2	0	1.12	2	3.12
3	0.33	0.41	1	1.74
4	0.5	0.06	0.5	1.06
5	0.6	0.15	0.2	0.95
6	0.67	0.29	0	0.96

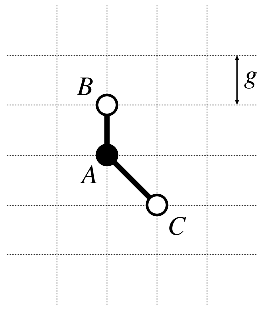


Figure 3.9: Problem with diagonal edge lengths and the edge length criterion.

$g\sqrt{2}$. Rather than making specific allowances for diagonal edges to break from the grid, we use a combination of the edge length criterion and other criteria (specifically the octilinearity criterion) to counteract this.

3.6.5 Line Straightness Criterion, c_5

One of the important features of metro maps is that lines appear to pass through nodes so that the entry edge is more-or-less directly opposite the exit edge. It is not desirable for the line to turn sharply as it passes through a node (so that it makes a 90° or 135° turn). This is made all the more important if there are two or more lines passing through a node–if both of the entry edges are opposite each other and both of the lines make a 90° turn so that the exit edges are opposite, the readability of the map is degraded (especially if the colour of the lines are similar). Figure 3.10 shows examples where the graphs on the left can have the line straightness improved by moving nodes.

To counter this, we introduce the line straightness criterion, c_5, calcuated using Equation 3.8:

$$c_5 = \sum_{(v \in V)} \left(\sum_{e_1, e_2 \in E} \theta\left(e_1, e_2\right) \right) \tag{3.8}$$

where $\theta\left(e_1, e_2\right)$ is the angle between adjacent edges e_1 and e_2 and e_1 and e_2 are the only two incident edges of the same line incident to the node v. If e_1 and

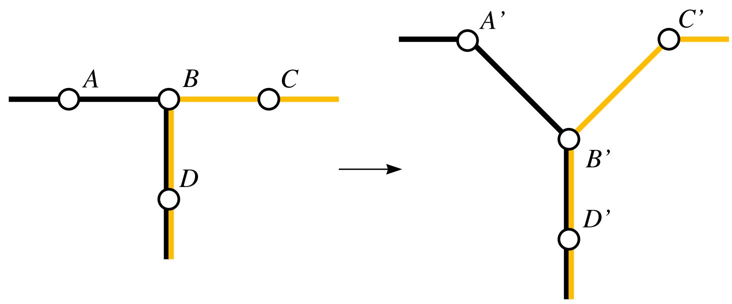


Figure 3.10: Examples of poor line straightness (left) and improved line straightness (right).

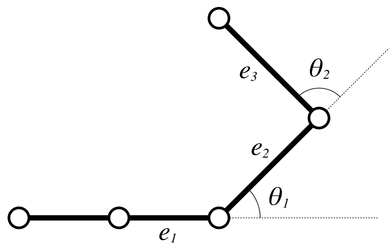


Figure 3.11: Example of the calculation of the line straightness criterion.

e_2 are parallel (i.e. they pass through v in a straight line), $\theta(e_1, e_2) = 0°$; if the two edges are at right angles, then $\theta(e_1, e_2) = 90°$. The intuitive effect of this is to penalise turns in edges where $\theta(e_1, e_2)$ is large more than turns where $\theta(e_1, e_2)$ is small or zero.

In the example in Figure 3.11, the line in question includes three edges, e_1, e_2 and e_3. The line makes two turns between each two pairs of adjacent edges. To calcuated the line straightness criterion of this example, we simply sum the angles between edges e_1 and e_2 (θ_1) and e_2 and e_3 (θ_2): $\theta_1 + \theta_2 = 45° + 90° = 135°$.

3.6.6 Octilinearity Criterion, c_6

The purpose of this criterion is to ensure that edges are drawn at some multiple of 45°, either orthogonally (vertically or horizontally) or diagonally with respect to the grid. The octilinearity criterion, c_6, has the effect of penalizing edges that are not some multiple of 45° and is calculated using Equation 3.9.

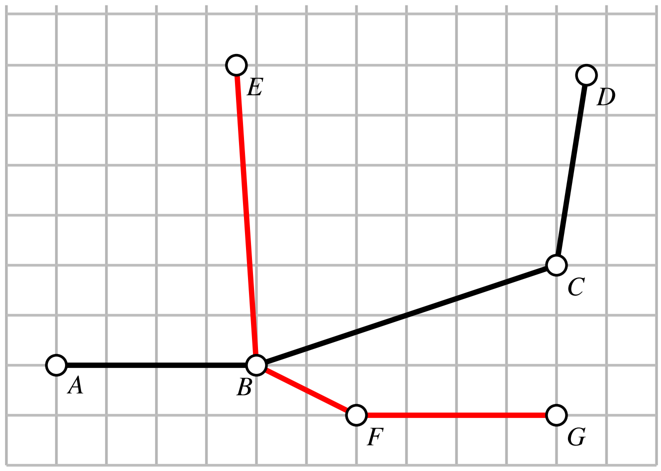


Figure 3.12: Example of the calculation of the octilinearity criterion.

$$c_6 = \sum_{\{u,v\}\in E} \left| \sin 4 \left(\tan^{-1} \frac{|y(u) - y(v)|}{|x(u) - x(v)|} \right) \right| \tag{3.9}$$

where $\{u, v\}$ is an edge between nodes u and v, and $y(v)$ and $x(v)$ are the y-and x-coordinate of node v respectively.

Figure 3.12 shows an example of a graph that we will use to illustrate the calculation of the octilinearity criterion. The result of calculating the criterion c_6^e for each edge $e = \{u, v\}$ in this example graph is shown in Table 3.2. As is expected, edges which are already at an angle of some multiple of 45° (AB and FG) evaluate to zero, whereas edges which are at angles furthest from multiples of 45° evaluate to the highest values (edges BC and BF are at about 26.57°).

3.6.7 Criteria Weightings

The intention of the weightings for each of the criteria is twofold. Firstly, the functions generate values which can vary by an order of magnitude or more between each of the criteria. The weightings allow the values of each criterion to be brought within the same magnitude of each other. This is important to ensure that one criterion doesnt completely overwhelm the other criteria. Secondly, by

Table 3.2: Examples of octilinearity criterion calculations with reference to Figure 3.12.

Edge, $e = \{u, v\}$	c_6^e
$\{A, B\}$	$\left\|\sin 4\left(\tan^{-1}\frac{0}{4}\right)\right\| = 0$
$\{B, C\}$	$\left\|\sin 4\left(\tan^{-1}\frac{2}{6}\right)\right\| = 0.96$
$\{C, D\}$	$\left\|\sin 4\left(\tan^{-1}\frac{3.8}{0.6}\right)\right\| = 0.586$
$\{B, E\}$	$\left\|\sin 4\left(\tan^{-1}\frac{6}{0.6}\right)\right\| = 0.388$
$\{B, F\}$	$\left\|\sin 4\left(\tan^{-1}\frac{1}{2}\right)\right\| = 0.96$
$\{F, G\}$	$\left\|\sin 4\left(\tan^{-1}\frac{0}{4}\right)\right\| = 0$

using a higher weighting, a preference can be placed on a particular criterion if the effects of that criterion are required to be more prominent. Similarly, a lower weighting can be used to reduce the effect of a particular criterion. If an application of the method doesnt require a particular criterion, the weighting can be set to zero. This allows flexibility with the criteria in deciding which criteria should contribute to the characteristics of the map for various applications.

The sum of the weighted criteria, m, is given by Equation 3.10:

$$m = w_1c_1 + w_2c_2 + w_3c_3 + w_4c_4 + w_5c_5 + w_6c_6 \tag{3.10}$$

The values for w_i can be modified by the user depending on the characteristics of the particular metro map being drawn.

[★ Put some examples of weightings here - Table 3.3 - Figure 3.13]

To illustrate how the weighted criteria combine and to give some idea of the magnitudes of each of the weightings, an example of a complete graph is shown in Figure 3.13, together with the weighting and weighted value for each criterion shown in Table 3.3.

The total of the weighted criteria in Figure 3.13 is:

Figure 3.13: Simple example of node movement criteria for a complete graph. The values for each criterion for this graph are shown in Table 3.3.

Table 3.3: Typical node movement criteria weightings.

Criterion, c_n	Weighting, w_n	$w_n c_n$ for graph in Figure 3.13
c_1		
c_2		
c_3		
c_4		
c_5		
c_6		

$$\begin{aligned} m &= x + x + x + x + x + x \\ &= X \end{aligned}$$

3.7 Node Movement Rules

As well as all the weighted criteria, a number of rules were devised which serve the purpose of restricting the potential movement of a node. These rules essentially forbid movement of a node to a new location if any of the rules are broken. The reason why rules are used in addition to criteria is that in some cases the criteria are unable to guard against introducing very undesirable node movements that would otherwise be difficult (or inefficient) to handle with more criteria. Specific examples justifying the reason for each rule that we used are detailed later.

We created rules to restrict the movement of a node to stay within the boundaries of a drawing area; to enforce the geographic relationships between nodes; to cope with occasions when other nodes or edges may be occluded; and to preserve the ordering of the edges around a node. This set of rules are described in the following sections.

3.7.1 Restrict Movement to Boundary of Drawing Area

When we were drawing maps on a finite drawing area, we needed some way to make sure that any node movement did not cause the node to move outside the drawing area. This wasn't always necessary—in practice, we always ensured that the drawing area was large enough to contain the graph and any potential growth of the overall dimensions of the graph. However, there were still some occasions when this rule was useful and effective.

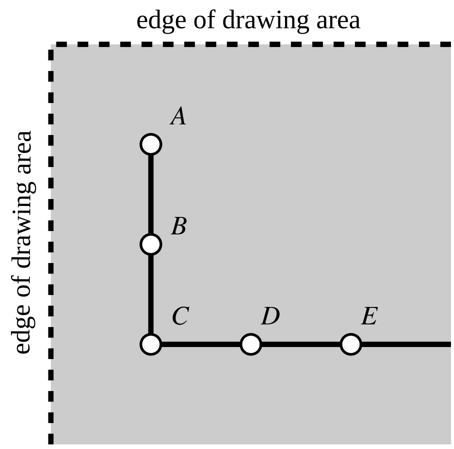


Figure 3.14: Restricting node movements to stay within a drawing area boundary. Nodes C, D and E form part of a longer straight line and ideally nodes A and B would move to be parallel to them, but the edge of the drawing area restricts this movement.

To illustrate this rule, consider the graph shown in Figure 3.14. In this example, nodes C, D ane E form part of a longer straight line and ideally nodes A and B should move so that they would be parallel to them. However, moving nodes A and B in this way would put them either wholly or partially outside the drawing area.

3.7.2 Enforcement of Geographic Relationships

Although metro maps are a generalization of the real geography of the network, relationships such as one node being north of another node still indicate some general meaning in the drawn map. This rule allows these geographic relationships to be enforced so that the relative positions between nodes do not change.

Figure 3.15 illustrates the effect of enforcing geographic relationships. In the left-hand diagram, node A is being moved but its initial position with respect to B means that it is only free to move in the area that is both above and to the left of B (the grey shaded area). Notice that we still allow a movement so that A is directly to the left or above B—if this was not the case then it wouldn't be

Figure 3.15: Enforcement of geographic relationships. The grey shaded area shows the degree of freedom afforded node A (left) and node C (right).

possible to move A into an optimal octilinear position with respect to B.

In the right-hand figure node C is being moved but is already immediately to the left of D. This allows C freedom to move anywhere that is to the left of—or directly above or below—D.

One of the side-effects of allowing nodes to move to be vertically or horizontally aligned with another connecting node (as C and D are initially) is that on the second iteration of moving a node it may move into a position that breaks the initial geographic relationships. For example, if we started with a graph as in the left-hand diagram of Figure 3.15 and node A moved to a position immediately to the left of B, the graph would become equivalent to the right-hand graph. At the next movement of A, the freedom of movement is the same as that for C, allowing A to move below B. In practice, this tended not to be the case—for it to happen, a new bend in a line is likely to be introduced which would make that position less optimal when calculating the weighted criteria, especially with regard to the line straightness criterion.

3.7.3 Handling Node and Edge Occlusions

In Section 3.6.3 we introduced a criterion which penalises edge crossings in the graph. While this criterion is fine for removing unwanted edge crossings, it is

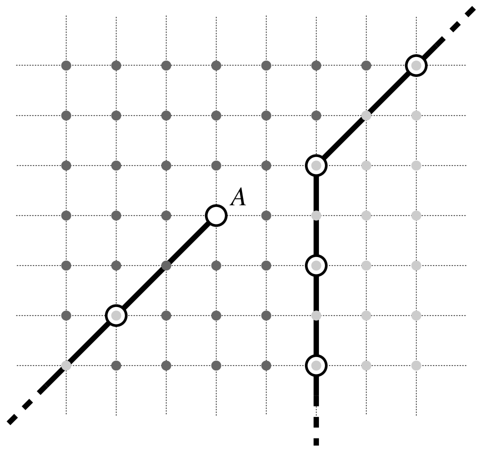


Figure 3.16: How node and edge occlusions can restrict the search space. Light grey dots show positions where node A can not be moved to because either a node or edge occlusion would result. Dark grey dots show valid positions where no occlusions would happen.

usually the case that we do not want new edge crossings to be introduced. We also want to make sure that any node movement does not cause another edge or node to be occluded. The node and edge occlusions rule takes care of this.

The node and edge occlusions rule specifically stops the introduction of any of these situations by moving a node A:

- any edge incident to A crossing or lying on top of any other edge
- any edge incident to A crossing any other node
- node A moving such that it will be on top of any other edge
- node A moving such that it will be on top of any other node

The second and third of these restrictions are especially important. If an unconnected node is moved so that it looks as though it is connected to another edge, someone reading the graph may interpret it as representing a genuine connection.

Figure 3.16 shows an example of the restrictions imposed by the node and edge occlusions rule. In this figure, node A is being moved and the grey spots

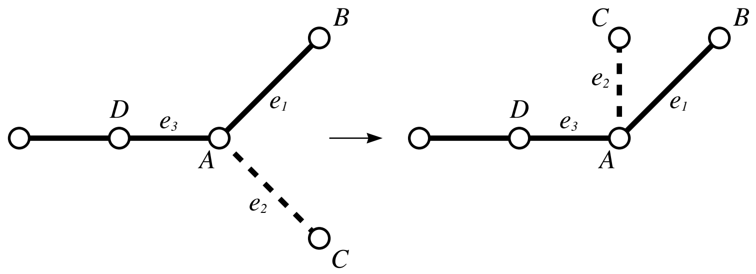


Figure 3.17: Preservation of edge ordering. Without preserving the ordering of edges, node C would be able to move as shown, changing the topography of the map.

represent possible new locations for A. In locations where any of the above four situations occur the spot is shown with light grey, meaning that that location would invalidate the node and edge occlusions rule. Node A cannot move to any of these locations. The spots coloured with dark grey do not introduce any node or edge occlusions so these are valid new locations for A. Out of the 49 possible locations for A, 20 would invalidate the rule, meaning that the node movement criteria need only be calculated for the remaining 29 locations.

3.7.4 Preservation of Edge Ordering

The geographic relationships criterion (Section 3.7.2) allows us to restrict the geographic relationships between two nodes. However, there are limitations to this rule that mean that the topology of the graph could be changed by the movement of a node. Figure 3.17 shows such a problem. In this example, node C is being moved. If we assume that the most optimal new position for C is as shown in the right-hand diagram, we can see that the topology of this graph has been changed. This is easy to show by considering the order of the incident edges to A. In the left-hand diagram the clockwise ordering of edges starting with e_1 is $\{e_1, e_2, e_3\}$. In the right-hand diagram, the change in position of C has changed the order of the edges to $\{e_1, e_3, e_2\}$.

To implement this rule we need to find the clockwise ordering of edges around the node being moved and any neighbouring node in the graph. Using the example of the left-hand diagram in Figure 3.17 as a starting point, the ordering around the node being moved, C is $\{e_2\}$ and the neighbouring node, A is $\{e_1, e_2, e_3\}$. We can move node C to any position that maintains these edge orderings, so the edge orderings at each potential new location for C must be checked and the location disregarded if the orderings change.

3.8 Moving Nodes

The way that nodes are moved greatly affects the outcome of the final drawing of the metro map. There are a number of points that need to be considered when selecting a position to move a node to: the sum of the weighted criteria, m; whether or not another node occupies that grid intersection; whether moving the node would occlude other nodes or edges; how far to move the node; whether the distance to move the node is reduced with each iteration (cooling); and whether the cyclic ordering of edges incident to a node would change.

Our approach is to specify a maximum radius within which a node can move. This is given as some multiple, r, of the grid spacing g. As the whole process effectively refines a sketch of the map or the geographic layout of the map, the value of r is usually fairly small (generally than 10). Larger values of r would allow movements that could alter the map so that it differs too greatly from reality. A larger value of r is chosen for maps with small values of g so that nodes can move greater distances if there are many grid intersections between connected nodes. In the case of a large map, such as the London Underground map, g has to be small to allow for enough grid intersections for nodes in the dense centre of the map, while the extremities are relatively sparse. An example of the potential movements for a node when $r = 3g$ is shown in Figure 3.18.

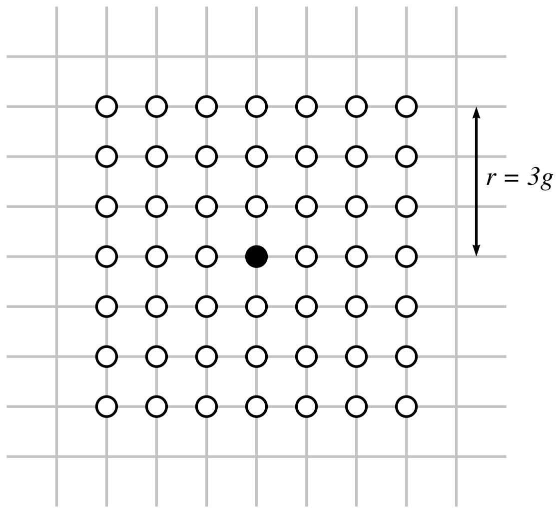


Figure 3.18: The node movement radius. The black node in the centre is being moved while $r = 3g$: all the potential new locations in this radius are shown with white nodes.

3.8.1 Cooling

A cooling process is used whereby the maximum radius that a node can move, r, is reduced to 1 by the last iteration, using a linear cooling schedule. The purpose of cooling is to allow large movements to be made during early iterations, with later iterations moving nodes less and less to allow for finer refinement. We experimented with a number of different cooling schedules such as a logarithmic schedule or an irregular schedule. However, the difference between results using each different schedule was negligible. It therefore made sense to use the least computationally expensive linear cooling schedule.

When considering potential movements for a node, the initial value of m_0, is first calculated. Each grid intersection up to r intersections from the initial node location is tested by moving the node there and recalculating m. A set of locations, T, is remembered for each potential movement where the new value of m is less than the initial value of m_0. The node is moved to the location in T that has the smallest value of m. In the case of no potential movements being discovered, the node is returned to its starting location; if there is more than one

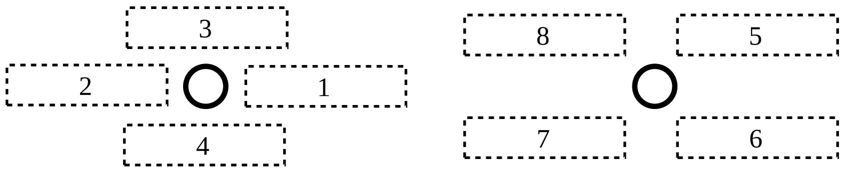


Figure 3.19: Label Positions

location with the same smallest value of m, we select the first improved location that was found.

3.9 Labelling

Labelling is an integral part of metro maps and hence it should form an integral part of our multicriteria optimization method. To this end, a number of criteria are introduced for label placement. In order to reduce the number of potential locations for labels and to enable us to be able to prefer one position over another, we limit the number of positions using a labelling space. Figure 3.19 shows our chosen labelling space, which consists of eight different potential label positions.

Occasionally a label might contain a large amount of text with several words, which is why we also allow long, many-word labels to span multiple lines. The way in which the decision whether the split a long label over multiple lines was to cause a line break in the label if the length of the label exceeded $0.75lg$ (where g is the spacing between adjacent grid intersections and l is the preferred edge length). This value was chosen because the distance between nodes tends towards lg: it would be better if labels were to fit between labels to avoid a label for one node appearing too close to another node. No attempt was made to split single words that were longer than $0.75lg$. Also, labels were still split over several lines even in the case where there is plenty of space for it in the graph. Figure **??** shows an example where a long label with two words can be split over two lines.

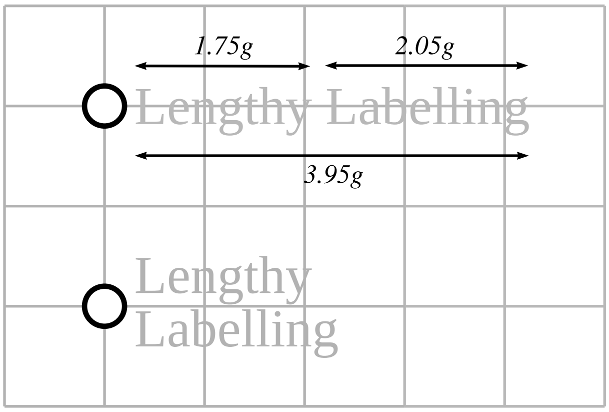


Figure 3.20: Splitting a long, multi-word label of multiple lines. The word "Lengthy" is about $1.75g$ and "Labelling" is about $2.05g$ causing the label to be about $3.95g$ in total. The bottom label shows how the label "Lengthy Labelling" can be split over two lines.

Labelling for all the nodes in the graph was performed once during each iteration, after all of the individual nodes had been moved. We could have attempted to label the map at the same time as nodes were being moved, but due to the large amount of computation required this proved to be excessively slow. Potential label positions are tested in the same way that potential node locations were tested when moving nodes; the sum of the weighted labelling criteria is calculated for each position and the one with the best improvement to the initial label position is chosen.

3.9.1 Labelling Criteria

We have implemented a total of seven labelling criteria:

- **Label Occlusion Criteria.** Three criteria that take into account the number of nodes, edges and other labels that intersect labels.
- **Label Position Criterion.** Places a preference on label positions by weighting each position.

- **Label Position Consistency Criterion.** Gives preference to labels along a line in the map that consistently appear on the same side of the line.
- **Node Proximity Criterion.** Considers labels as they come into close proximity to non-related nodes with the intention of discouraging labels from being positioned too close to other unrelated nodes.
- **Perpendicular Tick Criterion.** Attempts to ensure that the tick (and therefore the position of the label) for a particular node is perpendicular to the line.

As with the node movement criteria, the labelling criteria are weighted with individual weightings. The criteria are described in more detail in the following sections.

3.9.2 Label Occlusion Criteria

These are a set of three criteria which take into account occasions when a label intersects or occludes any other label, node or edge in the graph. As these intersections drastically reduce the readability of the map, it is highly desirable to ensure that they happen as infrequently as possible. However, there may be occasions where the readability of the graph would be improved if a label were allowed to occlude an edge. In these cases, the label may not fit in any position around a node such that it doesn't occlude an edge without first significantly altering the layout of that part of the graph. As this would then have a further impact on the rest of the graph, it would be very difficult to make space for the label.

The unweighted value of these occlusion criteria is the sum of the number of occlusions. For example, when counting node occlusions, if one label in the graph occluded two nodes, the unweighted value of the node occlusion criterion would be two.

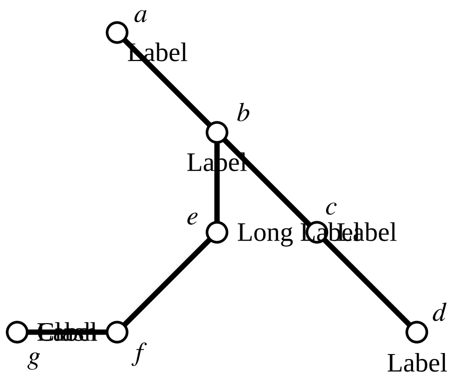


Figure 3.21: Illustration of the label occlusion criteria.

Table 3.4: Number and type of label occlusions for each node in Figure 3.21.

Occlusion Type	Number of Occlusions	Nodes with Occlusions
Label-label	4	c, e, f, g
Label-node	1	e
Label-edge	7	a, b (twice), e (twice), f, g

Figure 3.21 shows an example of a graph that has been poorly labelled. In this case, there are seven nodes which have labels and all except one of these labels (label d) occlude other labels, nodes and edges. Table 3.4 shows how many occlusions of each type are present in this example and which label exhibit occlusions. Notice that, although only there are only two examples of occluding labels, the count for the label-label occlusion is four.

3.9.3 Label Position Criterion

As discussed in Section refsection:labelling, the positions of labels can have a significant impact on the readability of maps. We use a position criterion with a limited search space to allow us to allocate a preferred position to each node label in the graph. Just as with the limitation of the search space for moving nodes (by the restriction of only allowing nodes to be positioned at the intersection of grid lines), we limit the search space for labelling to just a small selection of positions.

This isn't necessarily an artificial limitation as many existing metro maps restrict themselves to a small number of positions for labels. Another restriction on our search space is that we only consider labels drawn horizontally and not vertically or diagonally. While there are maps which use vertical or—more commonly—diagonal labels, they do tend to be a minority with most maps preferring to use only horizontal labels. The reasoning behind this is that it is easier to read the text of the labels if they are all drawn in the same orientation—frequent switching of the orientation of labels will make the maps harder to read.

A label can occupy any one of the eight locations in the labelling space shown in Figure 3.19. These positions will be referred to by their position with respect to a compass oriented so that north is to the top. Therefore, a label in position 1 is said to be to the east of the node and a label in position 7 is said to be to the south-west of the node. Some label positions are more preferential than others, so we use a set of weightings for each different position in the labelling space. Table 3.5 shows the set of weightings for each position in the labelling space. The label position criterion is then defined as the sum of the position weightings for each label in the graph.

By setting the weightings, it is possible to say which label position is the most preferential and which is least preferential. So for example, a label that is directly to the east of a node (position 1) has a weighting of 1.0 and is therefore the most preferred location, while a label which is oriented the the north-west of a node has weighting 1.8 and is the least preferred location. The values chosen for each of the weightings were determined through experimentation with a number of scenarios. Existing metro maps were also inspected to see whether there was a preference for any particular label position. The preference that designers place on label positions is reflected in our weightings. Where there wasn't a particular preference, the weightings were chosen so that they were similar but not identical

Table 3.5: Label Position Weightings. The positions refer to the positions shown in Figure 3.19.

	position	weighting
1	east	1.0
2	west	1.1
3	north	1.4
4	south	1.4
5	north-east	1.5
6	south-east	1.6
7	south-west	1.7
8	north-west	1.8

(for example, there is no difference between the weightings for north and south labels[1]). This then allows other labelling criteria to have a greater effect on the label position.

3.9.4 Label Position Consistency Criterion

The label position consistency criterion attempts to ensure that all the labels along a line appear on the same side of the line. In other words, the effect of the criterion is to prefer labels which have the same position as the labels of their neighbouring nodes. This has the effect of preferring labels that follow the same side of a line and therefore improving readability. Readability is improved because the labels appear as a list which can be read easily rather than having to follow from one side of the line to the other with your eyes. The criterion is only calculated for labels with exactly two neighbouring nodes—nodes with more than two neighbours would be complex to deal with as the label would have to be consistent with more than one line.

The calculation is fairly simple: for each node in the graph with degree greater

[1] This is because we sometimes want labels to alternate either side of a horizontal line where there isn't enough horizontal space between adjacent labels to fit each label. This is discussed with examples in Section 2.1.3.

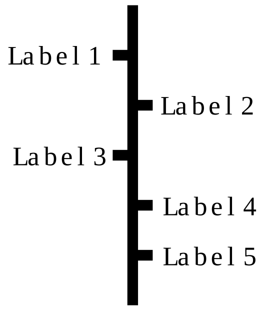


Figure 3.22: Label Position Consistency

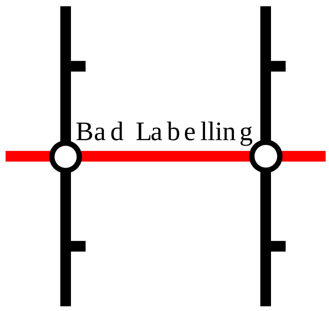


Figure 3.23: An example of ambiguous labelling

than or less than two, a count is kept of the number of times the position of the label of an adjacent node (if that node has degree less than three) differs to the position of the current node. Figure 3.22 shows an example of poor label position consistency where the unweighted value of the label position criterion would be three (there are differences in label position between Label 1 and Label 2, Label 2 and Label 3 and Label 3 and Label 4).

3.9.5 Node Proximity Criterion

The node proximity criterion addresses the problem whereby a label for an individual node is positioned such that the node to which it belongs is uncertain. For example, Figure 3.23 shows a label ("Bad Labelling") which may belong to either the left-hand or the right-hand node.

★ Describe calculation of criterion.

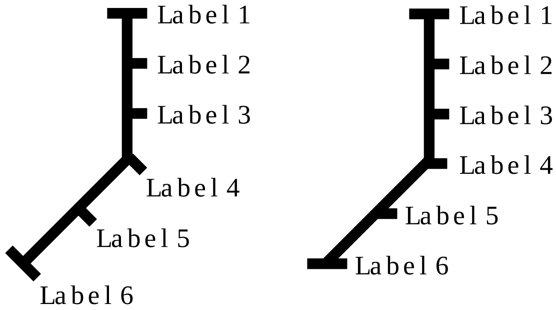


Figure 3.24: Examples of perpendicular tick labels (left) and non-perpendicular tick labels (right)

3.9.6 Perpendicular Tick Criterion

One of the disadvantages of using a combination of criteria to decide on the position of a label is that it may choose positions that don't fit with the characteristics of many existing metro maps. One such characteristic relates to the labelling of nodes with degree two which form part of a line. As we have seen earlier in Section 3.9.4, in these cases we prefer the labels to follow the same side of the line. If we are drawing the icon that represents a station as a tick or a bar, then that icon should be perpendicular to the line. Figure 3.24 illustrates this point. The left-hand diagram shows a line where the ticks showing stations have been drawn perpendicular to the line. The right-hand diagram shows ticks always drawn straight to the right (labels are positioned to the east). The difference is quite clear—while the labels and ticks for the vertical part of the line remain the same, the perpendicular ticks on the diagonal part of the line are more prominent. The minimum distance between the line and the labels on the diagonal part is also greater when the labels are drawn to the south-east, but the association with the relevant tick is not lost. This criterion does not become irrelevant if we use a different device for representing nodes (such as a ring) as the criterion will force the closest part of the label to the line to be furthest from the line but still close to the node.

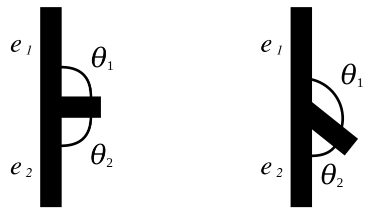


Figure 3.25: Calculating the perpendicular tick criterion

The example shown in Figure 3.24 also illustrates a problem with regards to labelling nodes on corners of lines. Label 4 is positioned at a point where the line changes direction and as such it is impossible to draw the tick so that it is perpendicular to both parts of the line. It could be possible to draw the tick so that it is as close as possible to being perpendicular to both parts of the line. However, this would require greater flexibility in the positioning of the label. In practice, existing maps avoid this problem either by ensuring that stations are only ever drawn on straight parts of lines (as on the London Underground map), or by not restricting the label position as much as we are, or by not using ticks as station icons.

