AUTOMATIC LAYOUT OF METRO MAPS USING MULTICRITERIA OPTIMISATION

A THESIS SUBMITTED TO

THE UNIVERSITY OF KENT

IN THE SUBJECT OF COMPUTER SCIENCE

FOR THE DEGREE

OF DOCTOR OF PHILOSOPHY.

By

Jonathan M. Stott

October 2008

Abstract

This thesis is concerned with the automatic layout of metro maps using a hill climbing multicriteria optimiser. Metro maps are a widely used tool for the visualisation of the interconnecting services on a public transport network, allowing the user of the network to quickly