Although we have called this criterion the perpendicular tick criterion, it still has merit when drawing maps that use different types of station icons. In the extreme case of using a circle as the station icon, the orientation of the icon is irrelevant. However, the position of the label is still important—it is desirable to keep the label as close to the icon as possible but still a reasonable distance from the line.

Calculation of the perpendicular tick criterion is fairly straightforward. Figure 3.25 shows two examples of nodes being labelled—the left-hand example has a label drawn to the east and the right-hand example has a label being drawn to the south-east. In both examples, e_1 and e_2 are the edges and θ_1 and θ_2 are the angles between the tick and e_1 and e_2 respectively. The unweighted value of this criterion for a single node, v, is the absolute difference between the two angles:

$$c_v = |\theta_1 - \theta_2| \tag{3.11}$$

The total value for all nodes, V, in the graph is therefore

$$c = \sum_{v \in V} |\theta_1 - \theta_2| \tag{3.12}$$

In the examples in Figure 3.25, the value of c for the left-hand example will be zero as both θ_1 and θ_2 are equal at 90°. This indicates that the tick is positioned to be perpendicular to the edges e_1 and e_2. In the right-hand example, the value of c is $135° - 45° = 90°$. This shows that the label is not perpendicular to either e_1 or e_2.

3.9.7 Label Criteria Weightings

Each criterion is independently weighted depending on the importance associated with that criterion. The values for weightings were determined through experimentation with various examples. Table 3 shows a typical set of weightings for each of the labelling criteria. As with the node criteria, it is very difficult to find a set of weightings that balance out in many cases. It is an inherent problem with optimization algorithms that local minima exist in the search space, so no one set of weightings would be applicable in every single case.

[★ Put some examples of weightings here - Table 3.6, Figure 3.26]

Figure 3.26: Simple example of label criteria for a complete graph. The values for each criterion for this graph are shown in Table 3.6.

Table 3.6: Typical label position criteria weightings.

Criterion c_n	Weighting w_n
c_1	
c_2	
c_3	
c_4	
c_5	
c_6	

Chapter 4

Clustering and Partitioning

When experimenting with our initial set of criteria for moving individual nodes we encountered many problems where we would ideally like to be able to move several nodes together at the same time. We needed to find some way of clustering the graph into groups of nodes that can be moved together and will have some chance of improving the graph with respect to the weighted node movement criteria. We used three different methods by which to find clusters of nodes:

- clustering based on overlength (or underlength) edges
- clustering based on bends in lines
- clustering based on partitioning the graph into two parts that can be moved closer together

Once clusters have been identified, they were moved in exactly the same way that individual nodes were moved with the only difference being that rather than moving one node at a time, all the nodes in the cluster are moved. Indeed, it was actually possible to treat moving single nodes as moving clusters with only one node.

This chapter will discuss some of the problems that caused us to investigate moving groups of nodes and then detail the process by which we identified clusters in the graphs using the three methods listed above.

4.1 Clustering Overlength Edges

4.1.1 Why Cluster Overlength Edges?

The biggest problem that we encountered when experimenting with our node movement criteria was that of long edges that refused to reduce in length. We define overlength edges as being edges which are longer than lg where g is the grid spacing and l is the preferred multiple of grid spacings for an edge. Figure 4.1(a) shows such an example where edge AB is too long. The optimal position for nodes B, C and D is as shown in (d). However, if we only allow one node to move at a time, it is impossible for this optimum to be found. Take the case of trying to move node B towards A as shown in (b): while this is a slightly more optimal location with regards to the length of edges AB and BC, it intorduces another turn in the line $ABCD$ and edge BC is no longer at 45° diagonal. This would increase the line straightness and octilinearity criteria in this new position for B.

One way to approach this problem without requiring any particular modifications to our method would be to increase the weighting for the edge length criterion so that the new position for B is more optimal even considering the increase in line straightness and octilinearity criteria. However, for this to be the case, the edge length weighting must be increased by several orders of magnitude to the point where the length of edges completely overwhelms all of the other criteria.

Another idea would be to impose some other kind of weighting based on the graph theoretic distance that a node is from the "centre" of the graph. This extra

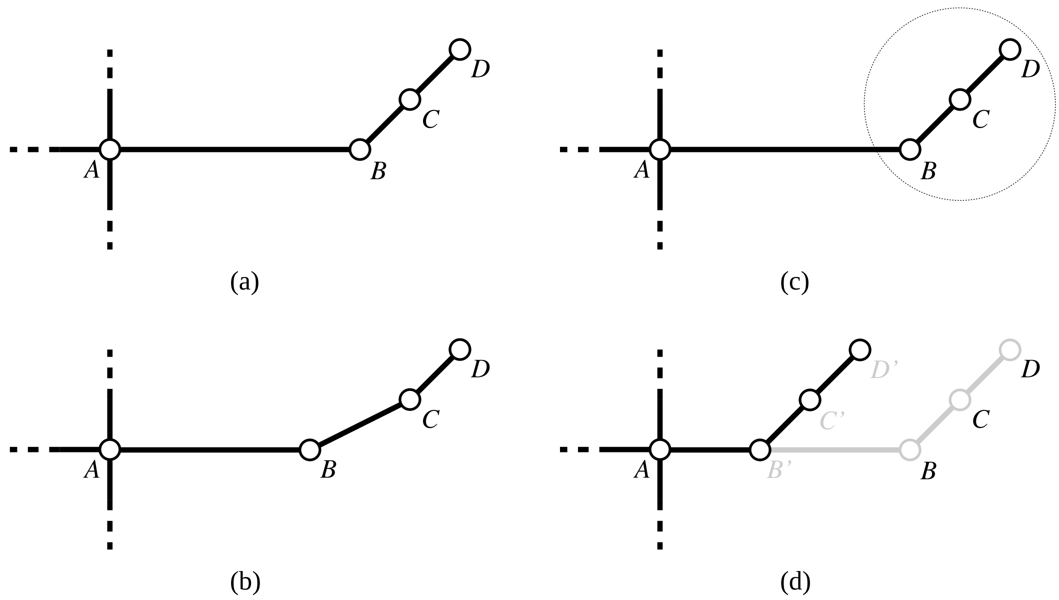


Figure 4.1: Clustering overlength edges. The edge AB is too long (a), but it is not possible to reduce the length of this edge (b) without moving nodes B, C and D (c) at the same time (d).

weighting would apply to individual nodes and edges and would decrease as the graph theoretic distance from the centre of the graph increased. In the example in Figure 4.1, the length of edge AB would contribute to the edge length criterion to a greater extent than the edge BC. It would then be possible to potentially move B closer to A without having to increase the overall weighting for the edge length criterion to such a great extent.

However, both of these solutions still rely on the movement of individual nodes. In our example, we could well need three iterations to allow all three nodes B, C and D to move into the optimal positions as shown in (d). It is also undesirable to have to significantly increase criteria weightings in order to overcome local minima as this could well have a detrimental effect on the rest of the graph.

4.1.2 Identifying Clusters Based On Overlength Edges

Our first implementation of an algorithm to find clusters of nodes only considered clusters separated by a single overlength edge. We later extended our algorithm to handle clusters separated by multiple overlength edges–this enhancement is discussed in the following section.

Our algorithm works by finding the partitions in the graph that would be created if a single overlength edge was cut. It can be summarised as follows:

1. Find the set of edges $\Phi \subset E$ such that the length of each edge $e \in \Phi > lg$.

2. For each edge $e \in \Phi$ being connected to nodes X and Y:

   (a) Perform a breadth-first search of the graph starting from X but not following e. We can determine if there is a cluster of nodes separated by the single edge e if we have exhausted the search for nodes without encountering Y. If at any point in the search from X we find Y, then the search should be terminated.

   (b) Repeat the previous step starting from Y and terminating the search if we encounter X.

For each cut made in the graph we only need to remember the partitions with fewest nodes (or an arbitrary partition if both partitions are of equal size). The smaller of the two paritions would require fewer criteria calculations when potential locations for the nodes are being tested.

★ More formal definition of the above?:

The overlength edges cut the set of nodes V into two subsets S and T such that the edge $(u, v) \in E$ has $u \in S$, $v \in T$ and $S \cap T = \emptyset$.

★ Illustrate with a worked example and figures.

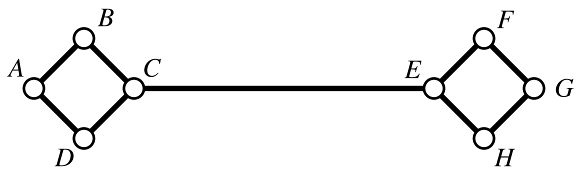


Figure 4.2: Example of clustering with a single overlength edge. An attempt to cut the graph based on the edge CE will produce two clusters of nodes: $\{A, B, C, D\}$ and $\{E, F, G, H\}$

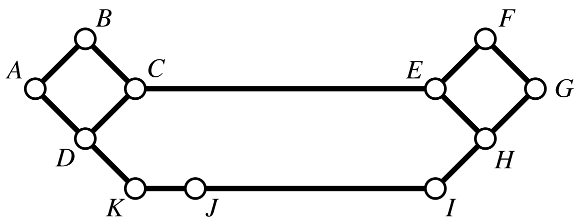


Figure 4.3: Example of failed clustering with a single overlength edge. In this case, cut the graph based on the edge CE will not find two clusters of nodes.

To illustrate this algorithm, consider the graph in Figure 4.2. In this graph just one edge, CE, is overlength and we wish to find the two clusters of nodes created by cutting the graph at this edge. Performing the depth-first search from node C will find the four nodes to the left of CE giving the cluster $\{A, B, C, D\}$. Similarly searching from E will find the cluster $\{E, F, G, H\}$.

Now consider the graph shown in Figure 4.3. In this case, an extra loop has been added between nodes D and H. As before, we wish to try cutting the graph at the overlength edge CE. Attempting the depth-first search from C results in potentially eight iterations (shown in Table 4.1), but our algorithm would stop at the seventh iteration as we would be adding the node at the opposite end of the edge CE, E, to the cluster. The loop from D to H causes this to happen, and as can be seen from the eighth iteration, would result in the entire graph being treated as a single cluster. This would clearly not achieve the aim that we were intending by attempting to cluster the graph based on a single multiple edge.

Graphs as in Figure 4.3 can be clustered, but we must take into account cases of multiple overlength edges. This is discussed in the next section.

Table 4.1: Depth-first search of the graph in Figure 4.3, starting at node C.

Search Iteration	Next Node(s)	Cluster	Remaining Nodes
	$\{\}$	$\{\}$	$\{A, B, C, D, E, F, G, H, I, J, K\}$
1	$\{C\}$	$\{C\}$	$\{A, B, D, E, F, G, H, I, J, K\}$
2	$\{B, D\}$	$\{B, C, D\}$	$\{A, E, F, G, H, I, J, K\}$
3	$\{A, K\}$	$\{A, B, C, D, K\}$	$\{E, F, G, H, I, J\}$
4	$\{J\}$	$\{A, B, C, D, J, K\}$	$\{E, F, G, H, I\}$
5	$\{I\}$	$\{A, B, C, D, I, J, K\}$	$\{E, F, G, H\}$
6	$\{H\}$	$\{A, B, C, D, H, I, J, K\}$	$\{E, F, G\}$
7	$\{E, G\}$	$\{A, B, C, D, E, G, H, I, J, K\}$	$\{F\}$
8	$\{F\}$	$\{A, B, C, D, E, F, G, H, I, J, K\}$	$\{\}$

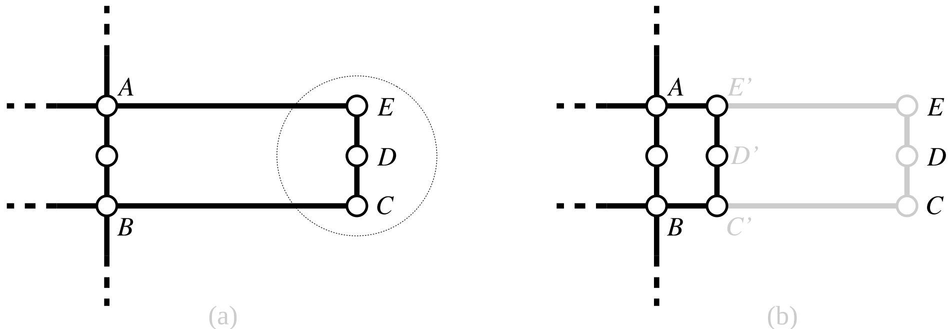


Figure 4.4: Clustering multiple overlength edges. The edges AE and BC are too long (a) and it is only possible to reduce the length of these edges by moving nodes C, D and E at the same time (b).

4.1.3 Multiple Overlength Edges

There are a number of cases when finding single overlength edges with with to partition the graph isn't enough. Figure 4.4(a) shows such an example where the cluster of C, D and E is separated from the rest of the graph by the overlength edges AE and BC. It would be better if the cluster could be moved to the new positions as shown by C', D' and E' in (b).

The algorithm presented in the previous section is unable to handle such circumstances. It relies on the overlength edge being the only edge to be cut in order to partition the graph into two disjoint subgraphs. This was illustrated using the

Table 4.2: Breadth-first search of the graph in Figure 4.3 taking multiple overlength edges into account, starting at node C.

Search Iteration	Next Node(s)	Cluster	Remaining Nodes
	$\{\}$	$\{\}$	$\{A, B, C, D, E, F, G, H, I, J, K\}$
1	$\{C\}$	$\{C\}$	$\{A, B, D, E, F, G, H, I, J, K\}$
2	$\{B, D\}$	$\{B, C, D\}$	$\{A, E, F, G, H, I, J, K\}$
3	$\{A, K\}$	$\{A, B, C, D, K\}$	$\{E, F, G, H, I, J\}$
4	$\{J\}$	$\{A, B, C, D, J, K\}$	$\{E, F, G, H, I\}$
Search stops here as edge IJ is overlength.			

example in Figure 4.3. We now need to extend this to take account of having multiple edges to cut to create two disjoint subgraphs.

To extend our existing algorithm, first recall that the existing termination condition was whether we found the node at the opposite end of the edge to that which we started from. Other that that, we could end up searching the entire graph. We now add a new condition to the algorithm:

- If at any point during the search another overlength edge is encountered, do not search beyond that edge.

Illustrating this with the example in Figure 4.3 and trying to cut the graph at the same point (edge CE) now produces a different result. The iterations of the search starting at node C are shown in Table 4.2. The search starts off as before but will not search beyond node J as the edge IJ is overlength.

A similar search can be performed starting at the other end of edge CE, E. This results in a cluster of nodes consisting of $\{E, F, G, H, I\}$. Note that an attempt to cluster the graph by cutting edge IJ will, in this case, produce exactly the same two clusters.

However, circumstances exist where it remains impossible to find two distinct clusters. An example of such a scenario is shown in Figure 4.5. This graph is

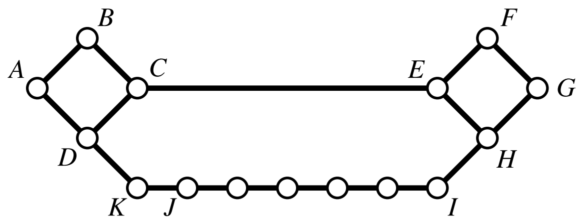


Figure 4.5: Example of failed clustering with overlength edges.

much the same as the one in Figure 4.3 except that the overlength edge IJ has been replaced by nodes and edges such that no overlength exists in this part of the graph. The overlength edge CE remains, but it should be fairly obvious that trying to cut this edge will result in a similar outcome as before other multiple overlength edges were added as a condition to the search algorithm. The search will follow the loop from D to H before encountering the other end of the edge CE. Having said this, in this example, it would not be possible to make in improvement such that the edge CE could be reduced in length. This is therefore an example of a local minimum that can't be improved using our multicriteria optimisation method.

4.2 Clustering Non-straight Lines

4.2.1 Why Cluster Non-straight Lines?

Related to the problem of overlength edges (discussed in the previous section) is the problem of lines that contain short deviations or kinks. Figure 4.6 shows examples of the kinds of common scenarios where these deviations become apparent. There are two main explanations as to why these deviations are so common:

1. The deviations become manifest when fitting a slightly off-straight line to the grid (as shown in (a) and (b) in Figure 4.6).

2. Three nodes are too close together to fit onto the grid without the middle

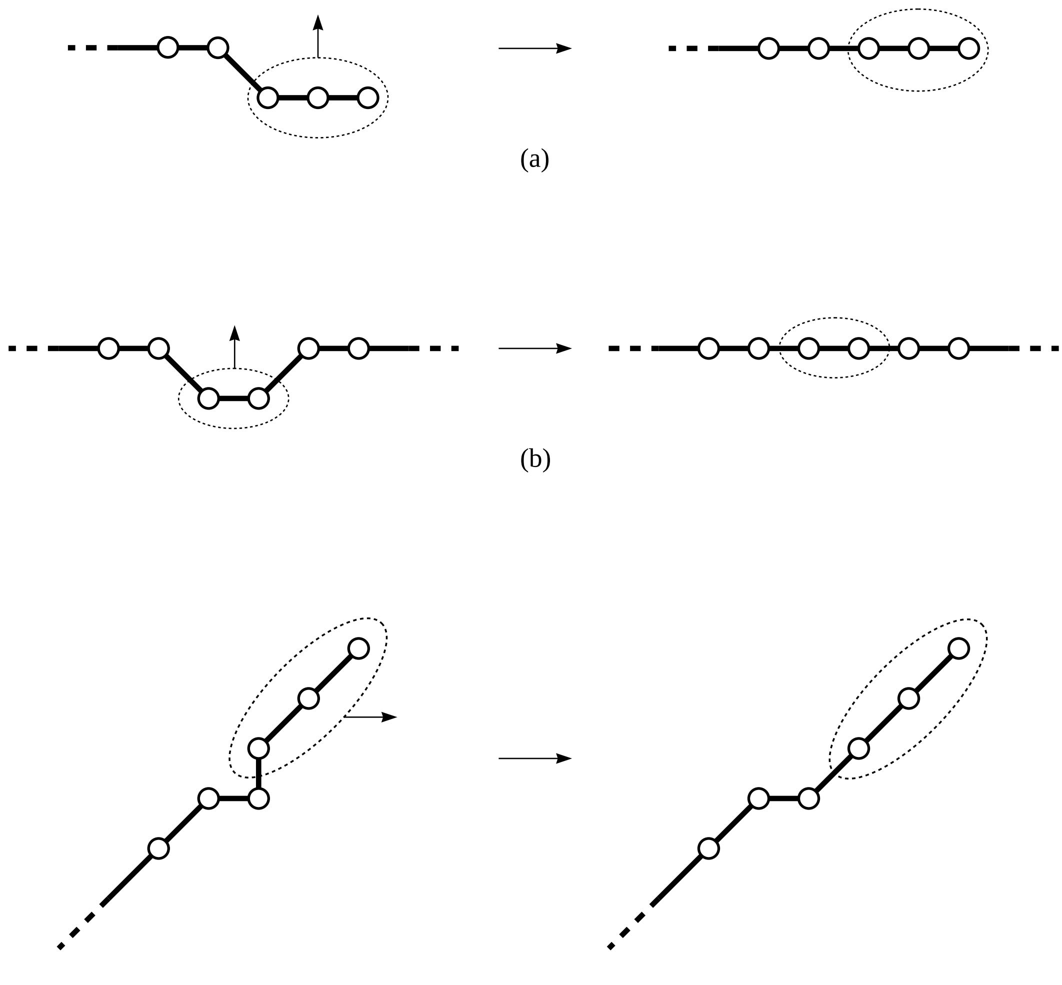


Figure 4.6: Finding non-straight lines. Potential clusters are highlighted by dashed areas. The right-hand side of (a), (b) and (c) show the result of moving the indicated clusters of nodes to a more optimal location. In the case of (c), the resulting movement of the cluster to the right introduces the possibility of reclustering in a similar manner to (a).

node being offset relative to the rest of the line (as shown in (c) in Figure 4.6.

The line straightness criterion (Section 3.6.5) helps in the case where a single node can be moved into a vacant grid intersection. For example, if the cluster indicated in Figure 4.6(a) contained only one node, this scenario can be catered for in our single-node-movement method. We would like to be able to apply the line straightness criterion to scenarios where more than one node should be moved simultaneously. As such, we need some way of being able to identify the cluster of nodes to be moved as one.

4.2.2 Identifying Clusters Based on Non-straight Lines

Our method for identifying clusters of nodes that can be moved to improve non-straight lines is very straightforward. Figure 4.7 shows a simple example that will be used to illustrate how these clusters are found. First off, as we are only considering improving deviations in lines, we only need to look at nodes which have exactly one or two neighbours. This means that nodes E and F are discounted for forming part of a cluster right from the outset (and could even be removed from the graph while we are searching for clusters). Clusters are then identified by finding the minimum set of nodes which are connected by edges which are parallel. In the example, this produces five clusters of two nodes and one cluster of three nodes: $\{BC\}$, $\{CD\}$, $\{DE\}$, $\{GH\}$, $\{HI\}$ and $\{IJK\}$.

4.3 Partitioning

Clustering overlength edges works quite well for simple scenarios and certainly in cases where there is just a single overlength edge separating clusters of nodes. However, some more complex situations can arise where a visual inspection of the graph would highlight cases where significant improvements to the visual

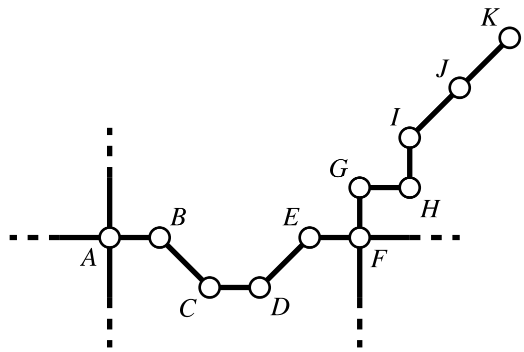


Figure 4.7: Clustering nodes to find non-straight lines. Ultimately, six clusters will be identified in this graph: $\{BC\}$, $\{CD\}$, $\{DE\}$, $\{GH\}$, $\{HI\}$ and $\{IJK\}$.

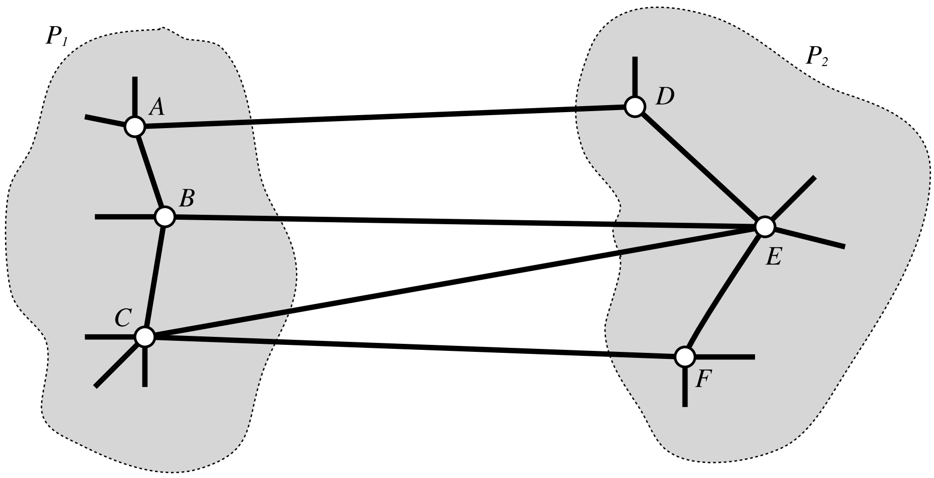


Figure 4.8: Partitioning a graph into two. The edges AD, BE, CE and CF are all too long but the only way of shortening them is to move either the left-hand partition P_1 or the right-hand partition P_2. Both P_1 and P_2 contain other overlength edges that would stop the method for clustering based on overlength edges from finding these partitions.

aesthetics of the graph can be made by partitioning the graph into two and moving one of these partitions. These partitions usually can't be discovered using the method for finding clusters by multiple overlength edges (Section 4.1.3) because the partition itself will more than likely contain more overlength edges. It would therefore be highly unlikely that the graph would be partitioned into exactly two clusters using the multiple overlength edges clustering method.

Figure 4.8 shows a simple example of partitioning a graph into two halves, P_1 and P_2. There are four edges in the graph that we would like to identify as separating the graph into two partitions: AD, BE, CE and CF. However, the method for finding clusters based on overlength edges is not likely to find P_1 and P_2 as both of these partitions contain other edges that are overlength. If we are able to identify P_1 and P_2, we can move one of these partitions as a cluster, causing the edges AD, BE, CE and CF to shorten.

Our approach to finding partitions in the graph can be summarised as follows:

1. Preprocess the graph to remove edge crossings, unwanted multiple edges and trailing edges. Edge crossings must be removed as the graph must be planar in order to find the faces. If there are any edge crossings, they can be replaced by dummy nodes for the duration of this process. Multiple edges (where there are several edges between two nodes) should be removed or at least considered as a single edge as this would otherwise affect the ability to identify faces. Trailing edges don't necessarily have to be removed at a preprocessing stage as they can be discovered and ignored as a by-product of the process of finding faces.

2. Find the set of faces (discussed in Section 4.3.1).

3. Use the set of faces to derive the dual graph (discussed in Section 4.3.2).

4. Diminish the dual graph to remove unnecessary edges (discussed in Section 4.3.3).

5. Cut the dual graph by finding edges which are *most opposite* each other (discussed in Section 4.3.4).

The following sections as indicated above detail the process of finding all the available partitions in a planar graph.

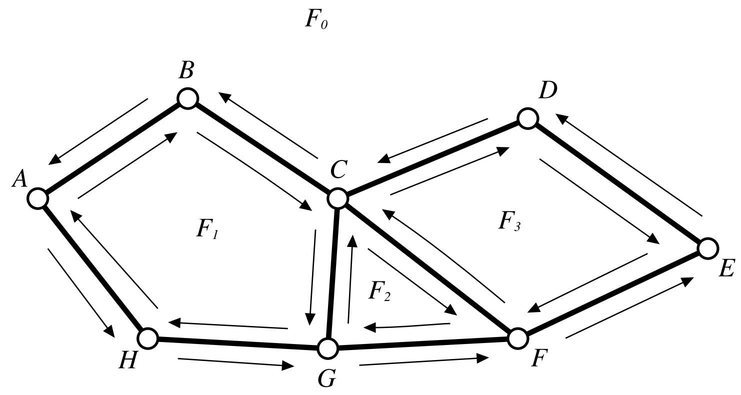


Figure 4.9: Finding faces. Arrows indicate the direction of traversal of each edge, for example, an arrow from A to B represents the from-to traversal $\overrightarrow{AB}$.

4.3.1 Finding Faces

★ Cycles. Face defined by a cycle that encloses a region in the planar graph.

★ Ignore multiple edges.

★ Explicitly ignore trailing edges.

★ Aim is to ensure that every edge in the graph is fully traversed.

★ An edge AB can be traversed *to-from*: $\overrightarrow{AB}$ or *from-to*: $\overleftarrow{AB}$.

★ An edge is *fully traversed* if it has been traversed both to-from and from-to.

★ To find a face F, get an edge that hasn't been fully traversed. Follow that edge either to-from or from-to by consistently taking the anticlockwise-most edge at each node adding each edge to F. Stop when the first edge has been reached again.

$F_0 = \{\overleftarrow{AB}, \overrightarrow{AH}, \overleftarrow{GH}, \overleftarrow{FG}, \overleftarrow{EF}, \overleftarrow{DE}, \overleftarrow{CD}, \overleftarrow{BC}\}$

$F_1 = \{\overrightarrow{AB}, \overrightarrow{BC}, \overrightarrow{CG}, \overrightarrow{GH}, \overleftarrow{AH}\}$

$F_2 = \{\overrightarrow{CF}, \overrightarrow{FG}, \overleftarrow{CG}\}$

$F_3 = \{\overrightarrow{CD}, \overrightarrow{DE}, \overrightarrow{EF}, \overleftarrow{CF}\}$

★ Talk about implicit trailing edge removal.

4.3.2 Deriving the Dual Graph

★ Complete dual graph G^* of planar graph G is the graph with nodes representing faces and with nodes being adjacent iff the corresponding faces in G are adjacent (share an edge). The number of edges between any two adjacent nodes F_1 and F_2 in G^* indicates the number of edges along the interface between F_1 and F_2.

★ *Outside edges* are those edges in G^* that are connected to the node that represents the outside face F_0.

★

★

★

4.3.3 Diminishing the Dual Graph

★ Remove edges from G^2 which represent faces that are adjacent by way of an edge that is $<= 1g$ long.

★

★

★

★

4.3.4 Finding Partitions

We considered a number of different strategies for partitioning the graph into two halves. All of these methods involved finding a path through the dual graph, G^*. A cycle in G^* can be considered as a cut through the planar graph G. In cutting G, we clearly need to find a path in G^* that joins two outside edges in G–in other words, a cycle in G^* that starts at the outside face F_0, passes through some other nodes in G^* and returns to F_0. Figure 4.10 shows an example of a graph,

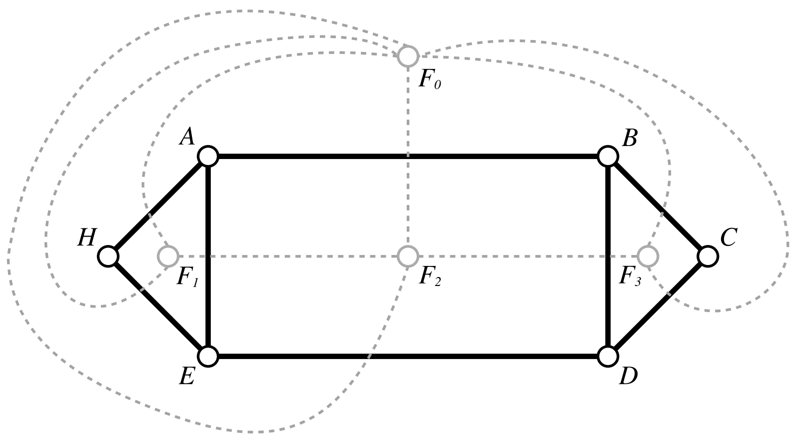


Figure 4.10: Example of a planar graph, G, and the complete dual graph, G^*. The planar graph $G = \{A, B, C, D, E, H\}$ is shown in black and the dual graph, $G^* = \{F_0, F_1, F_2, F_3\}$, is shown in grey with dotted edges.

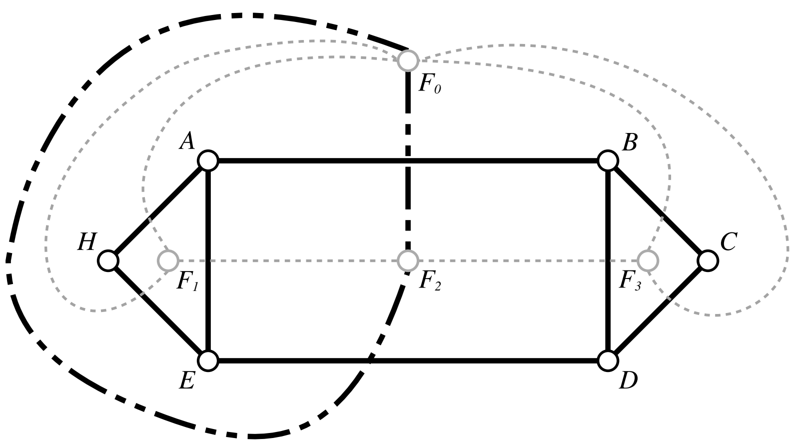


Figure 4.11: Example of a cycle in the dual graph equating to a cut in the planar graph represented by the dual graph. The cycle $\{F_0, F_2, F_0\}$ in the dual graph G^* (shown in bold) equates to the cut $P = \{AB, DE\}$ in the planar graph G.

$G = \{A, B, C, D, E, H\}$, and its complete dual graph, $G^* = \{F_0, F_1, F_2, F_3\}$. An illustration of a potential cut in this graph is shown in Figure 4.11, starting at F_0, passing through F_1 (cutting edge AB) and back to F_0 (cutting edge DE). In this case, $P = \{AB, DE\}$.

The cut must be chosen wisely with some thought as to how the partitions will respond to the movement criteria. If a cut is chosen which includes many edges of differing orientation, it is likely to be very difficult for either partition of the graph to move relative to each other. This becomes clear when you consider that when a partition is moved, some edges of the cut may end up being longer than before and other edges may end up becoming shorter. A more reliable approach would be to ensure that the cut contains as many edges as possible that can be reduced in length. So, rather than have edges of differing orientation, we try to ensure that the cut includes edges which ideally have the same orientation. This means finding a set of edges for a cut which are parallel to each other—or at least as parallel as possible. [★ Does this need illustrating with a figure and small example?]

Dual Graph Cycles

An obvious choice when looking for cycles in the dual graph is to use some shortest path algorithm [★ Such as?] to find paths through the dual graph. In order to be able to use such an algorithm it is necessary to know the starting point (i.e. which outside edge we start the cut from), the finishing point (i.e. which outside edge we finish the cut at) and some method of calculating the distance between each pair of adjacent edges in the cut (i.e. the weight on each edge in the dual graph to use when finding the shortest path).

The starting point for a cut can be any outside edge, e_0. Ultimately, every outside edge will be considered to find all the potential cuts in the graph. The finishing point is another outside edge–more specifically, the edge which is "most opposite" e_0. The concept of which edge is most opposite another edge is dealt with in the next section.

The weight on each edge in the dual graph can be calculated in a number of ways. [★ This is the hard bit! Do I need to show some example of distances between edges? None of them are very good, but I do need to show why I discounted a 'simple' shortest-path algorithm.]

★ Figure showing why shortest path isn't ideal. Figure showing why straightest path isn't ideal. Shortest path to which node - need to know where to go.

Entry and Exit Edges

In the process of finding a cut through a graph, we use the terms *entry edge* and *exit edge* to refer to the outside edges where the cut starts and finishes. Entry and exit edges also apply during the process of finding a cut, but in this case we are trying to find the entry and exit edges for each face that the cut passes through. This section is concerned with finding the entry and exit edges for a particular face.

There are a numbe of naïve approaches that can be taken to find an exit edge (such as taking the edge which is furthest away from the entry edge either geometrically or by graph theoretic distance). However, the whole purpose of finding an exit edge is that a suitable cut can be formed in the graph to allow the graph to be partitioned into two. As discussed earlier, this means that it is necessary for the cut to contain edges which are parallel or nearly parallel to each other. A naïve approach will not take this into account thereby leading to cuts containing edges with many different orientations and with not much likelihood of being mostly parallel.

Our approach is more considered in that it takes into account the direction of each edge in a cycle as well as the orientation of each edge in relation to the entry edge. It is therefore more likely that the set of edges forming the cut are not only parallel but also more likely to form a straight cut through the graph. In the task of finding an exit edge, we first assume that the entry edge, e_0 is known and that the set of edges that describe the cycle forming the edge of a face, F, is also known.

We have to consider each edge in F as being directed such that the direction

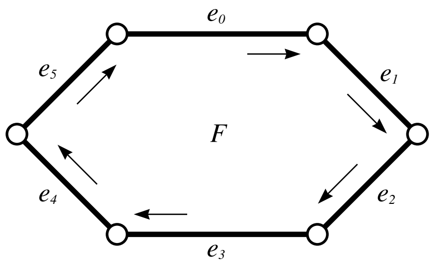


Figure 4.12: Finding the exit edge.

should follow around the cycle. The choice whether to follow the direction around a face clockwise or anticlockwise is arbitrary, but that choice must be consistent. Figure 4.12 shows a cycle where the clockwise direction of each edge has been marked with an arrow.

Once we have determined the direction of each edge in F, we calculate the normalised unit vector, $\boldsymbol{E}_n$, that represents the direction of each edge, e_n:

$$
\begin{aligned}
e_0 : \boldsymbol{E}_0 &= \begin{bmatrix} 1 \\ 0 \end{bmatrix} & \qquad e_1 : \boldsymbol{E}_1 &= \begin{bmatrix} 1 \\ -1 \end{bmatrix} \\
e_2 : \boldsymbol{E}_2 &= \begin{bmatrix} -1 \\ -1 \end{bmatrix} & \qquad e_3 : \boldsymbol{E}_3 &= \begin{bmatrix} -1 \\ 0 \end{bmatrix} \\
e_4 : \boldsymbol{E}_4 &= \begin{bmatrix} -1 \\ 1 \end{bmatrix} & \qquad e_5 : \boldsymbol{E}_5 &= \begin{bmatrix} 1 \\ 1 \end{bmatrix}
\end{aligned}
$$

We then calculate the sum of the entry edge vector, $\boldsymbol{E}_0$, and each other edge vector and find its magnitude. The exit edge is determined by selecting the minimum of these magnitudes. Comparing the edges in this manner means that any edge which is on the "other side" of F and also parallel to e_0 is considered as an exit edge.

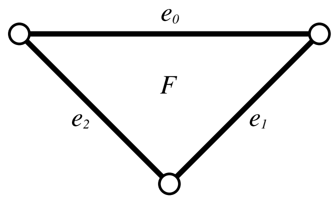


Figure 4.13: Finding the exit edge in a triangular cycle. Both edges e_1 and e_2 are candidates for being the exit edge but only one can be selected.

$$|\boldsymbol{E}_0 + \boldsymbol{E}_1| = \left|\begin{bmatrix}1\\0\end{bmatrix} + \begin{bmatrix}1\\-1\end{bmatrix}\right| = \left|\begin{bmatrix}2\\-1\end{bmatrix}\right| = \sqrt{5}$$

$$|\boldsymbol{E}_0 + \boldsymbol{E}_2| = \left|\begin{bmatrix}1\\0\end{bmatrix} + \begin{bmatrix}-1\\-1\end{bmatrix}\right| = \left|\begin{bmatrix}0\\-1\end{bmatrix}\right| = 1$$

$$|\boldsymbol{E}_0 + \boldsymbol{E}_3| = \left|\begin{bmatrix}1\\0\end{bmatrix} + \begin{bmatrix}-1\\0\end{bmatrix}\right| = \left|\begin{bmatrix}0\\0\end{bmatrix}\right| = 0$$

$$|\boldsymbol{E}_0 + \boldsymbol{E}_4| = \left|\begin{bmatrix}1\\0\end{bmatrix} + \begin{bmatrix}-1\\1\end{bmatrix}\right| = \left|\begin{bmatrix}0\\1\end{bmatrix}\right| = 1$$

$$|\boldsymbol{E}_0 + \boldsymbol{E}_5| = \left|\begin{bmatrix}1\\0\end{bmatrix} + \begin{bmatrix}1\\1\end{bmatrix}\right| = \left|\begin{bmatrix}2\\1\end{bmatrix}\right| = \sqrt{5}$$

As can be seen from these calculations, if the entry edge in Figure 4.12 is e_0, the corresponding exit edge is e_3 as the magnitude of the resulting vector is 0. This can clearly be seen in Figure 4.12 as e_3 is both parallel and has opposite direction to the entry edge e_0.

In the case where there are more than one potential exit edge, a choice has to be made as to which one should be used. The triangular cycle shown in Figure 4.13 poses such a dilemma.

$$e_0 : \boldsymbol{E}_0 = \begin{bmatrix}1\\0\end{bmatrix} \qquad e_1 : \boldsymbol{E}_1 = \begin{bmatrix}-1\\-1\end{bmatrix}$$

$$e_2 : \boldsymbol{E}_2 = \begin{bmatrix}-1\\1\end{bmatrix}$$

It is obvious by inspecting the cycle that taking edge e_0 as the entry edge will result in both edges e_1 and e_2 being candidates for the exit edge. This visual inspection is reinforced by calculating the magnitude of the sum of the vectors as follows:

$$|\boldsymbol{E}_0 + \boldsymbol{E}_1| = \left| \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 \\ -1 \end{bmatrix} \right| = \left| \begin{bmatrix} 0 \\ -1 \end{bmatrix} \right| = 1$$
$$|\boldsymbol{E}_0 + \boldsymbol{E}_2| = \left| \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 \\ 1 \end{bmatrix} \right| = \left| \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right| = 1$$

We have three potential strategies for resolving this dilemma:

1. *Make an arbitrary decision.* It doesn't matter which candidate exit edge we choose as both edges are equally preferable.
2. *Take the edge which is furthest away geometrically.* The geometric distance between the midpoints of the entry edge and each candidate exit edge is calculated and the one which is furthest away from the entry edge is selected as the exit edge.
3. *Take the edge which has greater graph-theoretic distance from the entry edge.* The minimum graph theoretic distance from the entry edge (remembering to follow the cycle both ways) to each candidate exit edge is found and the edge which is furthest away from the entry edge is selected as the exit edge.

Each of these strategies will fail and succeed in differing circumstances. In practice, the choice as to which strategy is used seems to be largely irrelevant which means that we used the least computationally expensive option and just made an arbitrary selection of the exit edge from the candidates.

Cutting Many Faces

We can now apply the method for finding entry and exit edges of a face discussed in the previous section to a graph containing several faces. Recall that in finding a cut in the graph we are trying to find a cycle in the dual graph, starting from the node that represents the outside face, F_0, passing through some inside faces and returning to F_0.

The cutting algorithm to find the set of edges in a cut, Φ, for a graph $G = \{V, E\}$ proceeds as follows:

1. Find the set of outside edges $E_{\text{out}} \subset E$.
2. For each outside edge $e_0 \in E_{\text{out}}$:
   (a) Add e_0 to Φ.
   (b) Treat e_0 as the entry edge into the next inside face F.
   (c) Find the exit edge, e_{exit} of F.
   (d) If $e_{\text{exit}} \in E_{\text{out}}$, add e_{exit} to Φ and terminate as we have reached the outside face again.
   (e) Otherwise, treat e_{exit} as e_0 and repeat step 2.

However, there is a particular circumstance which exists that could potentially lead to the algorithm never terminating. The way in which the exit edge is determined when cutting the graph can lead to situations where an infinite loop can occur. Such a situation is illustrated in Figure 4.14. In this example, the initial starting edge is e_0 and the cut continues through faces F_1, F_2 through to F_8. At this point, the cutting algorithm would take us back through F_1 and on in to F_2—thereby resulting in a loop that never finds another outside edge. To resolve this simply, an extra terminating condition can be added to the cutting algorithm which causes it to stop if $e_{\text{exit}} \in E_{\text{out}}$ or $e_{\text{rm}} \in \Phi$.

As well as a non-terminating algorithm, Figure 4.14 also illustrates a case where the cut may not include edges which are parallel or mostly parallel. In fact, the cut illustrated almost includes no two edges with the same orientation! This

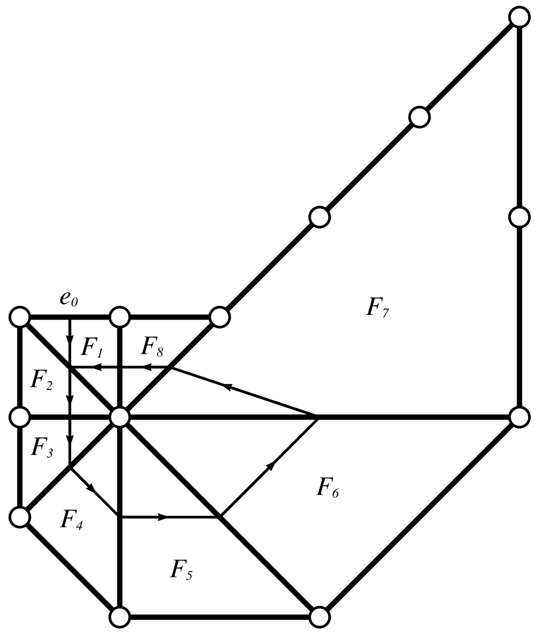


Figure 4.14: Non-terminating cutting algorithm. If the initial entry edge is e_0, the cut as indicated by the dotted lines with arrows will result in a perpetual loop in the cutting algorithm.

comes about because at each face we are not taking into account the orientation of the outside edge where the cut was started; just the current entry edge to that face is considered when finding the exit edge. We can modify the cutting algorithm subtly to improve this: instead of comparing each edge around a face with the vector of the entry edge to that face, we compare each edge with the vector of the outside edge where the cut was started. This gives the benefit of preferring much straighter cuts through the graph, as illustrated in Figure 4.15. Also notice that the cycle that previously stopped the algorithm from terminating is no longer apparent.

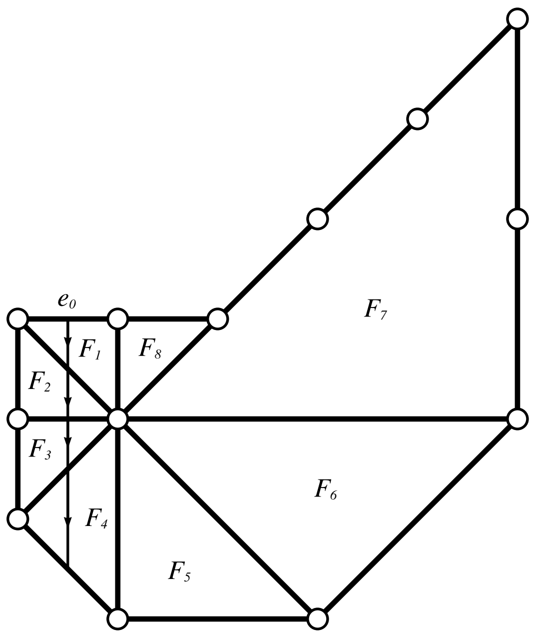


Figure 4.15: Modified cutting algorithm to remove the risk of an infinite loop in the cutting algorithm.

Chapter 5

Results and Discussion

Stuff

Chapter 6

Empirical Evaluation

6.1 Introduction

In the previous chapters, a number of aesthetic criteria for automatically drawing metro maps have been detailed, together with a method that combines these criteria in order to actually produce some automatically drawn metro maps. In order to validate these criteria and to show that the maps produced using this method are suitable, an empirical experiment involving a number of human subject performing route-planning tasks with a variety of different metro maps was devised.

6.2 Hypotheses

As introduced at the beginning of Chapter 3, a number of design principles have evolved throughout the history of the schematic metro map. When combined, these principles are thought to provide a map that is of greater quality than a map that doesn't follow the principles (for example, a geographic map of the metro system). Direct measurement of the quality of a map directly is hard (and is essentially what the multicriteria optimisation method aims to achieve!), so we instead intended to indirectly measure the quality by asking human participants to perform common route-planning tasks and comparing their speed and accuracy.

We have three hypotheses:

1. Maps drawn according to a metro map metaphor are better than those that are not with respect to the time that it takes to use the map for route-planning.

2. Maps drawn according to a metro map metaphor are better than those that are not with respect to the accuracy of finding the optimal route.

3. People "prefer" maps that are drawn according to a metro map metaphor over those that are not.

In order to be able to verify these hypotheses, we devised methods that will allow us to measure the accuracy and the time that it takes for someone to use a map for simple route-planning tasks, as well as a method to illicit the preference of one map over another. We chose two different types of questions that would be posed in order to ascertain the suitability of a map for route-planning:

1. How many changes are required to get from *Station X* to *Station Y*?

2. How many stations do I go through to get from *Station X* to *Station Y*?

These questions were chosen on the basis that they are likely to require the most common route-planning tasks when people use metro maps.

We used maps of six different real-life metro systems:

1. Atlanta Metropolitan Area Rapid Authority (MARTA) Rail Map [MAR06].

2. Bucharest Metro Map [Met06].

3. Mexico City *Sistema de Transporte Colectivo* Map [Mex06].

Table 6.1: Characteristics of metro maps used in the empirical evaluation.

Map	Stations	Lines
Atlanta	39	2
Bucharest	45	3
Mexico City	175	11
Stockholm	102	3
Toronto	70	4
Washington D.C.	85	5

4. Stockholm *Tunnelbana* Map [SL006].

5. Toronto Transit Commission (TTC) Subway/RT Route Map [TTC06].

6. Washington Metropolitan Area Transity Authority (WMATA) Metro System Map [WMA06].

These maps differ in characteristics and complexity from a fairly simple two-line, centralised network in the case of Atlanta through to a complex, highly-interconnected, decentralised map in the case of Mexico City. Table 6.1 gives details of the characteristics in terms of number of stations (nodes) and lines of each map. Three variants of each map were used: one of which was a geographic representation of the stations, the second was a representation of the published map and the third was drawn using the method described earlier in this thesis. We also used the Recife metro map as a training map so that subjects had time to familiarise themselves with the procedure.

One of the problems in using published maps is that they all use different graphic design approaches. We decided that all three maps would be drawn using the same graphic design to remove any question of the way that features such as stations and lines are drawn making a difference to the subjects' route planning tasks. To illustrate the three map variants, the geographic map (Figure 6.1), normalised published map (Figure 6.2) and automatically drawn map (Figure 6.3) are shown here. The official *Mapa de la Red* metro map for Mexico City is shown in Figure 6.4. All 18 maps as used in the evaluation are shown in Appendix B: Section B.1.1 (Atlanta), Section B.1.2 (Bucharest), Section B.1.3 (Mexico City), Section B.1.4 (Stockholm), Section B.1.5 (Toronto) and Section B.1.6 (Washington D.C.).

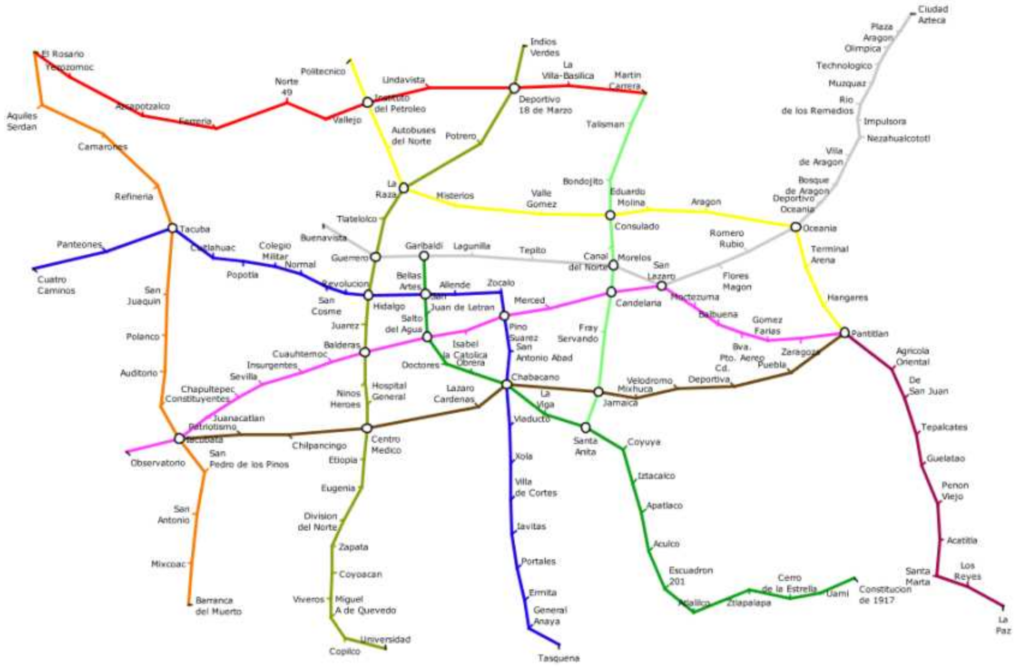


Figure 6.1: Mexico City geographic map.

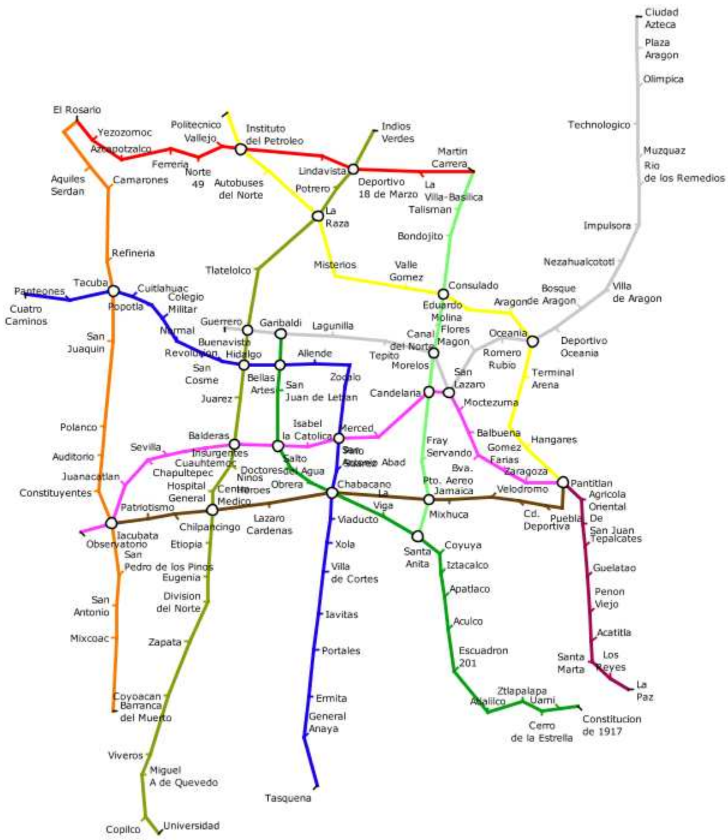


Figure 6.2: Mexico City normalised published map.

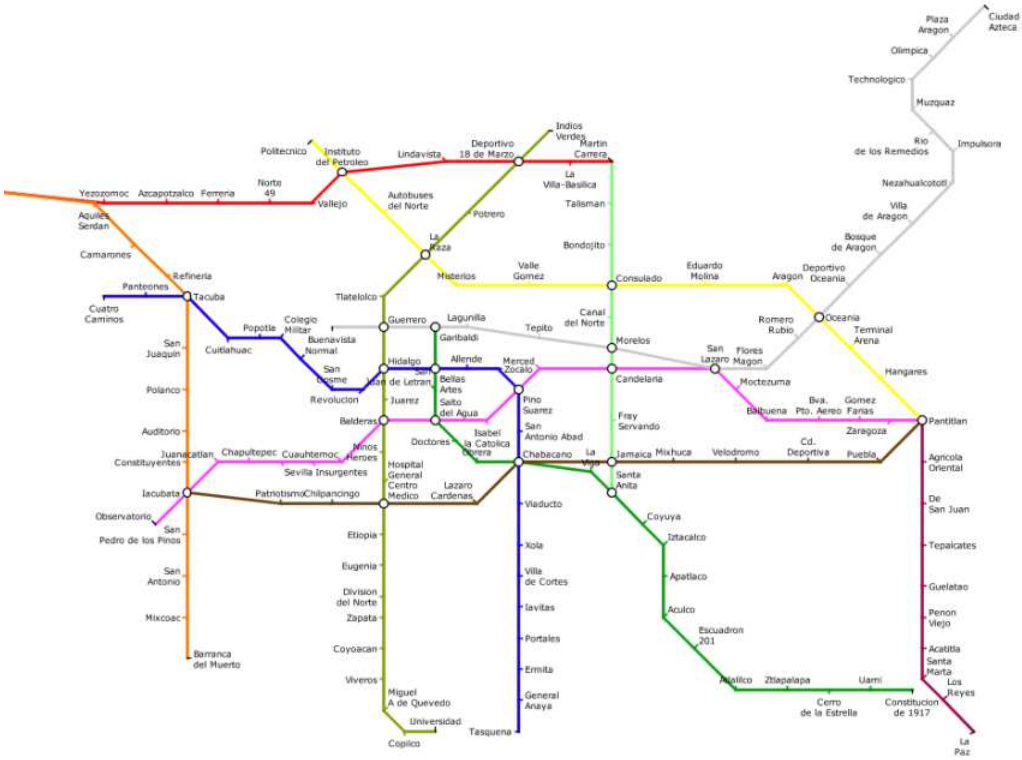


Figure 6.3: Mexico City automatically-drawn map.

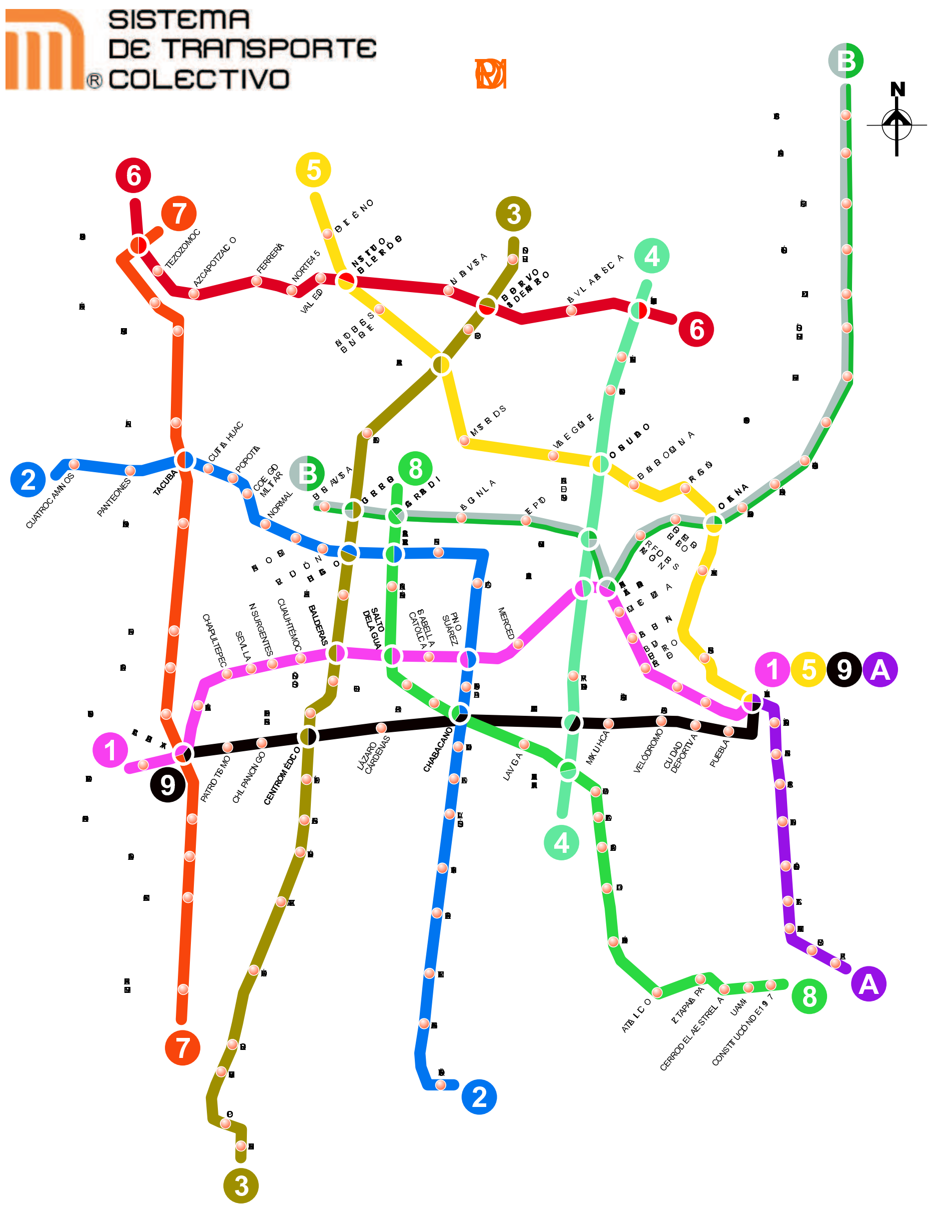


Figure 6.4: Mexico City *Mapa de la Red* official metro map.

Table 6.2: Number of subjects in each experiment group.

Group	Number of subjects
A	13
B	15
C	15

6.3 Methodology

We conducted a number of experiments over a two-week period. A total of 43 subjects participated, nearly all of whom were Computer Science undergraduates. [★ How is this representative?] As a number of experiments were run, we attempted to ensure that each experiment was as identical as possible in order to reduce any potential external influences. We split the set of subjects into three groups which we called A, B and C and ensured that each group contained roughly the same number of subjects. The number of subjects in each group is shown in Table 6.2.

Each group would receive exactly the same questions and in exactly the same order but they would only see one variant of each map per group for each question. For example, for question a using the Atlanta map, all subjects in all groups are performing the same task but group A will be doing this with the normalised published map, group B will use the geographic map and group C will use the automatically drawn map. The purpose of this was to ensure that the possibilities of learning the map between questions - while not completely eliminated - can at least be reduced as much as possible.

In order to conduct the experiment, a software application was written which ensured a controlled environment when showing the maps. The application took over the entire screen of the computer, ensuring that nothing else on the computer would distract the subjects. Interaction with the software during the tests was entirely with the mouse. Using a software application in this manner allowed

Figure 6.5: Screenshot of the software application used to conduct the empirical evaluation.

us to time each question in the experiment accurately and also to automatically see whether they answered correctly or incorrectly and to collate the results. It also allowed the environment to be more closely controlled and ensured uniformity amongst each subject. A screenshot of the software application is shown in Figure 6.5.

The following procedure was used for each experiment:

1. Subjects were directed to a specific computer. They were told not to touch anything until directed. Each computer in the room had previously been configured for a specific groups' questions.

2. A preliminary script was read out. This script introduced the subjects to the minimum amount of information that they needed in order to understand the questions asked of them in the experiment. To remove any potential

ambiguity, explanations of terminology used were provided in the script, including *Metro Map*, *station* and *line*. The script also mentioned that any stations joined by a line in a single colour are connected by direct services and that where two ore more lines pass through a single station it is possible to change from one line to another. The script included examples of questions that the subjects would be answering and introduced them to the application interface. To do this, two sample questions are worked by demonstrating the application to the subjects using an overhead projector. The script placed emphasis on the need to get questions correct and subjects were asked not to rush the questions. The script informed the subjects that the questions would be timed and were told during which parts of each question that timing was measured. The full script is shown in Appendix B, Section B.2.1.

3. The subjects were told when they could commence answering with the software application. Everyone started at the same time and were presented questions for 20 minutes. We decided to limit the total duration of the test so that everyone started and finished at the same time. This way we could avoid any anxiety that could be caused by one subject finishing before another. We chose 20 minutes as being a more-than-ample time in which to answer the minimum number of questions that we needed. If a subject managed to complete these questions they continued to be presented with questions until the 20 minutes was up. Each question proceeded in the following manner:

   (a) The map is shown together with the question. Timing of the subjects' answers started as soon as the map, question and potentials answers were shown.

   (b) The subject selects their answer from a list of five potential answers by clicking on a radio button. They can change their mind at this point but they must have selected an answer before they could continue.

   (c) When an answer is selected, the subject had to click on a button to continue. When this button was clicked, timing of that question stopped.

   (d) The subject then selected their perception of the level of difficulty from a list of five options from "very hard" to "very easy". The subjects were able to rest at this point before continuing to the next question.

   (e) The subject then clicks on another button in order to proceed to the next question.

4. After 20 minutes had elapsed, the subjects were presented with a screen than showed them how many questions they got correct or incorrect. For any questions that they got incorrect, they were shown the correct answer as well as the answer that they gave.

5. A second script was then read out which introduced the subjects to the questionnaire. The questionnaire was in two parts: the first involved viewing the three variants of each map on the overhead projector screen and them being asked to write down their preference for each variant from "most preferable" to "least preferable", the second involved answering four questions:

   (a) Have you seen any of the metro maps shown here before these tests? If so, which ones?

   (b) Which features of the metro map layout did you find most helpful when completing the tests?

   (c) Which features of the metro map layout did you find least helpful when completeing the tests?

(d) Did you find about of the questions ambiguous?

6. The subjects were then rewarded with £5 for their time and were allowed to leave.

6.4 Results

★ Statistical results here. Reference raw results in Appendix. Highlight significant/insignificant results.

★ Charts?

★ Preference for particular map.

★ Highlight responses in questionnaire.

★

6.5 Analysis

★

★

★

6.6 Conclusion

★ Which hypotheses are confirmed? Which hypotheses are unconfirmed?

★ Any future investigation to assess other hypotheses or unconfirmed hypotheses.

★

Chapter 7

Applications of the Metro Map Paradigm

7.1 Flexibility and Extensibility

Stuff.

7.2 Visualisation of Project Plans

At present, Gantt charts are predominantly used for the mapping of projects in organizations. Gantt charts show each task in a project as a horizontal bar, aligned against a time axis. The time axis allows temporal relationships between the tasks to be visualized. Each task bar is also annotated with the resources (people, teams of people, etc.) that are involved in that task. Dependencies of one task on another are shown by arrows pointing between the two tasks. [★ Refer to figure of Gantt chart]

While Gantt charts are effective for planning a project they are not effective for communication purposes, especially when different groups are involved (so-called interfunctional communication). This introduces a set of challenges: how to attract the attention of users; how to provide orientation and a shared vision; how to present both an overview of the project as well as certain details; and how to initiate discussions and motivate individuals to participate in the project. Burkhard and Meier [★ reference] introduced the Tube Map Visualization for projects in an organization and evaluated its strengths and limitations in an comparative study [★ reference]. This section expands on work published previously by the authors in [★ reference].

Modifications to the system described in Section [★ cross-reference] are needed for the project planning application. In particular, we introduce a new metric to take the time axis into account and to change the method for initial positioning of nodes.

7.2.1 Timescale Criterion

With the introduction of the time axis, it becomes important that task nodes appear correctly in relation to each other. For example, a task that starts before another should appear to the left of the other task. For a task node n with x-coordinate xn and start time tn the timescale metric is found by calculating $(|tn - xn|)^2$. This has the effect of severely penalizing nodes that have strayed a long way from their start time while having little effect for nodes that are close to their start time. As with other metrics, the timescale metric is sufficiently weighted so as to be effective when used in combination with the other metrics.

A side effect of the timescale metric is that the edge length metric is no longer required, as otherwise the two metrics would conflict with each other.

7.2.2 Initial Embedding

When drawing metro maps, the starting position for the multicriteria optimization method was the geographic position of the stations. However this does not apply in the case of project plan data where the diagram is abstract. Initially, all the nodes are placed along a line such that they all have the same y-coordinate. The initial x-coordinate of the node is determined by the start time of the task.

7.2.3 Implementation

[★ Rewrite this section]

Our results look promising. Figure 14 [★ figure] shows an example of a project map that has been automatically laid out using the metro map metaphor. The project plan that the map represents is the one shown in Figure 13. Each line is drawn with a different color to allow them to be easily distinguished. Therefore, each member of the project can trace his or her route through the project, and in particular, which order their tasks take place and how they relate to other tasks. It is also clear as to which tasks are important for more than one member of the project (such as Task 7) where the lines from more than one member intersect. Project members working together are also clearly shown where two lines run in parallel (such as either side of Task 8 and between Task 17 and Task 9).

While the automatically generated project map has obvious advantages, it suffers from a number of disadvantages such as an apparently unnecessary edge crossing, edges drawn very close to nodes (as with the Start 1 node) and edges which are not four-gonal. These problems could obviously be solved with manual editing of the project map, but this is not really desirable as it is meant to automatically generate usable project maps.

Cosmetic problems are also evident. These include the way in which different lines swap places with each other as they pass through a node (e.g. the lines passing through the Task 8 node) and the way that many parallel lines (as between the Task 17 and Task 9 nodes) are wider than the nodes.

Figure 7.1: Example of a website that uses both breadcrumb trails (underneath the page heading) and a hierarchical list of contents (down the left-hand side). From jstott.me.uk.

7.3 Website Navigation Visualisation

Making websites easy to navigate for the user is a constant challenge facing website designers. Several different approaches are commonly used, including simple tabulated menus, breadcrumb trails (which show the trail of pages from the home page to the current page), as shown in Figure 7.1, and site maps (which show all the pages on the website, normally grouped into categories or listed alphabetically), as shown in Figure 7.2. All of these methods are created using simple text very few actually use a graphical visualization of the website structure. There could be several reasons for this, including a reluctance to use a navigation technique which might be unfamiliar to the user, lack of time to develop new techniques, or lack of expertise with creating graphical navigation aids.

We believe that the use of the metro map as a metaphor for navigating websites could prove to be a more effective solution than a simple site map of the website. Indeed, one might be of the opinion that the phrase site map would already imply some kind of graphical visualization of the site. However, nearly all site maps

Figure 7.2: Example of a website site map. From the University of Kent Computer Science website.

are simple lists of pages, perhaps organised into groups or a hierarchy. With a metro map of a website, each station (node) represents a single page (or possibly a group of pages) with logical connections between pages represented by edges between nodes. Concepts that connect several related pages can be shown by distinguishing that concept as a line in the map with a separate colour.

The metro map metaphor can be used when designing websites not just as a visualization tool but also to influence the fundamental design of the structure of the website. By making sure that the metaphor influences the design of the website, the strength of the metro map as a navigation visualization is greatly enhanced.

In this case study, we examine how a metro map can be used as the navigational basis for a website with the aim of providing an online tutoring website for a programming language. This application is particularly appropriate as the mappings between individual language concepts and stations on a metro map are likely to be fairly intuitive to the website user. Also, metro lines can be used to highlight particular key concepts that join together several smaller language components (such as graphical user interfaces or file input/output).

One of the key differences in the approach required to draw a metro map style diagram for this case is that none of the nodes in the graph have any association with a geographic location, be it an absolute location (such as a grid reference or latitude and longitude) or a relative location (such as one node must be north of another node). This has one clear benefit in that we no longer need be concerned with the relative positions of nodes, allowing for greater flexibility when drawing the graph. However, deciding on an initial embedding for the graph is problematic. This is due to our method being based on optimizing an existing layout: if the initial embedding is not adequate, then our method may struggle to produce an acceptable optimization.

Another difference is motivated by the way in which the user is likely to interact with the diagram. In this case, the user must be able to accept the visualization and be able to use it effectively in order to justify our claim that using a metro map metaphor enhances peoples ability to navigate a website. In order to achieve this, we need to include cues to the user for such things as where to start and where to go next. Such cues could be implicit (for example, the map is oriented with the start at the top with navigation flowing from top to bottom) or explicit (such as directed edges indicating direction of navigation or emphasis of the starting point in some way).

We have presented two different results, using a different initial embedding for each one.

The first of our website navigation results uses an initial embedding created using a spring embedder [★ reference]. The spring embedder operates on a randomly produced embedding with the intention of untangling the edges and separating the various lines. The repulsive force between unconnected nodes ensures that nodes that are not immediately adjacent are placed further apart allowing more space

Figure 7.3: A website navigation map using the metro map metaphor using a spring embedder to produce an initial embedding.

for labelling. Enough iterations of the spring embedder are executed such that as many edge crossings as possible are removed from the initial random embedding.

After the spring embedder has finished running, the nodes are moved to the nearest point where grid lines intersect and the metro map method is applied. Figure 7.3 shows the result of drawing the website navigation map using the metro map metaphor when using a spring embedder to produce the initial embedding.

The map clearly exhibits many of the characteristics of metro maps. As metro maps are a familiar visualization to many people, this immediately grabs the attention of the user and causes them to become interested in the map. The metro map metaphor gives the implicit impression that the map is designed for the purpose of navigation. Each line is emphasised in a different colour, allowing a clear distinction to be made between each of the other line. Nearly all of the nodes are unambiguously labelled (with exceptions on the line between the for/while and Collections nodes.

In this case, the labelling is quite good in that only one label ("Abstract Classes" at the top of the map) occludes an edge and no labels occlude any other labels or nodes. The occlusion with the "Abstract Classes" label occurs because the edge passes between each line of the label and doesnt intersect the bounding boxes for each line of the label. A number of examples of local minima in the labelling process are evident on the line between the "for/while" and "Collections" nodes and on the line between the "Graphics" and "Printing" nodes. In these cases, all the labels are in positions where the label position criterion has not been minimised because the label position consistency criterion has taken precedence. It could be possible to move the position of a group of labels like this if more than one label was being moved at the same time (similar to how node clustering works).

As it stands, one of the main problems with the map is that the starting point for navigation is not obvious. The intention is for the "Objects and Classes" node to be the point where navigation should start, but the lack of any visual cues to support this doesnt make the start point very clear. Also, the direction of navigation from one node to the next is not particularly intuitive. These drawbacks could be solved to come extent by the introduction of extra visual cues such as directed edges and by highlighting the start node by making the label larger or increasing the size of the node symbol. However, adding these visual cues would make the map look less like a metro map therefore negating some of the effect of the metaphor.

Our second result uses a hierarchical layout algorithm [★ reference] to find an initial embedding. Using a hierarchical layout algorithm makes sense in the case of website navigation as websites generally have some kind of hierarchy. The root of the hierarchy is selected as the start point in the navigation for a website, this would be the home page. In our example, the intention is to have the most fundamental concepts at the top of the map with a downwards movement implying more complex concepts. A modification to the hierarchical layout algorithm is therefore

Figure 7.4: A website navigation map using the metro map metaphor using a hierarchical layout algorithm to produe an initial embedding.

required such that this conceptual ordering is maintained. Figure 7.4 shows the result of drawing a website navigation map using the metro map metaphor when using a hierarchical layout algorithm to produce the initial embedding. To show an idea of how the visualization might be incorporated into a website, a mock-up is available online at http://www.jstott.me.uk/javatube/.

As with the first result, this result clearly exhibits many of the characteristics of metro maps. One of they key advantages of this result over the first one is that the initial hierarchical embedding forces the starting point for navigation to be at the top of the map with a top-to-bottom ordering of concepts. From there, each subsequent node is either below or at the same level as the previous node. The short branch from “extends” to “GUIs” is a result of our modification to the hierarchical layout algorithm to enforce the conceptual ordering. A top-to-bottom navigation like this makes it more intuitive for a user to find their way around the map. The direction of navigation could again be reinforced using directed edges as extra visual cues.

One of the features of the hierarchical layout of the second result is that there are a number of parallel lines, making it quite difficult to find enough space for labels. As such, this result contains two examples of labels that occlude edges the "Unchecked Exceptions" label and the "Method Overriding" label. There are also more examples of ambiguously positioned labels than in the first result. These are labels that are not clearly associated with any particular node because they are too close to another unrelated node. Examples of ambiguously positioned labels in this result include the "Object Locking" and "Synchronizing Threads" nodes.

7.4 Other Applications

Stuff.

Chapter 8

Conclusion

8.1 Summary of Contributions

Stuff.

8.2 Further Work

Stuff.

Bibliography

[AS01] M. Agrawala and C. Stolte. Rendering effective route maps: Improving usability through generalization. In *SIGGRAPH 2001*, pages 241–249, Los Angeles, USA, 2001.

[AV00] P. K. Agarwal and K. R. Varadarajan. Efficient Algorithms for Approximating Polygonal Chains. *Discrete and Computational Geometry*, 23(2):273–291, 2000.

[BBS97] J. Branke, F. Bucher, and H. Schmeck. Using Genetic Algorithms for Drawing Undirected Graphs. In *Third Nordic Workshop on Genetic Algorithms and their Applications*, pages 193–206, Helsinki, Finland, August 1997. Finnish Artificial Intelligence Society (Hrsg.).

[BCC+06] P. Bose, O. Cheong, S. Cabello, J. Gundmundsson, M. van Kreveld, and B. Speckmann. Area-Preserving Approximations of Polygonal Paths. *Journal of Discrete Algorithms*, 4(4):554–566, 2006.

[BCD+02] G. Barequet, D. Z. Chen, O. Daescu, M. T. Goodrich, and J. Snoeyink. Efficiently Approximating Polygonal Paths in Three and Higher Dimensions. *Algorithmica*, 33(2):150–167, 2002.

[BEKW02] U. Brandes, M. Eiglsperger, M. Kaufmann, and D. Wagner. Sketch Driven Orthogonal Graph Drawing. In *Proceedings of Graph Drawing 2002*, number 2528 in LNCS, pages 1–11. Springer, 2002.

[Ber99] F. Bertault. A Force-Firected Algorithm that Preserves Edge Crossing Properties. In *Proceedings of Graph Drawing 1999*, number 1731 in LNCS, pages 351–358. Springer, 1999.

[BGH02] Tove Bang, Kaj Grønbæk, and Per Steen Hansen. Using a metro map metaphor for organizing web-based learning resources. In *Proceedings of ED-MEDIA 2002 World Conference on Educational Multimedia, Hypermedia and Telecommunications*, pages 647–652. Association for the Advancement of Computing in Education (AACE), 2002.

[BLR00] Thomas Barkowsky, Longin Jan Latecki, and Kai-Florian Richter. Schematizing maps: Simplification of geographic shapes by discrete curve elimination. In Christian Freksa, Wilfried Brauer, Christopher Habel, and Karl Friedrich Wender, editors, *Spatial Cognition II, Integrating Abstract Theories, Empirical Studies, Formal Methods and Practical Applications*, volume 1849 of *LNCS 1849*, pages 41–53. Springer, 2000.

[BNUW07] Marc Benkert, Martin Nöllenburg, Takeaki Uno, and Alexander Wolff. Minimizing Intra-Edge Crossings in Wiring Diagrams and Public Transportation Maps. In M. Kaufmann and D. Wagner, editors, *Proceedings of the 14th International Symposium on Graph Drawing (GD2006)*, volume 4372 of *Lecture Notes in Computer Science (LNCS)*, pages 270–281. Springer-Verlag, 2007.

[BVG07] Berlin Verkehrsbetriebe (BVG) website. http://www.bvg.de/index.php/en/Bvg/Start, 28 July 2007.

[CBKF01] H. Casakin, T. Barkowsky, A. Klippel, and C. Freska. Schematic Maps as Wayfinding Aids. In *Spatial Cognition II*, number 1849 in

LNCS, pages 54–71. Springer, 2001.

[CC96] W. S. Chan and F. Chin. Approximation of Polygonal Curves with Minimum Number of Line Segments or Minimum Error. *International Journal of Computational Geometry and Applications*, 6:59–77, 1996.

[CD03] D. Z. Chen and O. Daescu. Space-Efficient Algorithms for Approximating Polygonal Curves in Two-Dimensional Space. *International Journal of Computational Geometry and Applications*, 13:95–111, 2003.

[CDH+05] D. Z. Chen, O. Daescu, J. Hershberger, P. M. Kogge, N. Mi, and J. Snoeyink. Polygonal Path Simplification with Angle Constraints. *Computational Geometry: Theory and Applications*, 32(3):173–187, 2005.

[CMS95] J Christensen, J. Marks, and S. Shieber. An empirical study of algorithms for point-feature label placement. *ACM Transactions on Graphics*, 14(3):203–232, 1995.

[dBETT94] M. di Battista, P. Eades, R Tamassia, and I. G. Tollis. Algorithms for Drawing Graphs: an Annotated Bibliography. Available from ftp://wilma.cs.brown.edu/pub/papers/compgeo/gdbiblio.ps.Z, June 1994.

[dBETT99] M. di Battista, P. Eades, R Tamassia, and I. G. Tollis. *Graph Drawing: Algorithms for the Visualisation of Graphs*. Prentice Hall, 1999.

[DH96] R. Davidson and D. Harel. Drawing Graphs Nicely Using Simulated Annealing. *ACM Transactions on Graphics*, 15(4):301–331, 1996.

[DM90] C. Ding and P. Mateti. A Framework for the Automated Drawing of Data Structure Diagrams. *ACM Transactions on Software Engineering*, 16(5):543–557, May 1990.

[DP73] D. Douglas and T. Peucker. Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or its Caricature. *The Canadian Cartographer*, 10(2):112–122, 1973.

[Ead84] Peter Eades. A heuristic for graph drawing. *Congressus Numeratum*, 42:149–160, 1984.

[ET94] D. Eu and G. T. Toussaint. On Approximating Polygonal Curves in Two and Three Dimensions. *CVGIP: Graphical Model and Image Processing*, 56(3):231–246, 1994.

[FK95] U. Fößmeier and M. Kaufmann. Drawing High Degree Graphs with Low Bend Numbers. In *Proceedings of the Symposium on Graph Drawing*, pages 254–266, 1995.

[FLM95] A. Frick, A. Ludwig, and H. Mehldau. A Fast Adaptive Layout Algorithm for Undirected Graphs. In *Proceedings of Graph Drawing 1994.*, number 894 in LNCS, pages 388–403. Springer, 1995.

[Flo62] R. W. Floyd. Algorithm 97: Shortest Path. *Communications of the ACM*, 5:354, 1962.

[FR94] T. M. J. Fruchterman and E. M. Reingold. Graph Drawing by Force-Directed Placement. *Software - Practice and Experience*, 21(11):1129–1164, November 1994.

[Gar94] K. Garland. *Mr. Beck's Underground Map*. Capital Transport Publishing, 1994.

[GHMS93] L. J. Guibas, J. Hershberger, J. S. B. Mitchell, and J. Snoeyink. Approximating Polygons and Subdivisions with Minimum Link Paths. *International Journal of Computational Geometry and Applications*, 3(4):383–415, 1993.

[GJ83] M. R. Garey and D. S. Johnson. Crossing Number is NP-Complete. *SIAM Journal of Algebraic and Discrete Methods*, 4(3):312–316, 1983.

[GNS07] J. Gudmundsson, G. Narasimhan, and M. H. M. Smid. Distance-Preserving Approximations of Polygonal Paths. *Computational Geometry Theory and Applications*, 36(3):183–196, April 2007.

[Goo95] M. T. Goodrich. Efficient Piecewise-Linear Function Approximation Using the Uniform Metric. *Discrete and Computational Geometry*, 14:445–462, 1995.

[Hen29] G. Hennet. A Map of the County Palatine of Lancaster. http://www.lancashire.gov.uk/environment/oldmap/hennet/index.htm, 1829.

[HMdN04] Seok-Hee Hong, Damian Merrick, and Hugo A. D. do Noscimento. The metro map layout problem. In N. Churcher and C. Churcher, editors, *Information Visualisation 2004*, volume 35 of *Conferences in Research and Practice in Information Technology*, pages 91–100. ACS, 2004.

[HMdN06] Seok-Hee Hong, Damian Merrick, and Hugo A. D. do Noscimento. Automatic visualisation of metro maps. *Journal of Visual Languages and Computing*, 17:203–224, 2006.

[Hoc07] Hochbahn (Hamburg metro) website. http://www.hochbahn.com/, 28 July 2007.

[HS98] J. Hershberger and J. Snoeyink. Cartographic Line Simplification and Polygon CSG Formulae in O(n log* n) Time. *Computational Geometry Theory and Applications*, 11(3-4):178–185, 1998.

[HW02] William C. Hahn and Robert A Weinberg. A subway map of cancer pathways. *Nature Reviews Cancer*, 2(5), May 2002.

[II86a] H. Imai and M. Iri. Computational-Geometric Methods for Polygonal Approximations of a Curve. *Computational Vision, Graphic and Image Processing*, 36:31–41, 1986.

[II86b] H. Imai and M. Iri. An Optimal Algorithm for Approximating a Piecewise Linear Function. *Journal of Information Processing*, 9(3):159–162, 1986.

[II88] H. Imai and M. Iri. Polygonal Approximations of a Curve-Formulation and Algorithms. In G. T. Toussaint, editor, *Computational Morphology*, Amsterdam, 1988.

[KK89] T. Kamada and S. Kawai. An Algorithm for Drawing General Undirected Graphs. *Information Processing Letters*, 31:7–15, 1989.

[KT98] K. G. Kakoulis and I. G. Tollis. A Unified Approach to Labelling Graphical Features. In *14th Annual ACM Symposium in Computational Geometry*. ACM, June 1998.

[LS01] U. Lauther and A. Stübinger. Generating Schematic Cable Plans Using Spring Embedder Methods. In *Proceedings of Graph Drawing 2001*, number 2265 in LNCS, pages 465–466. Springer, 2001.

[MAR06] Atlanta Metropolitan Area Rapid Transit Authority (MARTA) website. http://www.itsmarta.com/, 2006.

[MBT07] Massachusetts Bay Transportation Authority (MBTA) website. http://www.mbta.com/, 28 July 2007.

[MdM07] Metro de Madrid website. http://www.metromadrid.es/, 28 July 2007.

[Met06] Metrorex (Bucharest Metro) website. http://www.metrorex.ro/, 2006.

[Mex06] Mexico City 'Sistema de Transporte Colectivo' website. http://www.metro.df.gob.mx/, 2006.

[MG06] Damian Merrick and Joachim Gudmundsson. Increasing the readability of graph drawings with centrality-based scaling. In K Misue, K Sugiyama, and J Tanaka, editors, *Proceedings of the Asia Pacific Symposium on Information Visualization 2006 (APVIS 2006)*, volume 60 of *Conferences in Research and Practice in Information Technology*, pages 67–76. ACS, 2006.

[MG07] Damian Merrick and Joachim Gudmundsson. Path simplification for metro map layout. In M Kaufmann and D Wagner, editors, *Proceedings of the 14th International Symposium on Graph Drawing (GD2006)*, volume 4372 of *Lecture Notes in Computer Science*, pages 258–269. Springer-Verlag, 2007.

[ML007] Metropolitano de Lisboa website. http://www.metrolisboa.pt/, 28 July 2007.

[MO85] Z. Miller and J. B. Orlin. NP-Completeness for Minimizing Maximum Edge Length in Grid Embeddings. *Journal of Algorithms*, 6:10–16, 1985.

[MO88] A. Melkman and J. O'Rourke. On Polygonal Chain Approximation. In G. T. Toussaint, editor, *Computational Morphology*, Amsterdam, 1988.

[MTA07] MTA New York City Transit website. http://www.mta.info/nyct/maps/submap.htm, 28 July 2007.

[MVG07] Münchner Verkehrsgesellschaft (MVG) website. http://www.mvg-mobil.de/, 28 July 2007.

[N05̈] Martin Nöllenburg. *Automated Drawing of Metro Maps*. PhD thesis, Institut fr Theoretische Informatik Universitt Karlsruhe, 20 August 2005.

[Nes04] K. V. Nesbitt. Getting to more abstract places with the metro map metaphor. In Ebad Banissia, Katy Brner, Chaomei Chen, Muhammad Dastbaz, Gordon Clapworthy, Anthony Faiola, Ebroul Izquierdo, Carsten Maple, Jonathan Roberts, Chris Moore, Anna Ursyn, and Jian J Zhang, editors, *Information Visualisation '04*, pages 488–493, London, England, 2004. IEEE Computer Society.

[Ney99] G. Neyer. Line Simplification with Restricted Orientations. In *Proceedings of the 6th International Workshop on Algorithms and Data Structures*, pages 13–24, 1999.

[ORe03] O'Reilly 2003 Open Source Route Map. http://www.oreilly.de/artikel/routemap.html, 2003.

[OTB07] Oslo T-Bane website. http://www.tbane.no/, 28 July 2007.

[Ove03] Mark Ovenden. *Metro Maps of the World*. Capital Transport Publishing, 2003.

[PCJ95] Helen C. Purchase, R. F. Cohen, and M. Jones. Validating graph drawing aesthetics. In Franz Brandenburg, editor, *Symposium on Graph Drawing*, volume 1027 of *LNCS 1027*, pages 435–446, Passau, Germany, 1995. Springer.

[PT97] A. Papakostas and I. G. Tollis. Orthogonal Drawing of High Degree Graphs with Small Area and Few Bends. In *WADS 1997*, pages 354–367, 1997.

[RAT07] Régie Autonome des Transports Parisiens (RATP) website. http://www.ratp.fr/, 28 July 2007.

[Rob05] Maxwell J Roberts. *Underground Maps After Beck*. Capital Transport Publishing, 2005.

[Ros04] Doug Rose. Personal communication, 2004.

[SGSK01] E. S. Sandvad, K. Grønbæk, L. Sloth, and J. L. Knudsen. A Metro Map Metaphor for Guided Tours on the Web: the Webvise Guided Tour System. In *Proceedings of the 10th International World Wide Web Conference*, pages 326–333, New York, May 2001. ACM.

[SKT00] J. M. Six, K. G. Kakoulis, and I. G. Tollis. Techniques for the Refinement of Orthogonal Graph Drawings. *Journal of Graph Algorithms and Applications*, 4(3):75–103, 2000.

[SL006] Storstockholms Lokaltrafik (SL) website. http://www.sl.se/, 2006.

[SM94] K. Sugiyama and K. Misue. A Simple and Unified Method for Drawing Graphs: Megnetic-Spring Algorithm. In *DIMACS International Workshop, Proceedings of Graph Drawing 1994*, number 894 in LNCS, pages 364–375. Springer, 1994.

[SM95] K. Sugiyama and K. Misue. Graph Drawing by Magnetic Spring Model. *Journal of Visual Languages and Computing*, 6(3):217–231, 1995.

[Tam87] R. Tamassia. On Embedding a Graph in the Grid with the Minimum Number of Bends. *SIAM Journal of Computing*, 16(3):421–444, June 1987.

[TfL05] Transport for London website. http://www.tfl.gov.uk/tube/maps/, 22 April 2005.

[TM007] Tokyo Metro website. http://www.tokyometro.jp/e/index.html, 28 July 2007.

[TMB07] Transports Metropolitans de Barcelona (TMB) website. http://www.tmb.net/, 28 July 2007.

[Tou85] G. T. Toussaint. On the Complexity of Approximating Polygonal Curves in the Plane. In *Proceedings of the International Symposium on Robotics and Automation (IASTED)*, pages 311–318, 1985.

[TTC06] Toronto Transit Commission (TTC) website. http://www.toronto.ca/ttc/index.htm, 2006.

[Tun94] D. Tunkelang. A Practical Approach to Drawing Undirected Graphs. Technical Report CMU-CS-94-161, Carnegie Mellon University, School of Computer Science, 1994.

[vKSW98] M. van Kreveld, T. Stijk, and A. Wolff. Point Labelling with Sliding Labels. In *Proceedings of the 14th Annual ACM Symposium of Computational Geometry*, pages 337–346, 1998.

[VVS07] Verkehrs- und Tarifverbund Stuttgart website. http://www.vvs.de/, 28 July 2007.

[Wil47] Charles Wilson. *Harwich and the Continent*. LNER, 1947.

[WMA06] Washington Metropolitan Area Transit Authority (WMATA) website. http://www.wmata.com/metrorail/systemmap.cfm, 2006.

[WPCM02] C. Ware, H. Purchase, L. Colpoys, and M. McGill. Cognitive Measurements of Graph Aesthetics. *Information Visualization*, 1(2):103–110, 2002.

[YH98] P. Yates and M. Humphries. The Generation of Schematic Diagrams from Geographic Representations of Networks. In *Proceedings of the Spatial Information Research Centre's 10th Colloquium*, 1998.

Appendix A

Examples of Method Output

This appendix contains full-page versions of automatically-drawn maps reproduced at a smaller scale elsewhere in this thesis:

- Figure A.1: Atlanta MARTA.
- Figure A.2: Bucharest.
- Figure A.3: Mexico City.
- Figure A.4: Stockholm.
- Figure A.5: Toronto.
- Figure A.6: Washington.

Figure A.1: Atlanta MARTA map.

Figure A.2: Bucharest map.

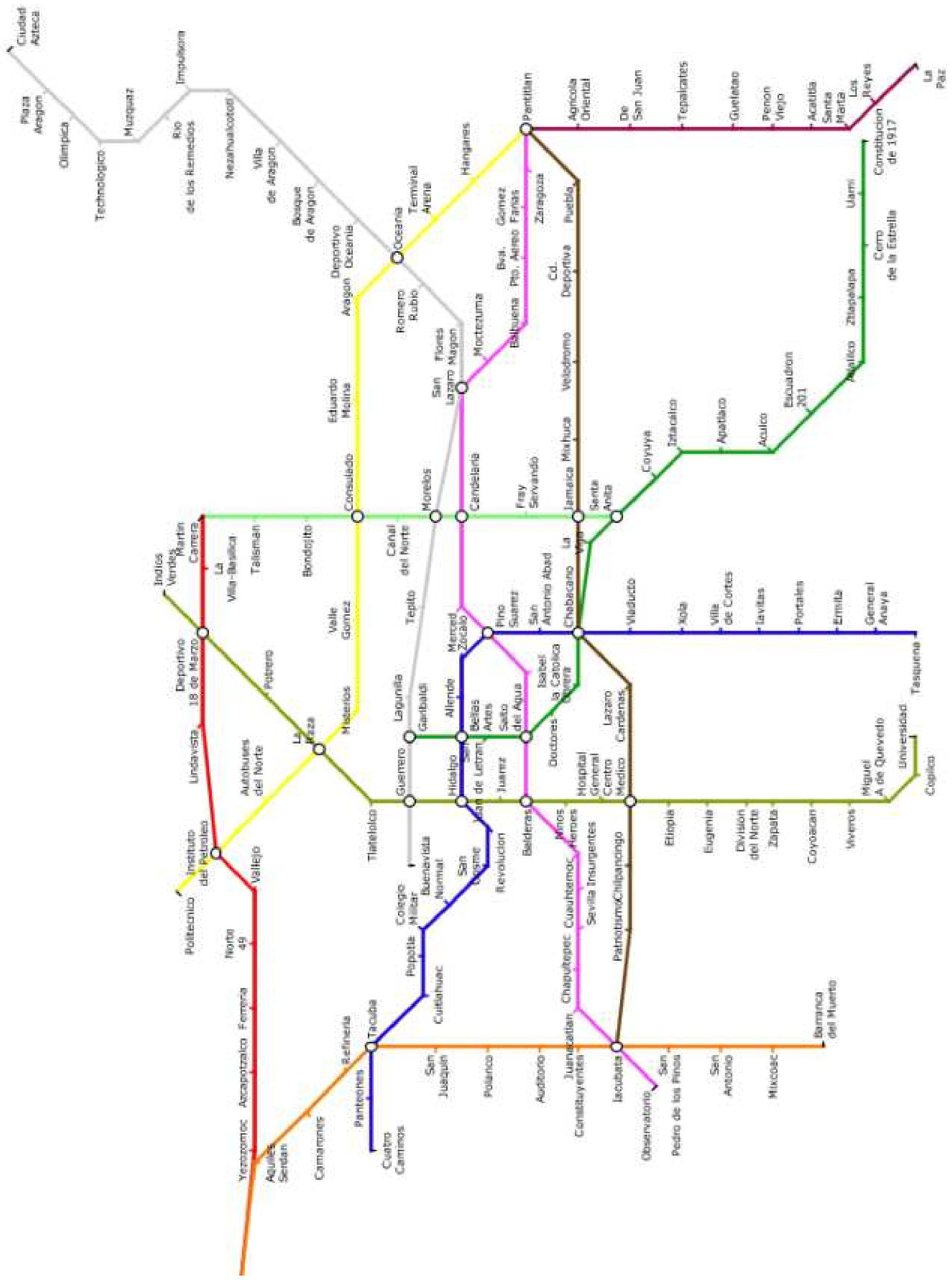


Figure A.3: Mexico City map.

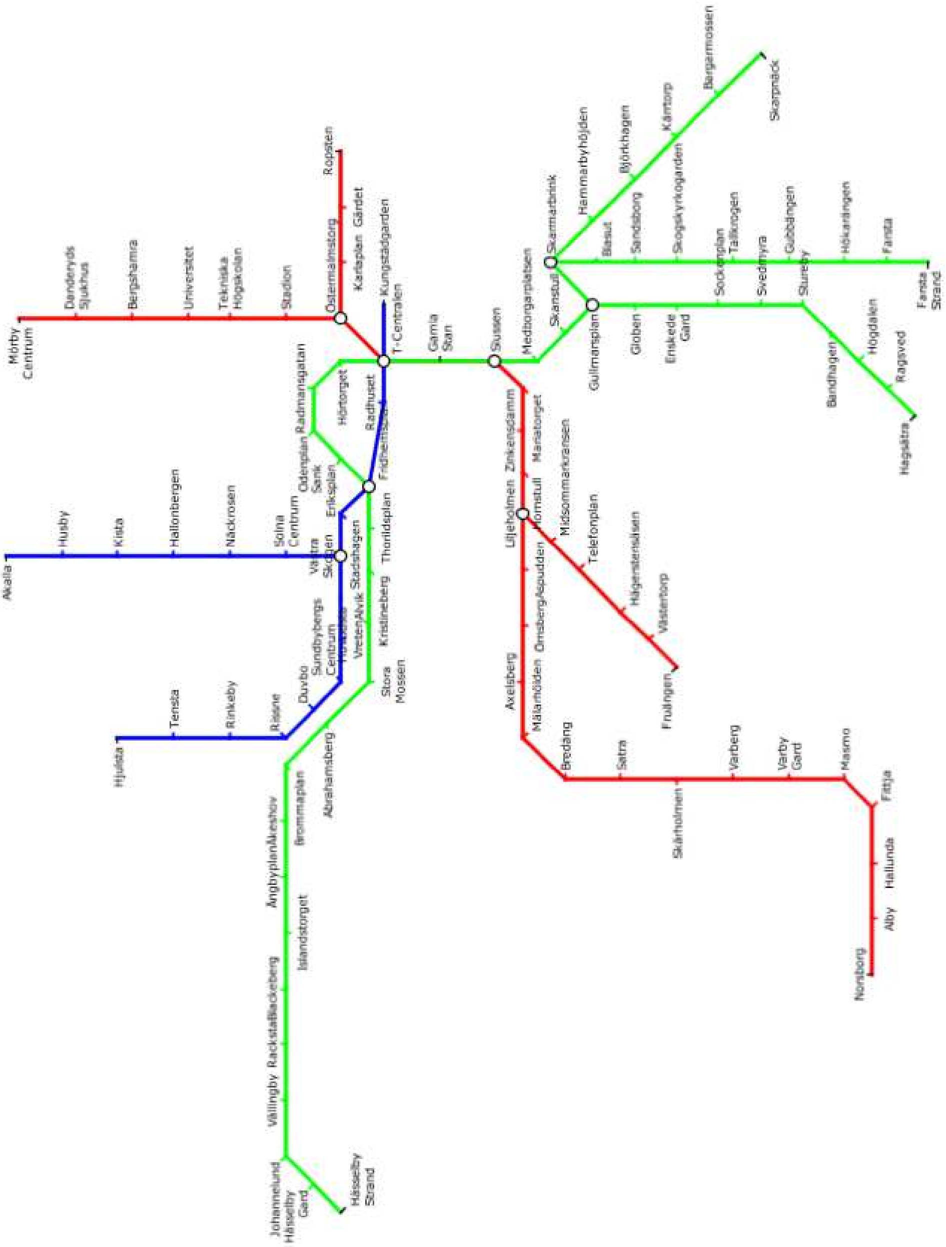


Figure A.4: Stockholm map.

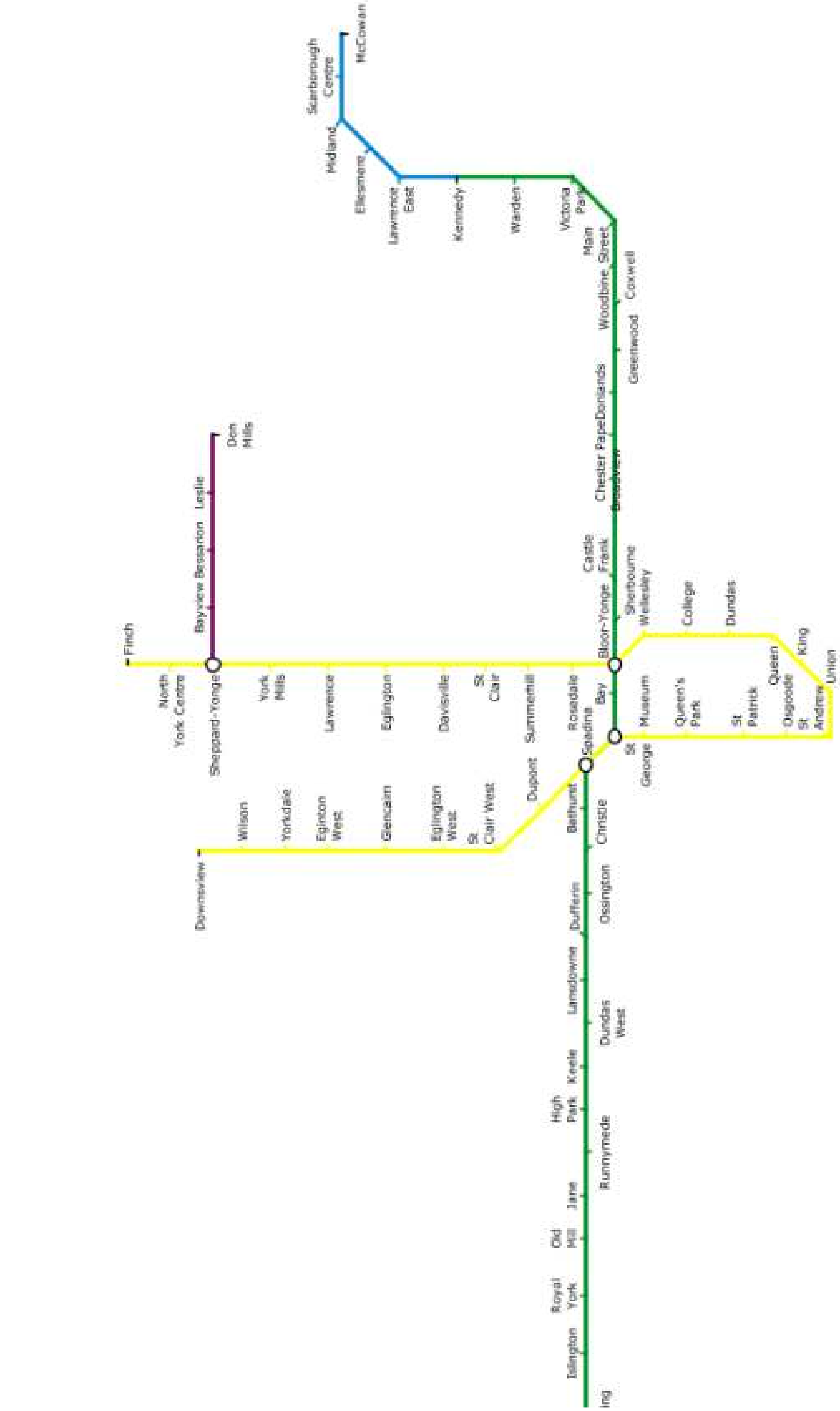


Figure A.5: Toronto map.

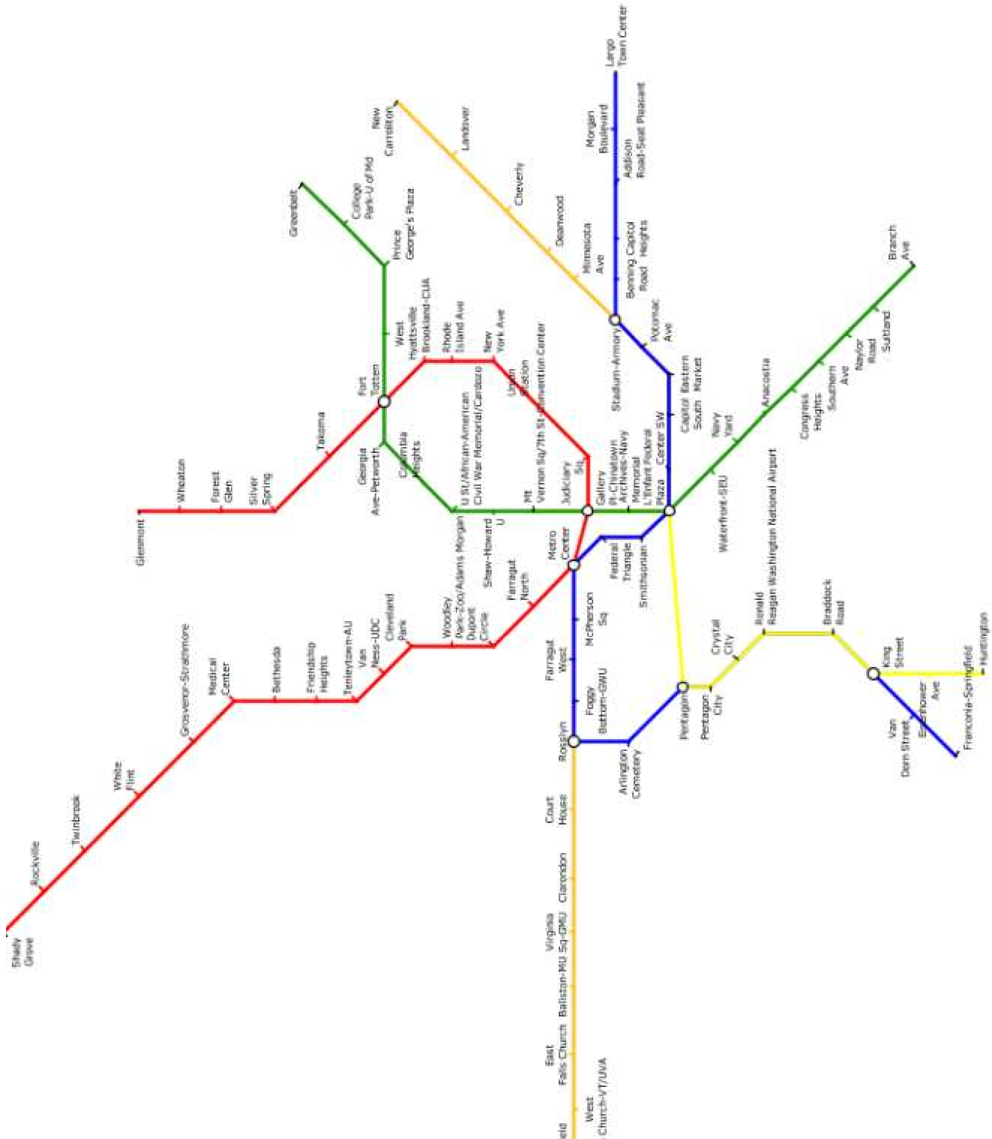


Figure A.6: Washington D.C. map.

Appendix B

Empirical Evaluation

B.1 Questions and Maps Used

B.1.1 Atlanta

- Geographic map (Figure B.1).
- Normalised published map (Figure B.2).
- Automatically-drawn map (Figure B.3).

Table B.1: Atlanta questions.

Number	Question	Answer Options	Answer
1.a	How many stations do you pass through to get from 'Bankhead' to 'Lenox'	$\{8, 9, 10, 11, 12\}$	10
1.b	How many stations do you pass through to get from 'Vine City' to 'Midtown'	$\{4, 5, 6, 7, 8\}$	5
1.c	What is the minimum number of changes to get from 'Indian Creek' to 'West Lake'	$\{0, 1, 2, 3, 4\}$	0
1.d	What is the minimum number of changes to get from 'College Park' to 'Five Points'	$\{0, 1, 2, 3, 4\}$	0

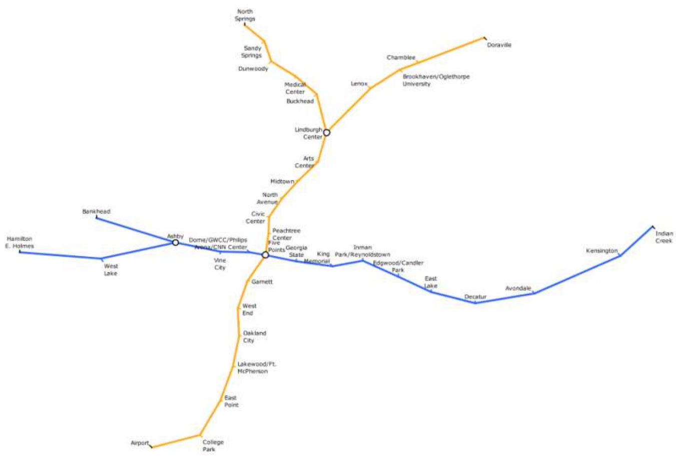


Figure B.1: Atlanta MARTA geographic map.

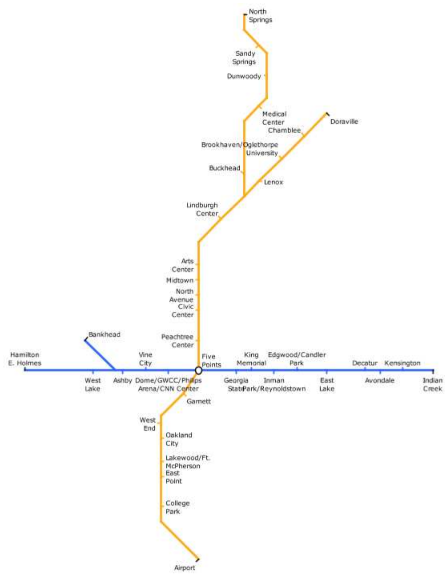


Figure B.2: Atlanta MARTA normalised published map.

Figure B.3: Atlanta MARTA automatically-drawn map.

B.1.2 Bucharest

- Geographic map (Figure B.4).
- Normalised published map (Figure B.5).
- Automatically-drawn map (Figure B.6).

Table B.2: Bucharest questions.

Number	Question	Answer Options	Answer
2.a	How many stations do you pass through to get from 'Pacii' to 'Titan'	$\{9, 10, 11, 12, 13\}$	10
2.b	How many stations do you pass through to get from 'Romana' to 'Piata Sudului'	$\{1, 2, 3, 4, 5\}$	5
2.c	What is the minimum number of changes to get from 'Eroilor' to 'Iancului'	$\{0, 1, 2, 3, 4\}$	0
2.d	What is the minimum number of changes to get from 'Eroii Revolutiei' to '1 Mai'	$\{0, 1, 2, 3, 4\}$	2

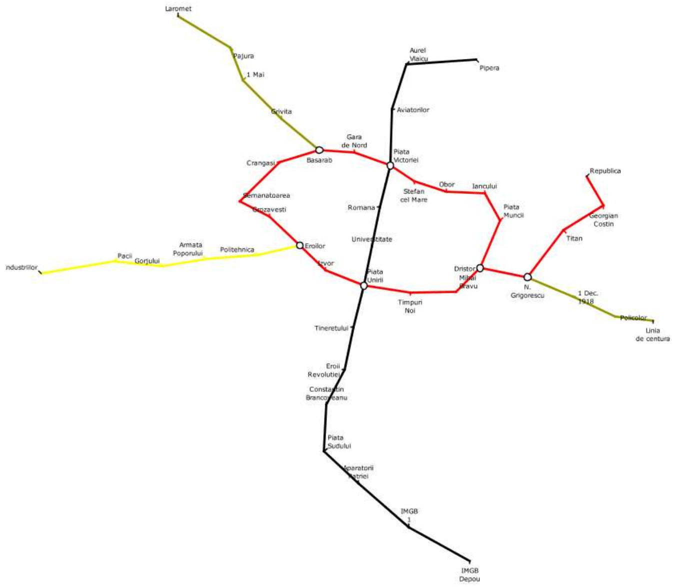


Figure B.4: Bucharest geographic map.

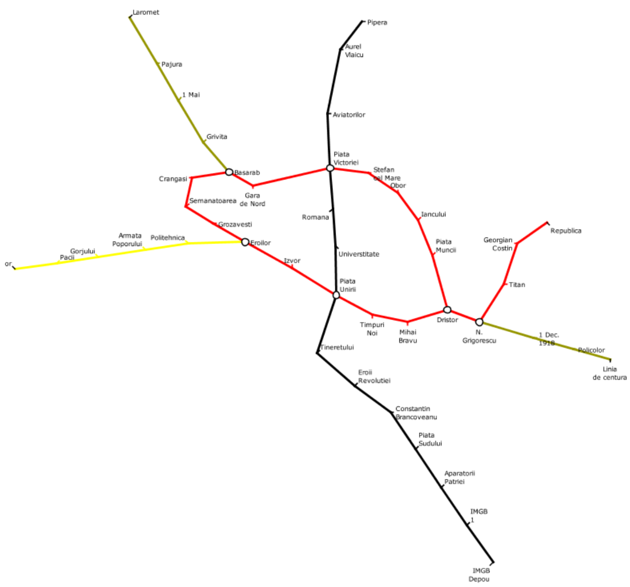


Figure B.5: Bucharest normalised published map.

Figure B.6: Bucharest automatically-drawn map.

B.1.3 Mexico City

- Geographic map (Figure B.7).
- Normalised published map (Figure B.8).
- Automatically-drawn map (Figure B.9).

Table B.3: Mexico City questions.

Number	Question	Answer Options	Answer
3.a	How many stations do you pass through to get from ‘Balderas’ to ‘Consulado’	$\{4,5,6,7,8\}$	7
3.b	How many stations do you pass through to get from ‘Refineria’ to ‘Patriotismo’	$\{4,5,6,7,8\}$	6
3.c	What is the minimum number of changes to get from ‘Sevilla’ to ‘Aragon’	$\{0,1,2,3,4\}$	2
3.d	What is the minimum number of changes to get from ‘Martin Carrera’ to ‘’	$\{0,1,2,3,4\}$	2

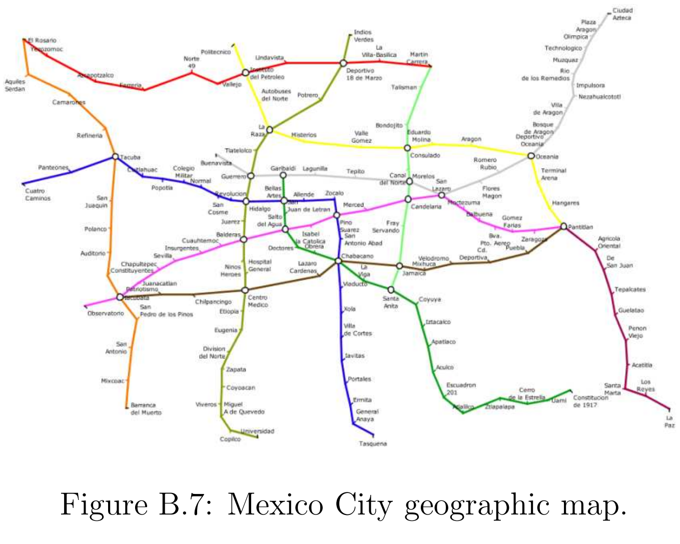


Figure B.7: Mexico City geographic map.

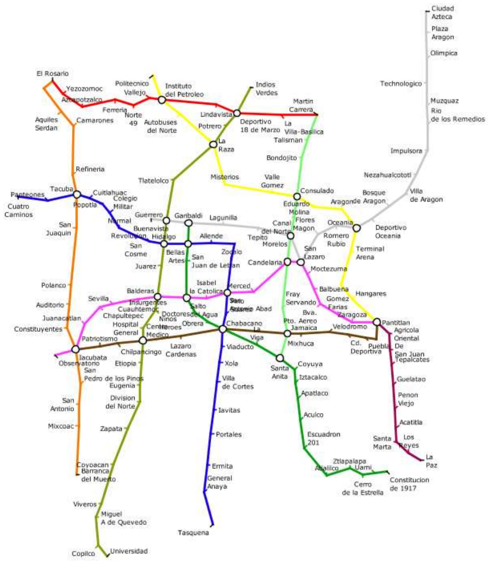


Figure B.8: Mexico City normalised published map.

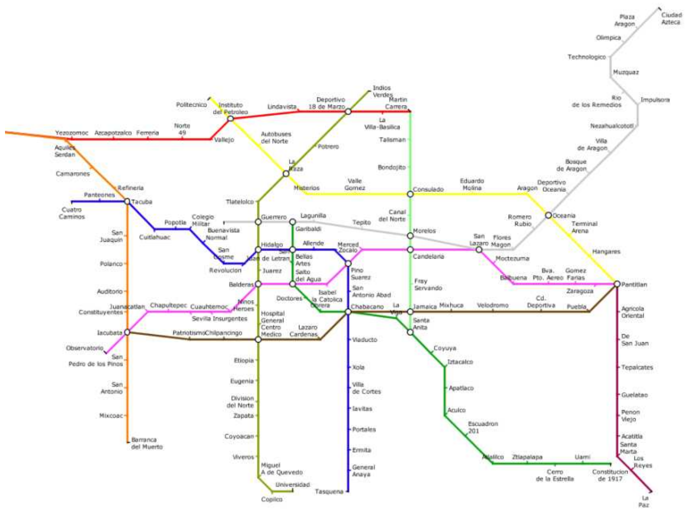


Figure B.9: Mexico City automatically-drawn map.

B.1.4 Stockholm

- Geographic map (Figure B.10).
- Normalised published map (Figure B.11).
- Automatically-drawn map (Figure B.12).

Table B.4: Stockholm questions.

Number	Question	Answer Options	Answer
4.a	How many stations do you pass through to get from ‘Stora Mossen’ to ‘Karlaplan’	$\{6, 7, 8, 9, 10\}$	10
4.b	How many stations do you pass through to get from ‘Liljeholmen’ to ‘Kungsträdgarden’	$\{2, 3, 4, 5, 6\}$	6
4.c	What is the minimum number of changes to get from ‘Kista’ to ‘T-Centralen’	$\{0, 1, 2, 3, 4\}$	0
4.d	What is the minimum number of changes to get from ‘Bergamossen’ to ‘Axelsburg’	$\{0, 1, 2, 3, 4\}$	1

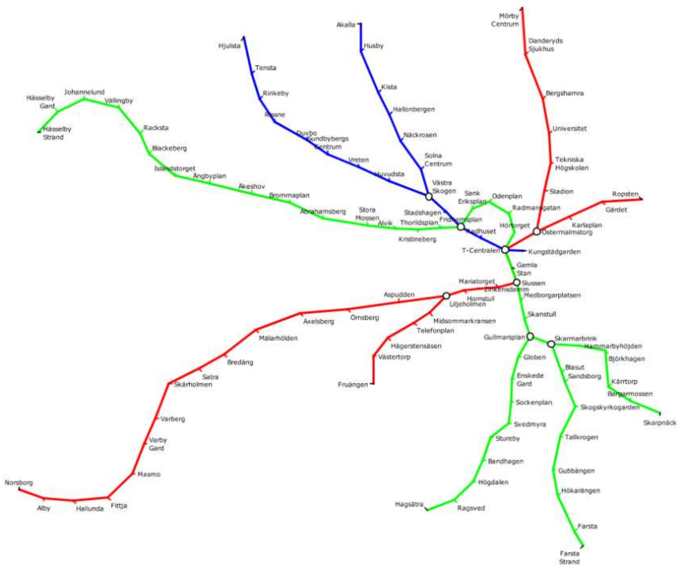


Figure B.10: Stockholm geographic map.

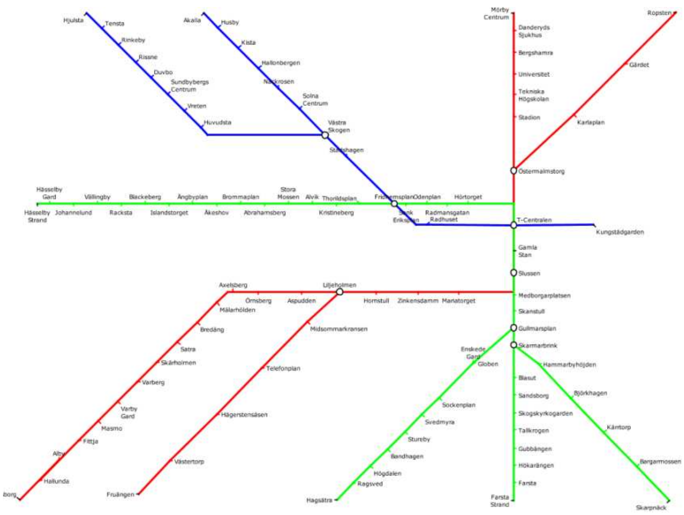


Figure B.11: Stockholm normalised published map.

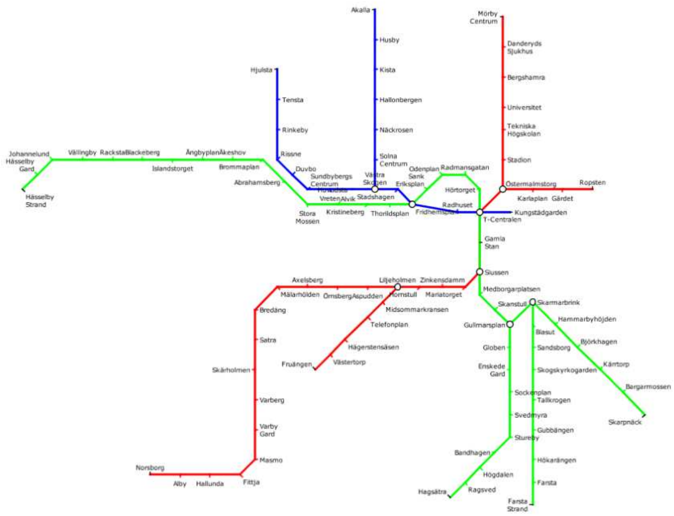


Figure B.12: Stockholm automatically-drawn map.

B.1.5 Toronto

- Geographic map (Figure B.13).
- Normalised published map (Figure B.14).
- Automatically-drawn map (Figure B.15).

Table B.5: Toronto questions.

Number	Question	Answer Options	Answer
5.a	How many stations do you pass through to get from 'Dupont' to 'Chester'	$\{6, 7, 8, 9, 10\}$	7
5.b	How many stations do you pass through to get from 'Lansdowne' to 'York Mills'	$\{10, 11, 12, 13, 14\}$	14
5.c	What is the minimum number of changes to get from 'Bayview' to 'Union'	$\{0, 1, 2, 3, 4\}$	1
5.d	What is the minimum number of changes to get from 'Midland' to 'Dundas'	$\{0, 1, 2, 3, 4\}$	2

Figure B.13: Toronto geographic map.

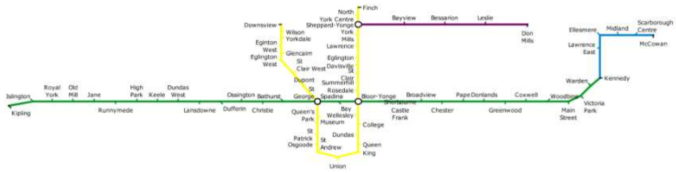


Figure B.14: Toronto normalised published map.

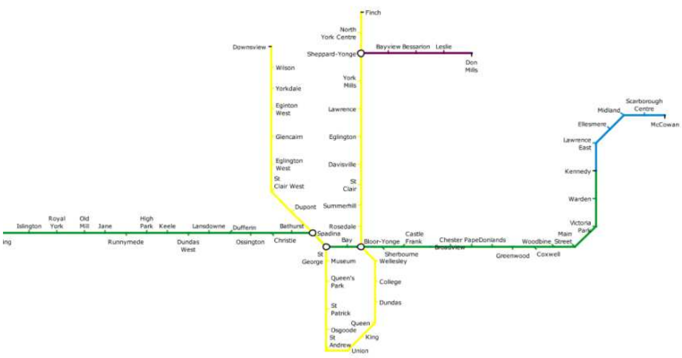


Figure B.15: Toronto automatically-drawn map.

B.1.6 Washington D.C.

- Geographic map (Figure B.16).
- Normalised published map (Figure B.17).
- Automatically-drawn map (Figure B.18).

Table B.6: Washington questions.

Number	Question	Answer Options	Answer
6.a	How many stations do you pass through to get from 'Pentagon' to 'Court House'	$\{1, 2, 3, 4, 5\}$	2
6.b	How many stations do you pass through to get from 'Cleveland Park' to 'Federal Triangle'	$\{4, 5, 6, 7, 8\}$	4
6.c	What is the minimum number of changes to get from 'Metro Center' to 'Takoma'	$\{0, 1, 2, 3, 4\}$	0
6.d	What is the minimum number of changes to get from 'Largo Town Center' to 'Eisenhower Ave.'	$\{0, 1, 2, 3, 4\}$	2

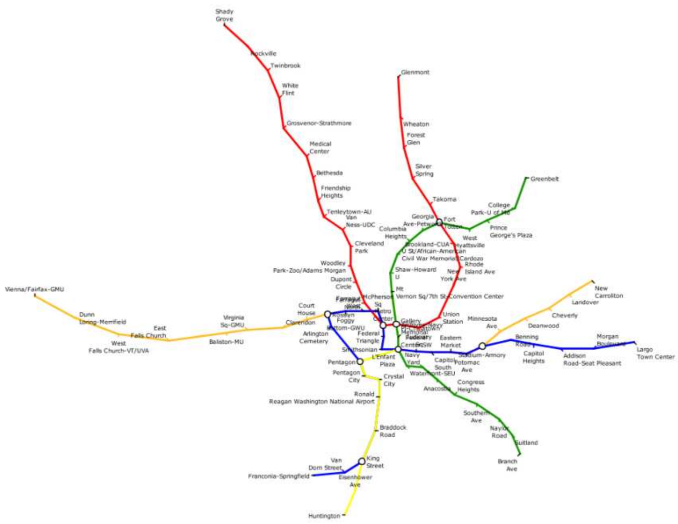


Figure B.16: Washington D.C. geographic map.

Figure B.17: Washington D.C. normalised published map.

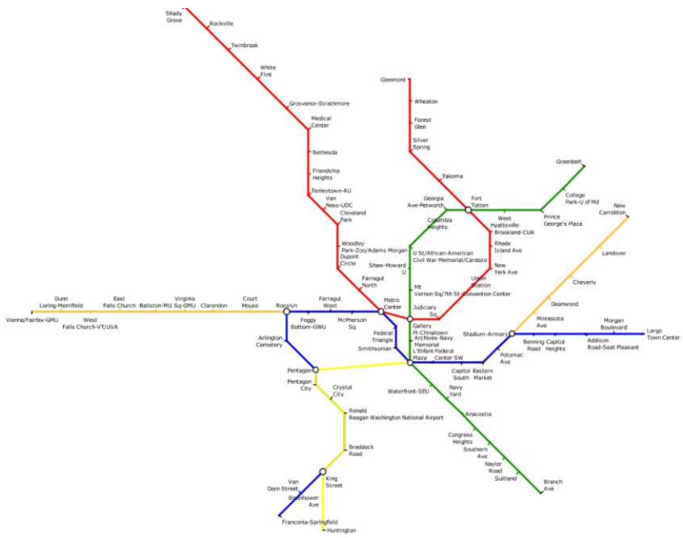


Figure B.18: Washington D.C. automatically-drawn map.

B.2 Associated Material

As a number of sessions of the empirical experiment were run at different times, it was essential that each session was as similar as possible. The following scripts and handouts were standard across all sessions.

B.2.1 Preliminary Script

This script was read aloud to candidates before they started the experiment.

> [The test software should be shown on the projector]
>
> Please do not start using a computer until told to do so. During this test, do not talk, or attempt to see what other participants are answering. If you have a query, please raise your hand.
>
> Although we ask for your login so that we can collate the data, the results of this test and questionnaire will be anonymized.
>
> You will be presented with a sequence of metro map diagrams. The test will pose a question for each map which requires an answer to be selected. The question will require you to plan a route between two stations on the map.
>
> You will first need to enter your login, level of study, year of study, age and gender and click OK. Do not do this yet, you will be told when to start the test.
>
> [Enter login test level of study Undergraduate, year of study 2, Age 25, Gender Male, then press OK. Press Start]
>
> A metro map is used to depict the interconnections on a public transport system so that the user is able to plan and undertake a specific journey. Stations are represented by circles which are labelled with the name of the station. A line in a single colour indicates which stations are connected by direct services. Where two or more lines pass through a single station you are able to change from one line to the other. See an example of a metro map on the projector.
>
> When you start the test, you will be shown a metro map in the main part of the screen [Point to map]. The question will be shown in the top-right corner [Point to question] with a selection of answers below [Point to answers].
>
> The questions that will be asked will involve planning a journey between two stations. The stations will be highlighted on the map in order that you can identify them more easily [Point to highlighted stations].
>
> Once you have worked out the route for the question, click the button next to the answer in the list shown before clicking the Go button.

In this case the question is How many changes are required to get from 'Shopping' to 'Barro'. I can see that you need to change once from one coloured line to another [point on screen]. So I click 1 [Click option 1]. Then I click GO [Click GO].

After each question you need to give an indication of the difficulty of the question. To do this, select the appropriate option from the list from very easy to very hard. Then click the OK button.

The period between clicking Go and answering the difficulty question are an opportunity to rest, if you need to do so, as timing does not start again until the OK button is clicked.

[Click Average then OK]

This next question asks me How many stations do I go through to get from Areoporto to Santa Luzia. With this sort of question, you do not count the end stations, only the stations in-between. Counting the stations, including the station that requires me to change I get an answer of 9[Point at each intermediate station, counting]. [Click option 9, then Go]

Please do not rush the questions, and take some effort to get the questions correct. Whilst we are measuring the time it takes to complete each answer, we do not mind if you dont complete all the questions.

You will be presented with questions for 20 minutes. After this time is up you will be shown how many questions you got correct as well the answers to any questions that you got incorrect.

At the end of the test, do not log off.

After the test, you need to complete a short questionnaire

Enter your details and press OK then the start button to begin the test now.

B.2.2 Postliminary Script

This script was read out after the interactive part of the experiment had concluded.

[The first slide should be showing on the projector]

[Hand out 1 questionnaire and 1 pen to each participant]

Please first fill in your login on the sheet in front of you, and then look up at the projector screen. You will be shown three metro maps at a time. Please decide which of these maps would be best for navigating a metro map system. As each slide is shown, write down 1, 2 or 3 in the spaces below, where 1 is the most preferable map and 3 is the least preferable map

I will count down from 5 before showing the next set of metro maps

> [Count down 5,4,3,2,1 then show the next slide]
>
> [Wait for a minute and then count down 5,4,3,2,1 for each slide, until the end of the presentation appears]
>
> Please now take 5 minutes to fill in questions 1,2,3 and 4 by hand.
>
> [Wait for 5 minutes and then start handing out five pound notes, getting signatures and handing out debriefing scripts. The experiment is now over, and you can answer questions about the tasks].

B.2.3 Concluding Handout

This text was given to candidates after they had completed the questionnaire and before they left the experiment.

> Thank you for participating in this research
>
> You were presented with a number of maps which were drawn using three different techniques. One version was drawn using the geographic layout of stations; the second version was drawn from a published map of the network; the final version was drawn using an automatic method which balances aesthetic criteria to try and find an optimal layout.
>
> The purpose of this research is to qualify some design aesthetics for automatically laying out metro maps and to see if our automated method is good at producing comprehensible diagrams.
>
> The idea is that being able to automatically produce metro maps might improve their use for navigating metro networks. In addition, being able to automatically generate the such maps could lead to them being more widely used for many other application areas.
>
> We would appreciate it if you did not discuss this experiment with other students in the university. These experiments will be continuing through the last two weeks of term, and having subjects who have prior knowledge of what the tests are about makes the data less useful.
>
> Thank you again for your contribution.

B.3 Results

B.3.1 Route Planning Tasks

Table B.7:

Candidate	Group	Map[a]	Question	Map Type[b]	Answer[c]	Given Answer	Correct?	Ease	Time (s)
1	C	1	a	F	10	11	FALSE	3 38579	38.58
2	C	1	a	F	10	10	TRUE	2 24063	24.06
3	C	1	a	F	10	10	TRUE	0 26579	26.58
4	A	1	a	P	10	10	TRUE	1 17922	17.92
5	B	1	a	G	10	10	TRUE	2 16812	16.81
6	B	1	a	G	10	9	FALSE	2 17703	17.7
7	B	1	a	G	10	10	TRUE	2 27738	27.74
8	A	1	a	P	10	10	TRUE	1 17062	17.06
9	A	1	a	P	10	10	TRUE	3 24296	24.3
10	A	1	a	P	10	10	TRUE	2 27860	27.86
11	C	1	a	F	10	10	TRUE	3 32446	32.45
12	B	1	a	G	10	10	TRUE	2 26486	26.49
13	C	1	a	F	10	10	TRUE	2 25655	25.66
14	A	1	a	P	10	10	TRUE	2 16923	16.92
15	A	1	a	P	10	10	TRUE	1 17798	17.8
16	C	1	a	F	10	10	TRUE	2 18687	18.69
17	B	1	a	G	10	11	FALSE	2 28813	28.81
18	B	1	a	G	10	10	TRUE	0 38893	38.89
19	C	1	a	F	10	10	TRUE	1 16562	16.56
20	C	1	a	F	10	10	TRUE	2 19752	19.75
21	A	1	a	P	10	10	TRUE	1 21173	21.17
22	B	1	a	G	10	10	TRUE	2 15312	15.31

continued on next page

[a] 1 = Atlanta; 2 = Bucharest; 3 = Mexico City; 4 = Stockholm; 5 = Toronto; 6 = Washington D.C.

[b] G = geographic layout; P = normalised published layout; F = automatically-drawn layout

[c] 1 = very easy . . . 5 = very hard

continued from previous page

Candidate	Group	Map	Question	Map Type	Answer	Given Answer	Correct?	Ease	Time (s)
23	A	1	a	P	10	10	TRUE	2 21576	21.58
24	B	1	a	G	10	9	FALSE	1 26284	26.28
25	C	1	a	F	10	10	TRUE	1 14829	14.83
26	B	1	a	G	10	9	FALSE	1 20000	20
27	A	1	a	P	10	10	TRUE	0 34313	34.31
28	B	1	a	G	10	9	FALSE	2 18053	18.05
29	A	1	a	P	10	10	TRUE	2 14939	14.94
30	C	1	a	F	10	10	TRUE	3 19249	19.25
31	A	1	a	P	10	10	TRUE	2 11198	11.2
32	C	1	a	F	10	10	TRUE	2 13486	13.49
33	B	1	a	G	10	11	FALSE	2 17391	17.39
34	B	1	a	G	10	10	TRUE	0 27562	27.56
35	C	1	a	F	10	10	TRUE	3 13391	13.39
36	C	1	a	F	10	11	FALSE	3 18110	18.11
37	B	1	a	G	10	9	FALSE	2 32547	32.55
38	A	1	a	P	10	10	TRUE	2 15198	15.2
39	A	1	a	P	10	12	FALSE	4 15828	15.83
40	B	1	a	G	10	10	TRUE	0 18971	18.97
41	B	1	a	G	10	11	FALSE	2 18281	18.28
42	C	1	a	F	10	10	TRUE	3 20020	20.02
43	C	1	a	F	10	10	TRUE	1 19895	19.9
1	C	1	b	G	5	5	TRUE	2 30657	30.66
2	C	1	b	G	5	5	TRUE	2 18190	18.19
3	C	1	b	G	5	5	TRUE	3 13161	13.16
4	A	1	b	F	5	4	FALSE	4 11047	11.05
5	B	1	b	P	5	5	TRUE	2 16359	16.36

continued on next page

continued from previous page

Candidate	Group	Map	Question	Map Type	Answer	Given Answer	Correct?	Ease	Time (s)
6	B	1	b	P	5	5	TRUE	0 10891	10.89
7	B	1	b	P	5	5	TRUE	1 13720	13.72
8	A	1	b	F	5	5	TRUE	1 9000	9
9	A	1	b	F	5	5	TRUE	2 20578	20.58
10	A	1	b	F	5	5	TRUE	2 19454	19.45
11	C	1	b	G	5	5	TRUE	2 17199	17.2
12	B	1	b	P	5	5	TRUE	2 23626	23.63
13	C	1	b	G	5	5	TRUE	2 10812	10.81
14	A	1	b	F	5	5	TRUE	2 11594	11.59
15	A	1	b	F	5	5	TRUE	2 12204	12.2
16	C	1	b	G	5	5	TRUE	1 17984	17.98
17	B	1	b	P	5	5	TRUE	1 17172	17.17
18	B	1	b	P	5	5	TRUE	1 12103	12.1
19	C	1	b	G	5	5	TRUE	1 19910	19.91
20	C	1	b	G	5	5	TRUE	2 15892	15.89
21	A	1	b	F	5	5	TRUE	1 12123	12.12
22	B	1	b	P	5	5	TRUE	1 9234	9.23
23	A	1	b	F	5	8	FALSE	2 19341	19.34
24	B	1	b	P	5	5	TRUE	1 13236	13.24
25	C	1	b	G	5	5	TRUE	2 9797	9.8
26	B	1	b	P	5	5	TRUE	0 15500	15.5
27	A	1	b	F	5	5	TRUE	1 19208	19.21
28	B	1	b	P	5	5	TRUE	2 10572	10.57
29	A	1	b	F	5	5	TRUE	3 15559	15.56
30	C	1	b	G	5	5	TRUE	3 18296	18.3
31	A	1	b	F	5	5	TRUE	2 10029	10.03

continued on next page

continued from previous page

Candidate	Group	Map	Question	Map Type	Answer	Given Answer	Correct?	Ease	Time (s)
32	C	1	b	G	5	5	TRUE	2 10366	10.37
33	B	1	b	P	5	5	TRUE	1 10560	10.56
34	B	1	b	P	5	5	TRUE	0 14410	14.41
35	C	1	b	G	5	5	TRUE	2 13250	13.25
36	C	1	b	G	5	5	TRUE	1 9579	9.58
37	B	1	b	P	5	5	TRUE	3 12694	12.69
38	A	1	b	F	5	5	TRUE	4 12978	12.98
39	A	1	b	F	5	8	FALSE	4 10406	10.41
40	B	1	b	P	5	5	TRUE	0 8288	8.29
41	B	1	b	P	5	5	TRUE	1 6792	6.79
42	C	1	b	G	5	5	TRUE	1 12992	12.99
43	C	1	b	G	5	5	TRUE	1 15991	15.99
1	C	1	c	F	0	0	TRUE	2 13687	13.69
2	C	1	c	F	0	1	FALSE	2 25266	25.27
3	C	1	c	F	0	0	TRUE	2 26763	26.76
4	A	1	c	P	0	0	TRUE	1 13688	13.69
5	B	1	c	G	0	0	TRUE	0 24875	24.88
6	B	1	c	G	0	0	TRUE	0 25625	25.63
7	B	1	c	G	0	0	TRUE	1 16643	16.64
8	A	1	c	P	0	0	TRUE	1 12078	12.08
9	A	1	c	P	0	0	TRUE	1 11969	11.97
10	A	1	c	P	0	1	FALSE	0 13610	13.61
11	C	1	c	F	0	0	TRUE	3 35592	35.59
12	B	1	c	G	0	1	FALSE	3 86957	86.96
13	C	1	c	F	0	0	TRUE	2 18686	18.69
14	A	1	c	P	0	0	TRUE	1 9454	9.45

continued on next page

continued from previous page

Candidate	Group	Map	Question	Map Type	Answer	Given Answer	Correct?	Ease	Time (s)
15	A	1	c	P	0	0	TRUE	1 8781	8.78
16	C	1	c	F	0	0	TRUE	1 22610	22.61
17	B	1	c	G	0	1	FALSE	3 44188	44.19
18	B	1	c	G	0	1	FALSE	1 29710	29.71
19	C	1	c	F	0	0	TRUE	2 44858	44.86
20	C	1	c	F	0	1	FALSE	1 25580	25.58
21	A	1	c	P	0	0	TRUE	0 32595	32.6
22	B	1	c	G	0	0	TRUE	1 15125	15.13
23	A	1	c	P	0	0	TRUE	2 30732	30.73
24	B	1	c	G	0	0	TRUE	2 51584	51.58
25	C	1	c	F	0	0	TRUE	0 15250	15.25
26	B	1	c	G	0	2	FALSE	0 12219	12.22
27	A	1	c	P	0	0	TRUE	0 11532	11.53
28	B	1	c	G	0	0	TRUE	2 23151	23.15
29	A	1	c	P	0	0	TRUE	1 18001	18
30	C	1	c	F	0	0	TRUE	2 28921	28.92
31	A	1	c	P	0	0	TRUE	1 14540	14.54
32	C	1	c	F	0	0	TRUE	1 14082	14.08
33	B	1	c	G	0	0	TRUE	0 13298	13.3
34	B	1	c	G	0	1	FALSE	1 35027	35.03
35	C	1	c	F	0	0	TRUE	1 18578	18.58
36	C	1	c	F	0	0	TRUE	2 21219	21.22
37	B	1	c	G	0	1	FALSE	2 33626	33.63
38	A	1	c	P	0	0	TRUE	2 14058	14.06
39	A	1	c	P	0	0	TRUE	1 9844	9.84
40	B	1	c	G	0	0	TRUE	0 17449	17.45

continued on next page

continued from previous page

Candidate	Group	Map	Question	Map Type	Answer	Given Answer	Correct?	Ease	Time (s)
41	B	1	c	G	0	0	TRUE	2 15735	15.74
42	C	1	c	F	0	0	TRUE	1 19626	19.63
43	C	1	c	F	0	0	TRUE	1 11978	11.98
1	C	1	d	G	0	0	TRUE	0 9532	9.53
2	C	1	d	G	0	0	TRUE	2 14953	14.95
3	C	1	d	G	0	0	TRUE	3 31438	31.44
4	A	1	d	F	0	0	TRUE	0 13813	13.81
5	B	1	d	P	0	0	TRUE	0 10031	10.03
6	B	1	d	P	0	0	TRUE	0 11390	11.39
7	B	1	d	P	0	0	TRUE	0 18939	18.94
8	A	1	d	F	0	0	TRUE	1 10938	10.94
9	A	1	d	F	0	0	TRUE	1 10312	10.31
10	A	1	d	F	0	0	TRUE	0 11000	11
11	C	1	d	G	0	0	TRUE	1 19540	19.54
12	B	1	d	P	0	0	TRUE	0 29939	29.94
13	C	1	d	G	0	0	TRUE	0 10577	10.58
14	A	1	d	F	0	0	TRUE	0 7641	7.64
15	A	1	d	F	0	0	TRUE	2 11282	11.28
16	C	1	d	G	0	0	TRUE	0 11922	11.92
17	B	1	d	P	0	0	TRUE	0 15047	15.05
18	B	1	d	P	0	1	FALSE	1 17280	17.28
19	C	1	d	G	0	0	TRUE	1 19548	19.55
20	C	1	d	G	0	0	TRUE	2 14533	14.53
21	A	1	d	F	0	0	TRUE	0 13578	13.58
22	B	1	d	P	0	0	TRUE	1 11547	11.55
23	A	1	d	F	0	0	TRUE	1 17263	17.26

continued on next page

continued from previous page

Candidate	Group	Map	Question	Map Type	Answer	Given Answer	Correct?	Ease	Time (s)
24	B	1	d	P	0	0	TRUE	0 14768	14.77
25	C	1	d	G	0	0	TRUE	0 8735	8.74
26	B	1	d	P	0	0	TRUE	0 16984	16.98
27	A	1	d	F	0	0	TRUE	0 14007	14.01
28	B	1	d	P	0	0	TRUE	2 13164	13.16
29	A	1	d	F	0	0	TRUE	1 16081	16.08
30	C	1	d	G	0	0	TRUE	1 15203	15.2
31	A	1	d	F	0	0	TRUE	1 10466	10.47
32	C	1	d	G	0	0	TRUE	1 14399	14.4
33	B	1	d	P	0	0	TRUE	0 6000	6
34	B	1	d	P	0	0	TRUE	0 14457	14.46
35	C	1	d	G	0	0	TRUE	0 13000	13
36	C	1	d	G	0	0	TRUE	0 10844	10.84
37	B	1	d	P	0	0	TRUE	3 19811	19.81
38	A	1	d	F	0	0	TRUE	2 13122	13.12
39	A	1	d	F	0	0	TRUE	1 20562	20.56
40	B	1	d	P	0	0	TRUE	0 18543	18.54
41	B	1	d	P	0	0	TRUE	0 5734	5.73
42	C	1	d	G	0	0	TRUE	0 13726	13.73
43	C	1	d	G	0	0	TRUE	0 12758	12.76
1	C	2	a	G	10	10	TRUE	2 32375	32.38
2	C	2	a	G	10	9	FALSE	2 37407	37.41
3	C	2	a	G	10	9	FALSE	3 24210	24.21
4	A	2	a	F	10	10	TRUE	2 41595	41.6
5	B	2	a	P	10	10	TRUE	2 15032	15.03
6	B	2	a	P	10	10	TRUE	1 13813	13.81

continued on next page

continued from previous page

Candidate	Group	Map	Question	Map Type	Answer	Given Answer	Correct?	Ease	Time (s)
7	B	2	a	P	10	10	TRUE	2 29503	29.5
8	A	2	a	F	10	10	TRUE	2 34095	34.1
9	A	2	a	F	10	10	TRUE	3 66825	66.83
10	A	2	a	F	10	10	TRUE	2 34360	34.36
11	C	2	a	G	10	9	FALSE	3 37264	37.26
12	B	2	a	P	10	10	TRUE	2 31548	31.55
13	C	2	a	G	10	10	TRUE	3 21749	21.75
14	A	2	a	F	10	10	TRUE	3 39017	39.02
15	A	2	a	F	10	10	TRUE	1 13329	13.33
16	C	2	a	G	10	9	FALSE	1 15563	15.56
17	B	2	a	P	10	10	TRUE	2 16609	16.61
18	B	2	a	P	10	10	TRUE	0 57441	57.44
19	C	2	a	G	10	10	TRUE	2 43826	43.83
20	C	2	a	G	10	9	FALSE	2 13532	13.53
21	A	2	a	F	10	10	TRUE	3 28844	28.84
22	B	2	a	P	10	10	TRUE	2 13203	13.2
23	A	2	a	F	10	10	TRUE	3 50807	50.81
24	B	2	a	P	10	10	TRUE	2 13720	13.72
25	C	2	a	G	10	9	FALSE	1 23656	23.66
26	B	2	a	P	10	10	TRUE	2 23531	23.53
27	A	2	a	F	10	10	TRUE	0 45080	45.08
28	B	2	a	P	10	10	TRUE	2 16725	16.73
29	A	2	a	F	10	11	FALSE	3 25425	25.43
30	C	2	a	G	10	9	FALSE	3 18765	18.77
31	A	2	a	F	10	10	TRUE	3 35234	35.23
32	C	2	a	G	10	10	TRUE	2 18104	18.1

continued on next page

continued from previous page

Candidate	Group	Map	Question	Map Type	Answer	Given Answer	Correct?	Ease	Time (s)
33	B	2	a	P	10	10	TRUE	2 11438	11.44
34	B	2	a	P	10	11	FALSE	0 51632	51.63
35	C	2	a	G	10	9	FALSE	2 37016	37.02
36	C	2	a	G	10	10	TRUE	3 23766	23.77
37	B	2	a	P	10	10	TRUE	2 12970	12.97
38	A	2	a	F	10	10	TRUE	2 10454	10.45
39	A	2	a	F	10	10	TRUE	2 12188	12.19
40	B	2	a	P	10	10	TRUE	0 21254	21.25
41	B	2	a	P	10	11	FALSE	2 18172	18.17
42	C	2	a	G	10	10	TRUE	3 16732	16.73
43	C	2	a	G	10	9	FALSE	1 22768	22.77
1	C	2	b	P	5	5	TRUE	1 13375	13.38
2	C	2	b	P	5	5	TRUE	2 17331	17.33
3	C	2	b	P	5	5	TRUE	3 18775	18.78
4	A	2	b	G	5	5	TRUE	0 9812	9.81
5	B	2	b	F	5	5	TRUE	2 10015	10.02
6	B	2	b	F	5	5	TRUE	2 10625	10.63
7	B	2	b	F	5	5	TRUE	2 27347	27.35
8	A	2	b	G	5	5	TRUE	1 15110	15.11
9	A	2	b	G	5	5	TRUE	2 22171	22.17
10	A	2	b	G	5	5	TRUE	1 9265	9.27
11	C	2	b	P	5	5	TRUE	1 20679	20.68
12	B	2	b	F	5	5	TRUE	2 32689	32.69
13	C	2	b	P	5	5	TRUE	1 20233	20.23
14	A	2	b	G	5	5	TRUE	2 9641	9.64
15	A	2	b	G	5	5	TRUE	2 14251	14.25

continued on next page

continued from previous page

Candidate	Group	Map	Question	Map Type	Answer	Given Answer	Correct?	Ease	Time (s)
16	C	2	b	P	5	5	TRUE	0 9828	9.83
17	B	2	b	F	5	5	TRUE	1 21172	21.17
18	B	2	b	F	5	5	TRUE	2 16968	16.97
19	C	2	b	P	5	5	TRUE	1 26044	26.04
20	C	2	b	P	5	5	TRUE	2 13595	13.6
21	A	2	b	G	5	3	FALSE	3 25307	25.31
22	B	2	b	F	5	5	TRUE	1 8297	8.3
23	A	2	b	G	5	5	TRUE	1 34797	34.8
24	B	2	b	F	5	5	TRUE	1 10860	10.86
25	C	2	b	P	5	5	TRUE	1 10048	10.05
26	B	2	b	F	5	5	TRUE	0 14203	14.2
27	A	2	b	G	5	5	TRUE	2 18395	18.4
28	B	2	b	F	5	4	FALSE	2 8951	8.95
29	A	2	b	G	5	5	TRUE	2 21573	21.57
30	C	2	b	P	5	5	TRUE	1 10671	10.67
31	A	2	b	G	5	5	TRUE	3 32849	32.85
32	C	2	b	P	5	5	TRUE	1 14696	14.7
33	B	2	b	F	5	5	TRUE	1 12670	12.67
34	B	2	b	F	5	5	TRUE	0 30664	30.66
35	C	2	b	P	5	5	TRUE	1 16516	16.52
36	C	2	b	P	5	5	TRUE	1 7375	7.38
37	B	2	b	F	5	5	TRUE	2 9587	9.59
38	A	2	b	G	5	5	TRUE	3 9995	10
39	A	2	b	G	5	5	TRUE	4 20250	20.25
40	B	2	b	F	5	5	TRUE	0 10270	10.27
41	B	2	b	F	5	5	TRUE	1 7792	7.79

continued on next page

continued from previous page

Candidate	Group	Map	Question	Map Type	Answer	Given Answer	Correct?	Ease	Time (s)
42	C	2	b	P	5	5	TRUE	1 19565	19.57
43	C	2	b	P	5	5	TRUE	1 11884	11.88
1	C	2	c	G	0	0	TRUE	1 12750	12.75
2	C	2	c	G	0	0	TRUE	2 26016	26.02
3	C	2	c	G	0	0	TRUE	2 22146	22.15
4	A	2	c	F	0	0	TRUE	0 14126	14.13
5	B	2	c	P	0	0	TRUE	1 19094	19.09
6	B	2	c	P	0	0	TRUE	0 22813	22.81
7	B	2	c	P	0	0	TRUE	1 34254	34.25
8	A	2	c	F	0	0	TRUE	1 12953	12.95
9	A	2	c	F	0	0	TRUE	1 29656	29.66
10	A	2	c	F	0	0	TRUE	1 12828	12.83
11	C	2	c	G	0	0	TRUE	3 48669	48.67
12	B	2	c	P	0	0	TRUE	2 52502	52.5
13	C	2	c	G	0	2	FALSE	3 23827	23.83
14	A	2	c	F	0	0	TRUE	1 8032	8.03
15	A	2	c	F	0	0	TRUE	1 18673	18.67
16	C	2	c	G	0	0	TRUE	0 9640	9.64
17	B	2	c	P	0	0	TRUE	2 35501	35.5
18	B	2	c	P	0	3	FALSE	2 45700	45.7
19	C	2	c	G	0	0	TRUE	2 27765	27.77
20	C	2	c	G	0	0	TRUE	1 19205	19.21
21	A	2	c	F	0	0	TRUE	1 17859	17.86
22	B	2	c	P	0	0	TRUE	1 22515	22.52
23	A	2	c	F	0	0	TRUE	2 44151	44.15
24	B	2	c	P	0	1	FALSE	3 42817	42.82

continued on next page

continued from previous page

Candidate	Group	Map	Question	Map Type	Answer	Given Answer	Correct?	Ease	Time (s)
25	C	2	c	G	0	0	TRUE	1 10875	10.88
26	B	2	c	P	0	2	FALSE	1 12765	12.77
27	A	2	c	F	0	0	TRUE	0 10422	10.42
28	B	2	c	P	0	0	TRUE	2 34986	34.99
29	A	2	c	F	0	0	TRUE	1 12673	12.67
30	C	2	c	G	0	0	TRUE	2 19406	19.41
31	A	2	c	F	0	0	TRUE	1 16617	16.62
32	C	2	c	G	0	0	TRUE	1 9413	9.41
33	B	2	c	P	0	0	TRUE	0 11032	11.03
34	B	2	c	P	0	0	TRUE	0 38601	38.6
35	C	2	c	G	0	0	TRUE	1 18125	18.13
36	C	2	c	G	0	0	TRUE	2 23594	23.59
37	B	2	c	P	0	0	TRUE	1 12625	12.63
38	A	2	c	F	0	0	TRUE	3 24248	24.25
39	A	2	c	F	0	0	TRUE	1 9234	9.23
40	B	2	c	P	0	1	FALSE	0 14496	14.5
41	B	2	c	P	0	0	TRUE	2 14719	14.72
42	C	2	c	G	0	0	TRUE	1 29346	29.35
43	C	2	c	G	0	0	TRUE	0 7277	7.28
1	C	2	d	P	2	3	FALSE	1 15890	15.89
2	C	2	d	P	2	2	TRUE	2 23313	23.31
3	C	2	d	P	2	2	TRUE	2 20604	20.6
4	A	2	d	G	2	2	TRUE	0 15094	15.09
5	B	2	d	F	2	2	TRUE	2 15813	15.81
6	B	2	d	F	2	2	TRUE	1 10141	10.14
7	B	2	d	F	2	2	TRUE	2 24581	24.58

continued on next page

continued from previous page

Candidate	Group	Map	Question	Map Type	Answer	Given Answer	Correct?	Ease	Time (s)
8	A	2	d	G	2	2	TRUE	1 13797	13.8
9	A	2	d	G	2	2	TRUE	2 31265	31.27
10	A	2	d	G	2	2	TRUE	3 37782	37.78
11	C	2	d	P	2	2	TRUE	2 26703	26.7
12	B	2	d	F	2	3	FALSE	2 42455	42.46
13	C	2	d	P	2	2	TRUE	2 27951	27.95
14	A	2	d	G	2	2	TRUE	2 17313	17.31
15	A	2	d	G	2	2	TRUE	2 9548	9.55
16	C	2	d	P	2	2	TRUE	1 15188	15.19
17	B	2	d	F	2	2	TRUE	1 19297	19.3
18	B	2	d	F	2	3	FALSE	1 15238	15.24
19	C	2	d	P	2	2	TRUE	2 44143	44.14
20	C	2	d	P	2	2	TRUE	2 23815	23.82
21	A	2	d	G	2	2	TRUE	2 25594	25.59
22	B	2	d	F	2	2	TRUE	1 14047	14.05
23	A	2	d	G	2	2	TRUE	3 56598	56.6
24	B	2	d	F	2	2	TRUE	1 14548	14.55
25	C	2	d	P	2	2	TRUE	0 11375	11.38
26	B	2	d	F	2	2	TRUE	1 21187	21.19
27	A	2	d	G	2	2	TRUE	0 22128	22.13
28	B	2	d	F	2	2	TRUE	2 19927	19.93
29	A	2	d	G	2	2	TRUE	2 19704	19.7
30	C	2	d	P	2	2	TRUE	2 22343	22.34
31	A	2	d	G	2	2	TRUE	2 19560	19.56
32	C	2	d	P	2	2	TRUE	2 14508	14.51
33	B	2	d	F	2	2	TRUE	2 20048	20.05

continued on next page

continued from previous page

Candidate	Group	Map	Question	Map Type	Answer	Given Answer	Correct?	Ease	Time (s)
34	B	2	d	F	2	2	TRUE	2 35457	35.46
35	C	2	d	P	2	2	TRUE	1 16719	16.72
36	C	2	d	P	2	3	FALSE	2 21625	21.63
37	B	2	d	F	2	2	TRUE	3 21407	21.41
38	A	2	d	G	2	2	TRUE	2 14230	14.23
39	A	2	d	G	2	2	TRUE	2 16500	16.5
40	B	2	d	F	2	2	TRUE	0 16233	16.23
41	B	2	d	F	2	2	TRUE	1 11750	11.75
42	C	2	d	P	2	2	TRUE	2 13540	13.54
43	C	2	d	P	2	2	TRUE	1 19521	19.52
1	C	3	a	P	7	8	FALSE	2 25187	25.19
2	C	3	a	P	7	7	TRUE	2 76922	76.92
3	C	3	a	P	7	6	FALSE	3 22821	22.82
4	A	3	a	G	7	7	TRUE	0 22188	22.19
5	B	3	a	F	7	7	TRUE	2 14406	14.41
6	B	3	a	F	7	6	FALSE	2 20641	20.64
7	B	3	a	F	7	7	TRUE	2 23112	23.11
8	A	3	a	G	7	7	TRUE	2 17735	17.74
9	A	3	a	G	7	7	TRUE	3 28702	28.7
10	A	3	a	G	7	7	TRUE	2 29110	29.11
11	C	3	a	P	7	7	TRUE	4 23747	23.75
12	B	3	a	F	7	6	FALSE	2 26048	26.05
13	C	3	a	P	7	6	FALSE	3 17265	17.27
14	A	3	a	G	7	7	TRUE	3 25674	25.67
15	A	3	a	G	7	7	TRUE	1 16329	16.33
16	C	3	a	P	7	5	FALSE	1 26063	26.06

continued on next page

continued from previous page

Candidate	Group	Map	Question	Map Type	Answer	Given Answer	Correct?	Ease	Time (s)
17	B	3	a	F	7	7	TRUE	2 37985	37.99
18	B	3	a	F	7	6	FALSE	2 59470	59.47
19	C	3	a	P	7	6	FALSE	2 50612	50.61
20	C	3	a	P	7	7	TRUE	2 18549	18.55
21	A	3	a	G	7	7	TRUE	3 34970	34.97
22	B	3	a	F	7	7	TRUE	2 22000	22
23	A	3	a	G	7	7	TRUE	2 33778	33.78
24	B	3	a	F	7	7	TRUE	2 54943	54.94
25	C	3	a	P	7	7	TRUE	1 34969	34.97
26	B	3	a	F	7	7	TRUE	2 42171	42.17
27	A	3	a	G	7	8	FALSE	1 37191	37.19
28	B	3	a	F	7	7	TRUE	2 15663	15.66
29	A	3	a	G	7	7	TRUE	3 16893	16.89
30	C	3	a	P	7	7	TRUE	3 18609	18.61
31	A	3	a	G	7	7	TRUE	3 29127	29.13
32	C	3	a	P	7	8	FALSE	2 16197	16.2
33	B	3	a	F	7	7	TRUE	3 39032	39.03
34	B	3	a	F	7	6	FALSE	0 25917	25.92
35	C	3	a	P	7	7	TRUE	2 19266	19.27
36	C	3	a	P	7	6	FALSE	3 24985	24.99
37	B	3	a	F	7	7	TRUE	3 13845	13.85
38	A	3	a	G	7	5	FALSE	2 13028	13.03
39	A	3	a	G	7	7	TRUE	2 9313	9.31
40	B	3	a	F	7	6	FALSE	0 21191	21.19
41	B	3	a	F	7	6	FALSE	1 15735	15.74
42	C	3	a	P	7	7	TRUE	4 18031	18.03

continued on next page

continued from previous page

Candidate	Group	Map	Question	Map Type	Answer	Given Answer	Correct?	Ease	Time (s)
43	C	3	a	P	7	7	TRUE	1 16835	16.84
1	C	3	b	F	6	6	TRUE	2 16891	16.89
2	C	3	b	F	6	6	TRUE	2 30977	30.98
3	C	3	b	F	6	6	TRUE	3 19602	19.6
4	A	3	b	P	6	5	FALSE	0 11844	11.84
5	B	3	b	G	6	6	TRUE	1 15593	15.59
6	B	3	b	G	6	6	TRUE	2 12046	12.05
7	B	3	b	G	6	6	TRUE	1 16298	16.3
8	A	3	b	P	6	6	TRUE	1 11859	11.86
9	A	3	b	P	6	6	TRUE	2 23046	23.05
10	A	3	b	P	6	6	TRUE	1 12422	12.42
11	C	3	b	F	6	6	TRUE	2 12252	12.25
12	B	3	b	G	6	6	TRUE	2 34392	34.39
13	C	3	b	F	6	6	TRUE	3 18249	18.25
14	A	3	b	P	6	6	TRUE	1 13704	13.7
15	A	3	b	P	6	6	TRUE	2 13516	13.52
16	C	3	b	F	6	8	FALSE	2 26547	26.55
17	B	3	b	G	6	6	TRUE	2 20406	20.41
18	B	3	b	G	6	6	TRUE	2 12415	12.42
19	C	3	b	F	6	5	FALSE	2 41377	41.38
20	C	3	b	F	6	6	TRUE	2 12689	12.69
21	A	3	b	P	6	6	TRUE	1 17809	17.81
22	B	3	b	G	6	6	TRUE	2 9469	9.47
23	A	3	b	P	6	7	FALSE	1 16954	16.95
24	B	3	b	G	6	6	TRUE	1 15939	15.94
25	C	3	b	F	6	6	TRUE	1 17234	17.23

continued on next page

continued from previous page

Candidate	Group	Map	Question	Map Type	Answer	Given Answer	Correct?	Ease	Time (s)
26	B	3	b	G	6	6	TRUE	1 12906	12.91
27	A	3	b	P	6	6	TRUE	1 16068	16.07
28	B	3	b	G	6	6	TRUE	2 11498	11.5
29	A	3	b	P	6	6	TRUE	2 11263	11.26
30	C	3	b	F	6	6	TRUE	1 15734	15.73
31	A	3	b	P	6	6	TRUE	2 18483	18.48
32	C	3	b	F	6	6	TRUE	2 12802	12.8
33	B	3	b	G	6	6	TRUE	1 7436	7.44
34	B	3	b	G	6	6	TRUE	0 51387	51.39
35	C	3	b	F	6	6	TRUE	1 11813	11.81
36	C	3	b	F	6	6	TRUE	1 10563	10.56
37	B	3	b	G	6	6	TRUE	3 11711	11.71
38	A	3	b	P	6	6	TRUE	3 11698	11.7
39	A	3	b	P	6	6	TRUE	3 11984	11.98
40	B	3	b	G	6	6	TRUE	0 15468	15.47
41	B	3	b	G	6	6	TRUE	2 12211	12.21
42	C	3	b	F	6	6	TRUE	3 17123	17.12
43	C	3	b	F	6	6	TRUE	1 13930	13.93
1	C	3	c	P	2	2	TRUE	3 27188	27.19
2	C	3	c	P	2	2	TRUE	3 129642	129.64
3	C	3	c	P	2	1	FALSE	2 20675	20.68
4	A	3	c	G	2	1	FALSE	1 14047	14.05
5	B	3	c	F	2	1	FALSE	1 17562	17.56
6	B	3	c	F	2	2	TRUE	2 18234	18.23
7	B	3	c	F	2	1	FALSE	2 23768	23.77
8	A	3	c	G	2	1	FALSE	1 14172	14.17

continued on next page

continued from previous page

Candidate	Group	Map	Question	Map Type	Answer	Given Answer	Correct?	Ease	Time (s)
9	A	3	c	G	2	2	TRUE	2 21624	21.62
10	A	3	c	G	2	1	FALSE	1 11923	11.92
11	C	3	c	P	2	2	TRUE	2 51455	51.46
12	B	3	c	F	2	2	TRUE	3 45049	45.05
13	C	3	c	P	2	2	TRUE	1 24842	24.84
14	A	3	c	G	2	1	FALSE	2 23501	23.5
15	A	3	c	G	2	3	FALSE	2 13079	13.08
16	C	3	c	P	2	2	TRUE	1 22391	22.39
17	B	3	c	F	2	1	FALSE	1 36313	36.31
18	B	3	c	F	2	1	FALSE	0 28614	28.61
19	C	3	c	P	2	1	FALSE	2 26906	26.91
20	C	3	c	P	2	2	TRUE	2 17595	17.6
21	A	3	c	G	2	1	FALSE	2 26407	26.41
22	B	3	c	F	2	1	FALSE	1 18937	18.94
23	A	3	c	G	2	2	TRUE	2 37693	37.69
24	B	3	c	F	2	2	TRUE	2 42567	42.57
25	C	3	c	P	2	1	FALSE	1 15484	15.48
26	B	3	c	F	2	2	TRUE	2 170637	170.64
27	A	3	c	G	2	1	FALSE	2 38485	38.49
28	B	3	c	F	2	2	TRUE	2 40189	40.19
29	A	3	c	G	2	1	FALSE	2 21893	21.89
30	C	3	c	P	2	1	FALSE	2 23999	24
31	A	3	c	G	2	1	FALSE	2 28065	28.07
32	C	3	c	P	2	1	FALSE	1 13138	13.14
33	B	3	c	F	2	2	TRUE	2 41767	41.77
34	B	3	c	F	2	1	FALSE	0 30146	30.15

continued on next page

continued from previous page

Candidate	Group	Map	Question	Map Type	Answer	Given Answer	Correct?	Ease	Time (s)
35	C	3	c	P	2	1	FALSE	2 22735	22.74
36	C	3	c	P	2	1	FALSE	2 33172	33.17
37	B	3	c	F	2	1	FALSE	1 19938	19.94
38	A	3	c	G	2	1	FALSE	2 13903	13.9
39	A	3	c	G	2	2	TRUE	2 14656	14.66
40	B	3	c	F	2	2	TRUE	0 12781	12.78
41	B	3	c	F	2	2	TRUE	2 22735	22.74
42	C	3	c	P	2	1	FALSE	2 15123	15.12
43	C	3	c	P	2	1	FALSE	2 17928	17.93
1	C	3	d	F	2	3	FALSE	2 53734	53.73
2	C	3	d	F	2	3	FALSE	2 30957	30.96
3	C	3	d	F	2	2	TRUE	3 30392	30.39
4	A	3	d	P	2	2	TRUE	0 15422	15.42
5	B	3	d	G	2	3	FALSE	1 14735	14.74
6	B	3	d	G	2	2	TRUE	2 18250	18.25
7	B	3	d	G	2	2	TRUE	2 19627	19.63
8	A	3	d	P	2	2	TRUE	2 23250	23.25
9	A	3	d	P	2	2	TRUE	2 39764	39.76
10	A	3	d	P	2	1	FALSE	4 42126	42.13
11	C	3	d	F	2	2	TRUE	2 18370	18.37
12	B	3	d	G	2	3	FALSE	2 41798	41.8
13	C	3	d	F	2	2	TRUE	2 29717	29.72
14	A	3	d	P	2	3	FALSE	2 28735	28.74
15	A	3	d	P	2	2	TRUE	2 12923	12.92
16	C	3	d	F	2	2	TRUE	0 23954	23.95
17	B	3	d	G	2	2	TRUE	4 66032	66.03

continued on next page

continued from previous page

Candidate	Group	Map	Question	Map Type	Answer	Given Answer	Correct?	Ease	Time (s)
18	B	3	d	G	2	4	FALSE	2 16001	16
19	C	3	d	F	2	2	TRUE	1 46800	46.8
20	C	3	d	F	2	3	FALSE	2 36519	36.52
21	A	3	d	P	2	2	TRUE	2 33891	33.89
22	B	3	d	G	2	2	TRUE	1 14469	14.47
23	A	3	d	P	2	2	TRUE	2 28403	28.4
24	B	3	d	G	2	3	FALSE	1 18908	18.91
25	C	3	d	F	2	2	TRUE	0 24297	24.3
26	B	3	d	G	2	2	TRUE	1 26874	26.87
27	A	3	d	P	2	2	TRUE	0 40946	40.95
28	B	3	d	G	2	1	FALSE	2 27438	27.44
29	A	3	d	P	2	2	TRUE	3 34380	34.38
30	C	3	d	F	2	2	TRUE	1 26811	26.81
31	A	3	d	P	2	2	TRUE	2 25674	25.67
32	C	3	d	F	2	2	TRUE	3 19589	19.59
33	B	3	d	G	2	2	TRUE	3 26672	26.67
34	B	3	d	G	2	2	TRUE	2 67619	67.62
35	C	3	d	F	2	2	TRUE	1 21688	21.69
36	C	3	d	F	2	2	TRUE	2 12516	12.52
37	B	3	d	G	2	2	TRUE	3 21595	21.6
38	A	3	d	P	2	3	FALSE	2 19082	19.08
39	A	3	d	P	2	2	TRUE	2 21608	21.61
40	B	3	d	G	2	2	TRUE	1 24927	24.93
41	B	3	d	G	2	2	TRUE	3 19938	19.94
42	C	3	d	F	2	2	TRUE	2 18923	18.92
43	C	3	d	F	2	2	TRUE	2 20333	20.33

continued on next page

continued from previous page

Candidate	Group	Map	Question	Map Type	Answer	Given Answer	Correct?	Ease	Time (s)
1	C	4	a	F	10	10	TRUE	2 31125	31.13
2	C	4	a	F	10	7	FALSE	2 101923	101.92
3	C	4	a	F	10	10	TRUE	4 21104	21.1
4	A	4	a	P	10	10	TRUE	0 15391	15.39
5	B	4	a	G	10	7	FALSE	2 16922	16.92
6	B	4	a	G	10	7	FALSE	3 14890	14.89
7	B	4	a	G	10	6	FALSE	2 23424	23.42
8	A	4	a	P	10	9	FALSE	3 29188	29.19
9	A	4	a	P	10	7	FALSE	3 56169	56.17
10	A	4	a	P	10	9	FALSE	3 31798	31.8
11	C	4	a	F	10	7	FALSE	2 30897	30.9
12	B	4	a	G	10	7	FALSE	2 26720	26.72
13	C	4	a	F	10	7	FALSE	2 20358	20.36
14	A	4	a	P	10	9	FALSE	2 25689	25.69
15	A	4	a	P	10	7	FALSE	2 24439	24.44
16	C	4	a	F	10	10	TRUE	1 17828	17.83
17	B	4	a	G	10	7	FALSE	3 28360	28.36
18	B	4	a	G	10	7	FALSE	2 18361	18.36
19	C	4	a	F	10	7	FALSE	1 41081	41.08
20	C	4	a	F	10	7	FALSE	2 19268	19.27
21	A	4	a	P	10	9	FALSE	2 16891	16.89
22	B	4	a	G	10	7	FALSE	2 23218	23.22
23	A	4	a	P	10	8	FALSE	3 54396	54.4
24	B	4	a	G	10	7	FALSE	1 25596	25.6
25	C	4	a	F	10	6	FALSE	1 13766	13.77
26	B	4	a	G	10	7	FALSE	2 31359	31.36

continued on next page

continued from previous page

Candidate	Group	Map	Question	Map Type	Answer	Given Answer	Correct?	Ease	Time (s)
27	A	4	a	P	10	10	TRUE	2 34112	34.11
28	B	4	a	G	10	7	FALSE	2 26032	26.03
29	A	4	a	P	10	10	TRUE	3 15129	15.13
30	C	4	a	F	10	7	FALSE	3 21249	21.25
31	A	4	a	P	10	10	TRUE	4 25098	25.1
32	C	4	a	F	10	7	FALSE	3 22905	22.91
33	B	4	a	G	10	7	FALSE	2 18141	18.14
34	B	4	a	G	10	7	FALSE	1 34719	34.72
35	C	4	a	F	10	7	FALSE	2 21423	21.42
36	C	4	a	F	10	7	FALSE	3 35720	35.72
37	B	4	a	G	10	8	FALSE	3 13939	13.94
38	A	4	a	P	10	9	FALSE	3 18427	18.43
39	A	4	a	P	10	7	FALSE	2 23375	23.38
40	B	4	a	G	10	7	FALSE	0 15674	15.67
41	B	4	a	G	10	7	FALSE	2 10704	10.7
42	C	4	a	F	10	7	FALSE	3 18720	18.72
43	C	4	a	F	10	7	FALSE	2 27610	27.61
1	C	4	b	G	6	5	FALSE	2 17579	17.58
2	C	4	b	G	6	5	FALSE	2 16332	16.33
3	C	4	b	G	6	4	FALSE	2 10885	10.89
4	A	4	b	F	6	6	TRUE	0 7438	7.44
5	B	4	b	P	6	6	TRUE	3 20313	20.31
6	B	4	b	P	6	6	TRUE	1 11000	11
7	B	4	b	P	6	6	TRUE	2 33207	33.21
8	A	4	b	F	6	6	TRUE	2 19813	19.81
9	A	4	b	F	6	6	TRUE	2 12327	12.33

continued on next page

continued from previous page

Candidate	Group	Map	Question	Map Type	Answer	Given Answer	Correct?	Ease	Time (s)
10	A	4	b	F	6	6	TRUE	2 15954	15.95
11	C	4	b	G	6	5	FALSE	3 25850	25.85
12	B	4	b	P	6	6	TRUE	4 82645	82.65
13	C	4	b	G	6	4	FALSE	2 10640	10.64
14	A	4	b	F	6	6	TRUE	2 10313	10.31
15	A	4	b	F	6	6	TRUE	1 9375	9.38
16	C	4	b	G	6	4	FALSE	0 16531	16.53
17	B	4	b	P	6	6	TRUE	3 23328	23.33
18	B	4	b	P	6	5	FALSE	1 10372	10.37
19	C	4	b	G	6	5	FALSE	1 19471	19.47
20	C	4	b	G	6	5	FALSE	1 12408	12.41
21	A	4	b	F	6	6	TRUE	1 12216	12.22
22	B	4	b	P	6	6	TRUE	3 22500	22.5
23	A	4	b	F	6	6	TRUE	2 33550	33.55
24	B	4	b	P	6	2	FALSE	2 26331	26.33
25	C	4	b	G	6	5	FALSE	1 8844	8.84
26	B	4	b	P	6	6	TRUE	2 32078	32.08
27	A	4	b	F	6	6	TRUE	1 16225	16.23
28	B	4	b	P	6	6	TRUE	2 9373	9.37
29	A	4	b	F	6	6	TRUE	3 7483	7.48
30	C	4	b	G	6	5	FALSE	3 11219	11.22
31	A	4	b	F	6	6	TRUE	3 10637	10.64
32	C	4	b	G	6	5	FALSE	2 11610	11.61
33	B	4	b	P	6	6	TRUE	2 14076	14.08
34	B	4	b	P	6	6	TRUE	4 65719	65.72
35	C	4	b	G	6	5	FALSE	1 15907	15.91

continued on next page

continued from previous page

Candidate	Group	Map	Question	Map Type	Answer	Given Answer	Correct?	Ease	Time (s)
36	C	4	b	G	6	5	FALSE	1 9156	9.16
37	B	4	b	P	6	6	TRUE	2 25030	25.03
38	A	4	b	F	6	6	TRUE	2 11307	11.31
39	A	4	b	F	6	6	TRUE	2 8438	8.44
40	B	4	b	P	6	6	TRUE	0 25911	25.91
41	B	4	b	P	6	6	TRUE	3 28762	28.76
42	C	4	b	G	6	5	FALSE	4 14121	14.12
43	C	4	b	G	6	4	FALSE	1 8635	8.64
1	C	4	c	F	0	0	TRUE	2 21704	21.7
2	C	4	c	F	0	1	FALSE	2 28204	28.2
3	C	4	c	F	0	0	TRUE	3 20112	20.11
4	A	4	c	P	0	0	TRUE	0 10313	10.31
5	B	4	c	G	0	0	TRUE	1 9953	9.95
6	B	4	c	G	0	0	TRUE	2 10468	10.47
7	B	4	c	G	0	0	TRUE	1 16205	16.21
8	A	4	c	P	0	0	TRUE	1 9438	9.44
9	A	4	c	P	0	0	TRUE	2 15327	15.33
10	A	4	c	P	0	0	TRUE	0 8313	8.31
11	C	4	c	F	0	1	FALSE	3 37823	37.82
12	B	4	c	G	0	0	TRUE	2 26095	26.1
13	C	4	c	F	0	0	TRUE	1 23311	23.31
14	A	4	c	P	0	0	TRUE	0 6969	6.97
15	A	4	c	P	0	0	TRUE	1 12063	12.06
16	C	4	c	F	0	0	TRUE	1 39438	39.44
17	B	4	c	G	0	1	FALSE	3 35938	35.94
18	B	4	c	G	0	2	FALSE	2 22386	22.39

continued on next page

continued from previous page

Candidate	Group	Map	Question	Map Type	Answer	Given Answer	Correct?	Ease	Time (s)
19	C	4	c	F	0	0	TRUE	2 21203	21.2
20	C	4	c	F	0	0	TRUE	2 22283	22.28
21	A	4	c	P	0	0	TRUE	1 11875	11.88
22	B	4	c	G	0	0	TRUE	1 13187	13.19
23	A	4	c	P	0	0	TRUE	2 18065	18.07
24	B	4	c	G	0	0	TRUE	2 16861	16.86
25	C	4	c	F	0	1	FALSE	2 46328	46.33
26	B	4	c	G	0	0	TRUE	1 17437	17.44
27	A	4	c	P	0	0	TRUE	0 14562	14.56
28	B	4	c	G	0	0	TRUE	2 16938	16.94
29	A	4	c	P	0	0	TRUE	1 6579	6.58
30	C	4	c	F	0	0	TRUE	2 17030	17.03
31	A	4	c	P	0	0	TRUE	1 9433	9.43
32	C	4	c	F	0	0	TRUE	1 6082	6.08
33	B	4	c	G	0	0	TRUE	0 9923	9.92
34	B	4	c	G	0	0	TRUE	1 31993	31.99
35	C	4	c	F	0	0	TRUE	1 10500	10.5
36	C	4	c	F	0	0	TRUE	2 26844	26.84
37	B	4	c	G	0	0	TRUE	1 11704	11.7
38	A	4	c	P	0	0	TRUE	2 10828	10.83
39	A	4	c	P	0	0	TRUE	2 10656	10.66
40	B	4	c	G	0	0	TRUE	0 11377	11.38
41	B	4	c	G	0	0	TRUE	2 16204	16.2
42	C	4	c	F	0	0	TRUE	1 9862	9.86
43	C	4	c	F	0	0	TRUE	0 6902	6.9
1	C	4	d	G	1	3	FALSE	3 24797	24.8

continued on next page

continued from previous page

Candidate	Group	Map	Question	Map Type	Answer	Given Answer	Correct?	Ease	Time (s)
2	C	4	d	G	1	1	TRUE	2 28323	28.32
3	C	4	d	G	1	1	TRUE	3 21275	21.28
4	A	4	d	F	1	3	FALSE	2 13922	13.92
5	B	4	d	P	1	1	TRUE	2 37469	37.47
6	B	4	d	P	1	1	TRUE	2 16125	16.13
7	B	4	d	P	1	1	TRUE	2 57615	57.62
8	A	4	d	F	1	1	TRUE	2 12578	12.58
9	A	4	d	F	1	1	TRUE	2 35233	35.23
10	A	4	d	F	1	1	TRUE	2 16110	16.11
11	C	4	d	G	1	2	FALSE	3 38268	38.27
12	B	4	d	P	1	2	FALSE	3 95473	95.47
13	C	4	d	G	1	1	TRUE	2 17249	17.25
14	A	4	d	F	1	1	TRUE	2 35205	35.21
15	A	4	d	F	1	1	TRUE	1 13828	13.83
16	C	4	d	G	1	1	TRUE	0 20047	20.05
17	B	4	d	P	1	1	TRUE	4 39938	39.94
18	B	4	d	P	1	2	FALSE	1 12134	12.13
19	C	4	d	G	1	1	TRUE	1 17564	17.56
20	C	4	d	G	1	1	TRUE	2 14454	14.45
21	A	4	d	F	1	1	TRUE	2 15657	15.66
22	B	4	d	P	1	1	TRUE	3 39625	39.63
23	A	4	d	F	1	1	TRUE	3 59894	59.89
24	B	4	d	P	1	1	TRUE	0 21065	21.07
25	C	4	d	G	1	1	TRUE	0 8672	8.67
26	B	4	d	P	1	1	TRUE	1 48655	48.66
27	A	4	d	F	1	1	TRUE	0 12212	12.21

continued on next page

continued from previous page

Candidate	Group	Map	Question	Map Type	Answer	Given Answer	Correct?	Ease	Time (s)
28	B	4	d	P	1	1	TRUE	2 46443	46.44
29	A	4	d	F	1	1	TRUE	3 31803	31.8
30	C	4	d	G	1	4	FALSE	4 10375	10.38
31	A	4	d	F	1	1	TRUE	2 17984	17.98
32	C	4	d	G	1	1	TRUE	2 12132	12.13
33	B	4	d	P	1	1	TRUE	2 17672	17.67
34	B	4	d	P	1	0	FALSE	3 52162	52.16
35	C	4	d	G	1	1	TRUE	1 6594	6.59
36	C	4	d	G	1	1	TRUE	2 11875	11.88
37	B	4	d	P	1	2	FALSE	4 37663	37.66
38	A	4	d	F	1	1	TRUE	2 12425	12.43
39	A	4	d	F	1	4	FALSE	4 28437	28.44
40	B	4	d	P	1	2	FALSE	2 20478	20.48
41	B	4	d	P	1	2	FALSE	2 17313	17.31
42	C	4	d	G	1	1	TRUE	2 10768	10.77
43	C	4	d	G	1	1	TRUE	1 16147	16.15
1	C	5	a	G	7	6	FALSE	2 17953	17.95
2	C	5	a	G	7	6	FALSE	2 20469	20.47
3	C	5	a	G	7	7	TRUE	2 21619	21.62
4	A	5	a	F	7	6	FALSE	1 14110	14.11
5	B	5	a	P	7	6	FALSE	3 27141	27.14
6	B	5	a	P	7	6	FALSE	4 39750	39.75
7	B	5	a	P	7	6	FALSE	2 55240	55.24
8	A	5	a	F	7	6	FALSE	3 23454	23.45
9	A	5	a	F	7	6	FALSE	2 24718	24.72
10	A	5	a	F	7	6	FALSE	2 26501	26.5

continued on next page

continued from previous page

Candidate	Group	Map	Question	Map Type	Answer	Given Answer	Correct?	Ease	Time (s)
11	C	5	a	G	7	6	FALSE	4 45508	45.51
12	B	5	a	P	7	6	FALSE	2 70926	70.93
13	C	5	a	G	7	6	FALSE	3 25483	25.48
14	A	5	a	F	7	6	FALSE	1 11891	11.89
15	A	5	a	F	7	6	FALSE	1 11938	11.94
16	C	5	a	G	7	6	FALSE	4 54923	54.92
17	B	5	a	P	7	6	FALSE	3 27563	27.56
18	B	5	a	P	7	6	FALSE	3 96403	96.4
19	C	5	a	G	7	6	FALSE	2 49143	49.14
20	C	5	a	G	7	6	FALSE	2 16626	16.63
21	A	5	a	F	7	6	FALSE	1 11954	11.95
22	B	5	a	P	7	6	FALSE	2 27422	27.42
23	A	5	a	F	7	7	TRUE	2 27925	27.93
24	B	5	a	P	7	6	FALSE	2 36504	36.5
25	C	5	a	G	7	6	FALSE	3 85126	85.13
26	B	5	a	P	7	10	FALSE	3 103139	103.14
27	A	5	a	F	7	6	FALSE	0 16336	16.34
28	B	5	a	P	7	6	FALSE	2 25970	25.97
29	A	5	a	F	7	6	FALSE	2 9305	9.31
30	C	5	a	G	7	6	FALSE	3 22343	22.34
31	A	5	a	F	7	6	FALSE	2 24064	24.06
32	C	5	a	G	7	6	FALSE	2 9052	9.05
33	B	5	a	P	7	6	FALSE	4 68002	68
34	B	5	a	P	7	6	FALSE	2 33593	33.59
35	C	5	a	G	7	6	FALSE	4 47423	47.42
36	C	5	a	G	7	6	FALSE	4 71845	71.85

continued on next page

continued from previous page

Candidate	Group	Map	Question	Map Type	Answer	Given Answer	Correct?	Ease	Time (s)
37	B	5	a	P	7	6	FALSE	4 20626	20.63
38	A	5	a	F	7	6	FALSE	3 15166	15.17
39	A	5	a	F	7	6	FALSE	2 11422	11.42
40	B	5	a	P	7	6	FALSE	0 63356	63.36
41	B	5	a	P	7	6	FALSE	3 32078	32.08
42	C	5	a	G	7	6	FALSE	4 32102	32.1
43	C	5	a	G	7	6	FALSE	3 55079	55.08
1	C	5	b	P	14	12	FALSE	2 29750	29.75
2	C	5	b	P	14	11	FALSE	2 21390	21.39
3	C	5	b	P	14	11	FALSE	4 21723	21.72
4	A	5	b	G	14	13	FALSE	2 9766	9.77
5	B	5	b	F	14	14	TRUE	2 15985	15.99
6	B	5	b	F	14	14	TRUE	3 18204	18.2
7	B	5	b	F	14	14	TRUE	2 13673	13.67
8	A	5	b	G	14	13	FALSE	3 32626	32.63
9	A	5	b	G	14	13	FALSE	2 26234	26.23
10	A	5	b	G	14	13	FALSE	3 33517	33.52
11	C	5	b	P	14	14	TRUE	4 30735	30.74
12	B	5	b	F	14	13	FALSE	3 53221	53.22
13	C	5	b	P	14	13	FALSE	3 19123	19.12
14	A	5	b	G	14	13	FALSE	2 13251	13.25
15	A	5	b	G	14	13	FALSE	2 12563	12.56
16	C	5	b	P	14	13	FALSE	2 19500	19.5
17	B	5	b	F	14	14	TRUE	3 35750	35.75
18	B	5	b	F	14	13	FALSE	1 14442	14.44
19	C	5	b	P	14	12	FALSE	3 27059	27.06

continued on next page

continued from previous page

Candidate	Group	Map	Question	Map Type	Answer	Given Answer	Correct?	Ease	Time (s)
20	C	5	b	P	14	13	FALSE	3 20924	20.92
21	A	5	b	G	14	10	FALSE	2 16388	16.39
22	B	5	b	F	14	14	TRUE	3 15140	15.14
23	A	5	b	G	14	13	FALSE	2 35420	35.42
24	B	5	b	F	14	14	TRUE	2 19658	19.66
25	C	5	b	P	14	13	FALSE	1 14906	14.91
26	B	5	b	F	14	14	TRUE	2 30077	30.08
27	A	5	b	G	14	13	FALSE	1 12368	12.37
28	B	5	b	F	14	14	TRUE	2 14607	14.61
29	A	5	b	G	14	13	FALSE	3 13934	13.93
30	C	5	b	P	14	13	FALSE	3 20640	20.64
31	A	5	b	G	14	13	FALSE	2 13897	13.9
32	C	5	b	P	14	14	TRUE	2 15573	15.57
33	B	5	b	F	14	14	TRUE	2 13998	14
34	B	5	b	F	14	14	TRUE	0 29210	29.21
35	C	5	b	P	14	13	FALSE	1 14953	14.95
36	C	5	b	P	14	13	FALSE	2 17047	17.05
37	B	5	b	F	14	14	TRUE	3 12070	12.07
38	A	5	b	G	14	13	FALSE	3 25285	25.29
39	A	5	b	G	14	13	FALSE	2 12281	12.28
40	B	5	b	F	14	14	TRUE	0 12159	12.16
41	B	5	b	F	14	14	TRUE	3 15146	15.15
42	C	5	b	P	14	13	FALSE	4 16780	16.78
43	C	5	b	P	14	13	FALSE	1 13945	13.95
1	C	5	c	G	1	1	TRUE	1 13422	13.42
2	C	5	c	G	1	1	TRUE	2 26438	26.44

continued on next page

continued from previous page

Candidate	Group	Map	Question	Map Type	Answer	Given Answer	Correct?	Ease	Time (s)
3	C	5	c	G	1	1	TRUE	2 13679	13.68
4	A	5	c	F	1	1	TRUE	1 9359	9.36
5	B	5	c	P	1	1	TRUE	1 7828	7.83
6	B	5	c	P	1	1	TRUE	2 10625	10.63
7	B	5	c	P	1	1	TRUE	2 13845	13.85
8	A	5	c	F	1	1	TRUE	2 13156	13.16
9	A	5	c	F	1	1	TRUE	2 13406	13.41
10	A	5	c	F	1	1	TRUE	1 15110	15.11
11	C	5	c	G	1	1	TRUE	2 26814	26.81
12	B	5	c	P	1	2	FALSE	1 20767	20.77
13	C	5	c	G	1	1	TRUE	2 16515	16.52
14	A	5	c	F	1	1	TRUE	0 8375	8.38
15	A	5	c	F	1	1	TRUE	2 13782	13.78
16	C	5	c	G	1	1	TRUE	1 15001	15
17	B	5	c	P	1	1	TRUE	0 28985	28.99
18	B	5	c	P	1	2	FALSE	0 23311	23.31
19	C	5	c	G	1	1	TRUE	1 21452	21.45
20	C	5	c	G	1	1	TRUE	1 13267	13.27
21	A	5	c	F	1	1	TRUE	1 8515	8.52
22	B	5	c	P	1	1	TRUE	1 10281	10.28
23	A	5	c	F	1	1	TRUE	2 11010	11.01
24	B	5	c	P	1	1	TRUE	1 12923	12.92
25	C	5	c	G	1	1	TRUE	1 9797	9.8
26	B	5	c	P	1	1	TRUE	1 10125	10.13
27	A	5	c	F	1	1	TRUE	0 11375	11.38
28	B	5	c	P	1	1	TRUE	2 22767	22.77

continued on next page

continued from previous page

Candidate	Group	Map	Question	Map Type	Answer	Given Answer	Correct?	Ease	Time (s)
29	A	5	c	F	1	1	TRUE	2 4438	4.44
30	C	5	c	G	1	1	TRUE	2 11124	11.12
31	A	5	c	F	1	1	TRUE	2 8668	8.67
32	C	5	c	G	1	1	TRUE	1 9570	9.57
33	B	5	c	P	1	1	TRUE	1 13594	13.59
34	B	5	c	P	1	1	TRUE	0 21423	21.42
35	C	5	c	G	1	1	TRUE	2 7766	7.77
36	C	5	c	G	1	1	TRUE	2 11266	11.27
37	B	5	c	P	1	1	TRUE	2 12000	12
38	A	5	c	F	1	1	TRUE	2 10829	10.83
39	A	5	c	F	1	1	TRUE	2 9109	9.11
40	B	5	c	P	1	1	TRUE	0 9283	9.28
41	B	5	c	P	1	1	TRUE	1 8188	8.19
42	C	5	c	G	1	1	TRUE	1 20354	20.35
43	C	5	c	G	1	1	TRUE	1 10244	10.24
1	C	5	d	P	2	3	FALSE	2 10641	10.64
2	C	5	d	P	2	1	FALSE	2 11866	11.87
3	C	5	d	P	2	2	TRUE	3 12690	12.69
4	A	5	d	G	2	1	FALSE	2 7047	7.05
5	B	5	d	F	2	2	TRUE	2 11203	11.2
6	B	5	d	F	2	1	FALSE	2 21125	21.13
7	B	5	d	F	2	1	FALSE	1 15971	15.97
8	A	5	d	G	2	1	FALSE	2 19219	19.22
9	A	5	d	G	2	2	TRUE	2 20327	20.33
10	A	5	d	G	2	1	FALSE	4 40172	40.17
11	C	5	d	P	2	2	TRUE	3 21319	21.32

continued on next page

continued from previous page

Candidate	Group	Map	Question	Map Type	Answer	Given Answer	Correct?	Ease	Time (s)
12	B	5	d	F	2	1	FALSE	1 19610	19.61
13	C	5	d	P	2	2	TRUE	1 15499	15.5
14	A	5	d	G	2	2	TRUE	2 34283	34.28
15	A	5	d	G	2	2	TRUE	2 15220	15.22
16	C	5	d	P	2	1	FALSE	4 28641	28.64
17	B	5	d	F	2	2	TRUE	4 20048	20.05
18	B	5	d	F	2	1	FALSE	0 15955	15.96
19	C	5	d	P	2	2	TRUE	2 38863	38.86
20	C	5	d	P	2	1	FALSE	1 13329	13.33
21	A	5	d	G	2	1	FALSE	1 19060	19.06
22	B	5	d	F	2	1	FALSE	1 16499	16.5
23	A	5	d	G	2	1	FALSE	3 26687	26.69
24	B	5	d	F	2	2	TRUE	2 29504	29.5
25	C	5	d	P	2	2	TRUE	4 19438	19.44
26	B	5	d	F	2	1	FALSE	1 14672	14.67
27	A	5	d	G	2	2	TRUE	3 47098	47.1
28	B	5	d	F	2	1	FALSE	2 8823	8.82
29	A	5	d	G	2	1	FALSE	3 25261	25.26
30	C	5	d	P	2	1	FALSE	1 17343	17.34
31	A	5	d	G	2	2	TRUE	3 23084	23.08
32	C	5	d	P	2	2	TRUE	2 10366	10.37
33	B	5	d	F	2	2	TRUE	3 19434	19.43
34	B	5	d	F	2	1	FALSE	0 17049	17.05
35	C	5	d	P	2	2	TRUE	2 15594	15.59
36	C	5	d	P	2	2	TRUE	3 25875	25.88
37	B	5	d	F	2	2	TRUE	3 10415	10.42

continued on next page

continued from previous page

Candidate	Group	Map	Question	Map Type	Answer	Given Answer	Correct?	Ease	Time (s)
38	A	5	d	G	2	1	FALSE	1 9042	9.04
39	A	5	d	G	2	1	FALSE	3 22203	22.2
40	B	5	d	F	2	2	TRUE	2 25583	25.58
41	B	5	d	F	2	1	FALSE	2 10338	10.34
42	C	5	d	P	2	1	FALSE	1 10800	10.8
43	C	5	d	P	2	2	TRUE	2 23674	23.67
1	C	6	a	P	2	1	FALSE	1 12343	12.34
2	C	6	a	P	2	2	TRUE	2 55767	55.77
3	C	6	a	P	2	3	FALSE	0 28706	28.71
4	A	6	a	G	2	1	FALSE	2 10875	10.88
5	B	6	a	F	2	2	TRUE	1 15610	15.61
6	B	6	a	F	2	1	FALSE	1 16031	16.03
7	B	6	a	F	2	2	TRUE	0 20752	20.75
8	A	6	a	G	2	2	TRUE	3 14969	14.97
9	A	6	a	G	2	1	FALSE	1 22047	22.05
10	A	6	a	G	2	1	FALSE	1 12266	12.27
11	C	6	a	P	2	5	FALSE	4 61130	61.13
12	B	6	a	F	2	2	TRUE	1 24766	24.77
13	C	6	a	P	2	2	TRUE	3 12234	12.23
14	A	6	a	G	2	2	TRUE	1 8969	8.97
15	A	6	a	G	2	2	TRUE	2 13923	13.92
16	C	6	a	P	2	5	FALSE	4 72845	72.85
17	B	6	a	F	2	2	TRUE	0 16125	16.13
18	B	6	a	F	2	2	TRUE	0 21709	21.71
19	C	6	a	P	2	2	TRUE	2 48096	48.1
20	C	6	a	P	2	1	FALSE	1 11407	11.41

continued on next page

continued from previous page

Candidate	Group	Map	Question	Map Type	Answer	Given Answer	Correct?	Ease	Time (s)
21	A	6	a	G	2	1	FALSE	1 10797	10.8
22	B	6	a	F	2	2	TRUE	1 7125	7.13
23	A	6	a	G	2	2	TRUE	1 16639	16.64
24	B	6	a	F	2	2	TRUE	1 33332	33.33
25	C	6	a	P	2	5	FALSE	4 102064	102.06
26	B	6	a	F	2	2	TRUE	1 16968	16.97
27	A	6	a	G	2	1	FALSE	0 15163	15.16
28	B	6	a	F	2	1	FALSE	2 18724	18.72
29	A	6	a	G	2	5	FALSE	2 12942	12.94
30	C	6	a	P	2	5	FALSE	4 42420	42.42
31	A	6	a	G	2	1	FALSE	1 20558	20.56
32	C	6	a	P	2	5	FALSE	4 76186	76.19
33	B	6	a	F	2	2	TRUE	1 9110	9.11
34	B	6	a	F	2	2	TRUE	1 14520	14.52
35	C	6	a	P	2	2	TRUE	2 28032	28.03
36	C	6	a	P	2	3	FALSE	4 82627	82.63
37	B	6	a	F	2	2	TRUE	1 27299	27.3
38	A	6	a	G	2	1	FALSE	4 9939	9.94
39	A	6	a	G	2	1	FALSE	2 8203	8.2
40	B	6	a	F	2	2	TRUE	0 10177	10.18
41	B	6	a	F	2	2	TRUE	1 6313	6.31
42	C	6	a	P	2	2	TRUE	3 28643	28.64
43	C	6	a	P	2	1	FALSE	0 10900	10.9
1	C	6	b	F	4	4	TRUE	2 11922	11.92
2	C	6	b	F	4	4	TRUE	2 14942	14.94
3	C	6	b	F	4	4	TRUE	3 20475	20.48

continued on next page

continued from previous page

Candidate	Group	Map	Question	Map Type	Answer	Given Answer	Correct?	Ease	Time (s)
4	A	6	b	P	4	4	TRUE	2 9751	9.75
5	B	6	b	G	4	4	TRUE	3 20047	20.05
6	B	6	b	G	4	4	TRUE	3 15328	15.33
7	B	6	b	G	4	4	TRUE	3 26394	26.39
8	A	6	b	P	4	4	TRUE	2 19485	19.49
9	A	6	b	P	4	4	TRUE	2 27390	27.39
10	A	6	b	P	4	5	FALSE	4 28438	28.44
11	C	6	b	F	4	4	TRUE	1 23359	23.36
12	B	6	b	G	4	4	TRUE	3 54487	54.49
13	C	6	b	F	4	4	TRUE	1 13937	13.94
14	A	6	b	P	4	4	TRUE	3 12688	12.69
15	A	6	b	P	4	4	TRUE	2 14016	14.02
16	C	6	b	F	4	4	TRUE	0 9422	9.42
17	B	6	b	G	4	4	TRUE	3 25516	25.52
18	B	6	b	G	4	4	TRUE	2 31145	31.15
19	C	6	b	F	4	4	TRUE	1 24389	24.39
20	C	6	b	F	4	4	TRUE	2 12970	12.97
21	A	6	b	P	4	4	TRUE	3 22527	22.53
22	B	6	b	G	4	4	TRUE	3 8484	8.48
23	A	6	b	P	4	4	TRUE	2 28177	28.18
24	B	6	b	G	4	4	TRUE	1 21690	21.69
25	C	6	b	F	4	4	TRUE	1 6782	6.78
26	B	6	b	G	4	5	FALSE	4 21296	21.3
27	A	6	b	P	4	4	TRUE	2 27211	27.21
28	B	6	b	G	4	6	FALSE	2 20122	20.12
29	A	6	b	P	4	4	TRUE	3 14012	14.01

continued on next page

continued from previous page

Candidate	Group	Map	Question	Map Type	Answer	Given Answer	Correct?	Ease	Time (s)
30	C	6	b	F	4	4	TRUE	2 10343	10.34
31	A	6	b	P	4	4	TRUE	2 11884	11.88
32	C	6	b	F	4	4	TRUE	1 10486	10.49
33	B	6	b	G	4	4	TRUE	4 34823	34.82
34	B	6	b	G	4	5	FALSE	1 34388	34.39
35	C	6	b	F	4	4	TRUE	1 11704	11.7
36	C	6	b	F	4	4	TRUE	1 11875	11.88
37	B	6	b	G	4	4	TRUE	2 11508	11.51
38	A	6	b	P	4	4	TRUE	3 12900	12.9
39	A	6	b	P	4	5	FALSE	4 16469	16.47
40	B	6	b	G	4	4	TRUE	0 21115	21.12
41	B	6	b	G	4	4	TRUE	2 12211	12.21
42	C	6	b	F	4	4	TRUE	3 18045	18.05
43	C	6	b	F	4	4	TRUE	0 11713	11.71
1	C	6	c	P	0	0	TRUE	2 20484	20.48
2	C	6	c	P	0	0	TRUE	2 32125	32.13
3	C	6	c	P	0	0	TRUE	0 17247	17.25
4	A	6	c	G	0	0	TRUE	1 11234	11.23
5	B	6	c	F	0	0	TRUE	0 7016	7.02
6	B	6	c	F	0	2	FALSE	3 14125	14.13
7	B	6	c	F	0	0	TRUE	1 13596	13.6
8	A	6	c	G	0	0	TRUE	2 22406	22.41
9	A	6	c	G	0	0	TRUE	3 36358	36.36
10	A	6	c	G	0	0	TRUE	1 12969	12.97
11	C	6	c	P	0	0	TRUE	2 29912	29.91
12	B	6	c	F	0	0	TRUE	2 36002	36

continued on next page

continued from previous page

Candidate	Group	Map	Question	Map Type	Answer	Given Answer	Correct?	Ease	Time (s)
13	C	6	c	P	0	1	FALSE	2 22655	22.66
14	A	6	c	G	0	0	TRUE	1 12094	12.09
15	A	6	c	G	0	0	TRUE	1 14579	14.58
16	C	6	c	P	0	0	TRUE	0 15235	15.24
17	B	6	c	F	0	0	TRUE	1 31281	31.28
18	B	6	c	F	0	2	FALSE	1 32632	32.63
19	C	6	c	P	0	0	TRUE	1 21483	21.48
20	C	6	c	P	0	0	TRUE	2 19548	19.55
21	A	6	c	G	0	0	TRUE	1 12328	12.33
22	B	6	c	F	0	0	TRUE	1 12734	12.73
23	A	6	c	G	0	2	FALSE	2 16922	16.92
24	B	6	c	F	0	4	FALSE	2 28909	28.91
25	C	6	c	P	0	0	TRUE	1 8047	8.05
26	B	6	c	F	0	0	TRUE	1 16656	16.66
27	A	6	c	G	0	0	TRUE	0 11594	11.59
28	B	6	c	F	0	0	TRUE	2 9481	9.48
29	A	6	c	G	0	0	TRUE	1 18361	18.36
30	C	6	c	P	0	0	TRUE	2 11890	11.89
31	A	6	c	G	0	0	TRUE	1 13354	13.35
32	C	6	c	P	0	0	TRUE	2 12966	12.97
33	B	6	c	F	0	0	TRUE	1 10266	10.27
34	B	6	c	F	0	0	TRUE	0 16631	16.63
35	C	6	c	P	0	0	TRUE	1 13407	13.41
36	C	6	c	P	0	0	TRUE	1 14641	14.64
37	B	6	c	F	0	0	TRUE	1 13734	13.73
38	A	6	c	G	0	0	TRUE	4 14324	14.32

continued on next page

continued from previous page

Candidate	Group	Map	Question	Map Type	Answer	Given Answer	Correct?	Ease	Time (s)
39	A	6	c	G	0	0	TRUE	2 11781	11.78
40	B	6	c	F	0	0	TRUE	4 7767	7.77
41	B	6	c	F	0	0	TRUE	1 12750	12.75
42	C	6	c	P	0	0	TRUE	1 13580	13.58
43	C	6	c	P	0	0	TRUE	1 22097	22.1
1	C	6	d	F	1	2	FALSE	2 25094	25.09
2	C	6	d	F	1	2	FALSE	2 28089	28.09
3	C	6	d	F	1	1	TRUE	2 24042	24.04
4	A	6	d	P	1	1	TRUE	0 15719	15.72
5	B	6	d	G	1	1	TRUE	2 11938	11.94
6	B	6	d	G	1	2	FALSE	2 23719	23.72
7	B	6	d	G	1	1	TRUE	2 14033	14.03
8	A	6	d	P	1	1	TRUE	4 27485	27.49
9	A	6	d	P	1	1	TRUE	4 79264	79.26
10	A	6	d	P	1	0	FALSE	1 17422	17.42
11	C	6	d	F	1	1	TRUE	2 31541	31.54
12	B	6	d	G	1	1	TRUE	2 46268	46.27
13	C	6	d	F	1	1	TRUE	3 26452	26.45
14	A	6	d	P	1	1	TRUE	2 31127	31.13
15	A	6	d	P	1	1	TRUE	2 23361	23.36
16	C	6	d	F	1	2	FALSE	2 22219	22.22
17	B	6	d	G	1	1	TRUE	4 41626	41.63
18	B	6	d	G	1	2	FALSE	2 20524	20.52
19	C	6	d	F	1	1	TRUE	2 32347	32.35
20	C	6	d	F	1	3	FALSE	2 20439	20.44
21	A	6	d	P	1	1	TRUE	2 34134	34.13

continued on next page

continued from previous page

Candidate	Group	Map	Question	Map Type	Answer	Given Answer	Correct?	Ease	Time (s)
22	B	6	d	G	1	1	TRUE	1 17734	17.73
23	A	6	d	P	1	1	TRUE	2 31694	31.69
24	B	6	d	G	1	1	TRUE	1 20518	20.52
25	C	6	d	F	1	1	TRUE	1 13141	13.14
26	B	6	d	G	1	1	TRUE	1 14281	14.28
27	A	6	d	P	1	1	TRUE	2 20957	20.96
28	B	6	d	G	1	1	TRUE	2 22847	22.85
29	A	6	d	P	1	1	TRUE	4 57179	57.18
30	C	6	d	F	1	1	TRUE	2 15937	15.94
31	A	6	d	P	1	1	TRUE	3 48227	48.23
32	C	6	d	F	1	1	TRUE	3 13070	13.07
33	B	6	d	G	1	1	TRUE	2 14513	14.51
34	B	6	d	G	1	1	TRUE	2 35457	35.46
35	C	6	d	F	1	1	TRUE	1 23594	23.59
36	C	6	d	F	1	1	TRUE	2 15078	15.08
37	B	6	d	G	1	1	TRUE	3 11227	11.23
38	A	6	d	P	1	2	FALSE	4 18663	18.66
39	A	6	d	P	1	1	TRUE	4 24047	24.05
40	B	6	d	G	1	1	TRUE	1 19167	19.17
41	B	6	d	G	1	1	TRUE	2 11446	11.45
42	C	6	d	F	1	2	FALSE	2 21083	21.08
43	C	6	d	F	1	1	TRUE	2 22690	22.69

B.3.2 Design Preferences

Table B.8: Design Preferences: Geographic Map. “1” represents the design that the candidate felt was most preferable and “3” represents the least preferable map.

Candidate	Atlanta	Bucharest	Mexico	Stockholm	Toronto	Washington
1	3	2	2	3	3	3
2	3	2	3	3	3	3
3	2	1	2	3	2	2
4	3	3	2	3	3	3
5	3	2	3	3	3	3
6	2	1	3	3	3	3
7	3	2	2	3	3	3
8	3	2	2	3	3	3
9	2	2	2	3	2	3
10	3	2	2	2	2	3
11	3	2	2	3	2	3
12	3	3	2	3	3	3
13	1	3	3	1	2	3
14	3	2	2	3	2	3
15	1	3	2	3	1	3
16	2	3	2	3	2	3
17	3	2	2	3	3	3
18	3	3	3	3	3	3
19	2	1	1	2	3	3
20	2	3	2	3	1	3
21	1	2	1	2	1	3
22	1	1	2	3	2	3
23	2	1	2	3	1	3
24	1	2	2	2	3	3
25	1	1	2	3	1	3
26	3	2	2	3	3	3
27	1	1	2	3	2	3
28	2	1	2	3	2	2
29	3	3	2	3	3	3
30	3	3	2	3	3	3
31	3	2	2	2	2	3
32	2	2	2	3	2	3
33	2	1	2	2	2	2
34	3	3	2	3	3	3
35	3	2	3	2	2	3

continued on next page

continued from previous page

Candidate	Atlanta	Bucharest	Mexico	Stockholm	Toronto	Washington
36	3	2	2	2	2	3
37	1	2	2	1	2	3
38	3	2	2	3	2	3
39	3	2	2	3	2	3
40	3	3	2	3	2	3
41	1	2	2	2	2	3
42	1	3	2	3	3	3
43	3	2	2	3	3	3

Table B.9: Design Preferences: Normalised Published Map. "1" represents the design that the candidate felt was most preferable and "3" represents the least preferable map.

Candidate	Atlanta	Bucharest	Mexico	Stockholm	Toronto	Washington
1	2	3	3	1	1	1
2	1	3	2	1	1	2
3	3	3	3	1	1	1
4	2	2	3	1	1	2
5	2	3	2	1	1	1
6	3	2	2	1	1	2
7	2	3	3	1	2	1
8	1	3	3	2	2	2
9	3	3	3	2	3	2
10	1	3	3	1	3	2
11	2	3	3	1	3	2
12	2	2	3	1	1	2
13	3	2	2	3	3	2
14	2	3	3	1	3	2
15	2	1	3	1	3	1
16	1	2	3	1	3	2
17	2	3	3	2	1	2
18	2	2	2	1	1	1
19	1	3	2	1	1	2
20	1	1	3	1	3	2
21	3	1	3	1	3	2
22	2	2	3	1	3	2
23	3	2	3	1	3	2
24	3	3	3	1	2	2
25	3	3	3	1	3	1
26	2	3	3	1	2	2

continued on next page

continued from previous page

Candidate	Atlanta	Bucharest	Mexico	Stockholm	Toronto	Washington
27	2	2	3	1	3	2
28	1	3	3	2	1	3
29	2	2	3	1	2	2
30	1	2	3	1	1	1
31	2	3	3	1	3	2
32	1	1	3	1	3	2
33	1	2	3	1	3	3
34	1	2	3	1	1	1
35	1	3	2	1	3	2
36	2	3	3	1	3	2
37	3	3	3	2	3	2
38	2	3	3	1	3	2
39	2	3	3	2	3	2
40	2	2	3	1	3	2
41	3	1	3	1	3	2
42	2	2	3	1	2	1
43	2	3	3	1	1	2

Table B.10: Design Preferences: Automatically-Drawn Map. “1” represents the design that the candidate felt was most preferable and “3” represents the least preferable map.

Candidate	Atlanta	Bucharest	Mexico	Stockholm	Toronto	Washington
1	1	1	1	2	2	2
2	2	1	1	2	2	1
3	1	2	1	2	3	3
4	1	1	1	2	2	1
5	1	1	1	2	2	2
6	1	3	1	2	2	1
7	1	1	1	2	1	2
8	2	1	1	1	1	1
9	1	1	1	1	1	1
10	2	1	1	3	1	1
11	1	1	1	2	1	1
12	1	1	1	2	2	1
13	2	1	1	2	1	1
14	1	1	1	2	1	1
15	3	2	1	2	2	2
16	3	1	1	2	1	1
17	1	1	1	1	2	1

continued on next page

continued from previous page

Candidate	Atlanta	Bucharest	Mexico	Stockholm	Toronto	Washington
18	1	1	1	2	2	2
19	3	2	3	3	2	1
20	3	2	1	2	2	1
21	2	3	2	3	2	1
22	3	3	1	2	1	1
23	1	3	1	2	2	1
24	2	1	1	3	1	1
25	2	2	1	2	2	2
26	1	1	1	2	1	1
27	3	3	1	2	1	1
28	3	2	1	1	3	1
29	1	1	1	2	1	1
30	2	1	1	2	2	2
31	1	1	1	3	1	1
32	3	3	1	2	1	1
33	3	3	1	3	1	1
34	2	1	1	2	2	2
35	2	1	1	3	1	1
36	1	1	1	3	1	1
37	2	1	1	3	1	1
38	1	1	1	2	1	1
39	1	1	1	1	1	1
40	1	1	1	2	1	1
41	2	3	1	3	1	1
42	3	1	1	2	1	2
43	1	1	1	2	2	1

B.3.3 Questionnaire Feedback

Table B.11: Have you seen any of the metro maps shown here before these tests? If so, which ones?

Candidate	Answer
1	Yes, I have seen some metro maps before the test. I have seen and used the New York Metro
2	None, but some layouts seemed familiar, like the london underground
3	Yes Slide 6
4	No
5	No
6	No

continued on next page

continued from previous page

Candidate	Answer
7	N/A
8	No
9	None
10	Yes, the one of the Mexico City subway system
11	Map C from slide 1, most familiar metro maps are straight lined, definite in shape
12	No
13	No
14	No
15	No
16	No
17	None
18	No, I dont use metro
19	No
20	No
21	No
22	No
23	No, never
24	No
25	No
26	London Underground
27	Think so, London ones including DLR?
28	No
29	No
30	No -Have seen the London Underground ones but do not remember them in the test
31	No
32	None
33	Briefly seen book of Metro maps, but not used/studied
34	None
35	Yes, alas I cannot remember which ones!
36	Can't remember!
37	No
38	No. Very familiar with London Underground which is similar
39	None
40	I dont know
41	No
42	No
43	Washington DC

Table B.12: Which features of the metro map layout did you find most helpful when completing the tests?

Candidate	Answer
1	The maps indicated the relevant stations which we needed to find
2	One that was not too scattered, but had a defined straight and clear structure as it was easier to read
3	s+8 line maps
4	Straight lines, lines spread out for ease on eye
5	Clear naming and placement
6	Straight lines and colours
7	Simple non overlaying colours. Basic shapes with straight edges. Basic symettry/even distribution of station/rail. Well spaced out and easy to read train station names
8	Interchanges clearly marked
9	Geometrical symetry (straight lines); spacing between labels; highlighting end points
10	Uniformity, straight lines, labels not on top of each other
11	Definate and spaced out layouts with clear indication of colour and station points such as [inverted T stubby diagram]
12	Interchange Signs
13	Different colours, changes clearly pointed out
14	Having enough space between stations to make them distinct - some maps had overlapping text and lines which confused things.
15	Colour
16	Layout and colour
17	Circular nodes on interconnecting stations, sensible layout, sensible angles on lines, straight lines, sensible colours, lines not randomly joining
18	Straight lines and the fact that some metro maps had clear stations, not confusing
19	The ones with biggest text and biggest ticks
20	Indentations representing stations
21	Different Colours
22	Colour coded tracks, clear circles on join stations
23	Different colours for different lines is helpful; When the map was large and different stations were well spaced out and not placed close together, it was easier to answer the questions
24	The blue or red coloured lines (made it easier to see the stations)

continued on next page

continued from previous page

Candidate	Answer
25	Clear images, the on which you could see the station stops
26	The small lines indicating a station was their. Also the bigger maps which didnt have station names overlapping were much better
27	. for stations, colours of different lines; circle for crossover stations
28	Well spread with appropriate use of colour
29	Large gaps between stations (so names dont overlap)
30	Clear linear route layout, uncluttered
31	Metro maps with straight lines; metro maps with easily distingishable colours; maps where overlapping lines had half one colour, half the other, such as the blue/orange line to suggest the line occupied the same tracks
32	Straight lines
33	Names of stations next to the markers on the lines
34	The colours indicating the lines
35	Circles for track interchange, some had a more 'sensible' layout
36	distance between station lables; straight routes not curly; colour coding of routes
37	clearly marked stations that were spaced out
38	Stations spread out; Stations especially where lines cross not too close together; Maps with fewest line crossings/colours
39	When the lines could only be from lines at other angles made it harder. When station names were not near each other. If stations near one above one below [DIAGRAMS]
40	Maps normalized to straight lines (where possible); where the station and names were spaced out were most helpful but if there was a choice between a crowded straight line map and a cursive spaced out map, I favoured the spaced out map
41	Enough spacing between station names; Clearly marked stations. Spacing between tracks
42	Colour coding of line, compact text close to stations
43	Different symbols where lines meet

Table B.13: Which features of the metro map layout did you find least helpful when completing the tests?

Candidate	Answer
1	The maps were hard to read and would be better enlarged
2	When there was more than three change connections and the routes did not have a clear straight grid map which made it difficult to read

continued on next page

continued from previous page

Candidate	Answer
3	ones with all the curves
4	All bundelled together, not clear, station name writing overlaps
5	Messy naming and the misplaced change station
6	When names were too close together
7	Curved lines representing routes. Squashed text and squashed track/lines into small areas. Overlayed colours for multiple tracks.
8	Yellow 'tick' marks against white background; Station names overalapping onto other station names; lines overlapping each other when going in the same direction
9	Vertical length
10	Labels that cant be read cuz they're on top of each other; lines changing colour without a hollow circle where they meet
11	When routes were the same colour and followed curved or jagged lines, and when route changes were not clearly indicated with a circle
12	untidy lines (not straight)
13	Unexplained changes, from one colour to the next
14	NULL
15	Size of map (when stations were too close to each other to distinguish between them)
16	Positioning of station names and station indicators
17	Poor space management, differently shaped station nodes, changes with no icon indication, poor choice of colours
18	The confusing lines
19	Couldnt read some of them as text label obscured ticks or obscured other labels
20	stations with long names usually overwrite something
21	When the font was too small and the names were too close together
22	Cramp spacing between station writing, unclear ticks on tracks
23	When the stations were placed too close together it was hard to distinguish between them, especially when they had long names which took up a lot of space on the map
24	The yellow coloured lines (made it harder to see the stations)
25	Station names which ran into each other and overlapped
26	Overlapping station names. Very squashed maps were bad.

continued on next page

continued from previous page

Candidate	Answer
27	Close together station names, change of line colour with no circle to indicate line crossover
28	The almost handdrawn metro maps without space between the stations. Ever though this may not be precise but provides better navigation and understanding
29	When line colours overlapped or "jumped" stations
30	Station names overlapping
31	Metro maps where lines fully overlapped one lines. Maps where names overlapped the lines. Maps with bright colours, such as yellow
32	Yellow lines were hard to see on white background
33	Overlapping text with lines or other text
34	None
35	Colours were often hard to see clearly
36	overlapping lines; non-distinctive "change"-station symbols
37	Cramped maps where it was difficult to see stations or connections
38	Station names not spread out or stations too close together; Some names not easy to read
39	When you could not see join for a line. Station names too close together
40	Being crowded, swapping line colurs made it difficult and made me think changes were needed. Also, where interchange stations lacked a white circle made more difficult
41	Places where two tracks follow the same route shown as an alternating line
42	Overlapping text/stations;yellow-an-white station marks hard to see; text close to the stations
43	Station names overlapping

Table B.14: Did you find any of the questions ambiguous?

Candidate	Answer
1	No, the questions were clearly laid out
2	Do you count a change if the route . Has a same colour or do you always assume change lead to only one to one mapping
3	Yes
4	There were a few, but I dont remember which ones
5	No
6	No
7	No.

continued on next page

continued from previous page

Candidate	Answer
8	Not really
9	No
10	yes
11	None
12	No
13	Whether you have seen some metro maps. Might have seen some but not remembering them.
14	Several of the 'how many do you pass though' questions had multiple routes
15	No
16	Yes, some questions seemed impossible to answer given the answers provided. Because the route from A to B is not possible by train, e.g. routes dont connect
17	Some questions were potentially ambiguous, due to the variety of routes in some circumstances
18	No
19	No
20	no
21	Yes, most of them
22	Yes, some tracks did not join
23	No
24	No
25	Yes, coloured lines which run into other colours without line change marks - why were these diff colours? Also lines which were the same colour with line change marks. Why change of line but no change of colour?
26	No
27	Yes
28	Yes
29	Some did not have the correct list of questions
30	It was unclear if you had to change when two lines intersect but the route continues
31	No
32	There were often many routes with different numbers of stations to go by
33	Stora Massen to Karlaplan; Dupont to Chester; Pentagon to Court House
34	Yes, some of the questions were not clear
35	Some, as stations were not always clear

continued on next page

continued from previous page

Candidate	Answer
36	Yes, Q1, as we were not told to remember map layouts beforehand
37	When it asked for a number of changes on lines with multiple endings, what counts as a change?
38	No
39	No
40	Many, especially the ones that were incorrect. The abiguous ones were the ones with inconsistent station and line labeling
41	No
42	No -Although some of the metro maps had yellow-an-white lines/stations - hard to read
43	No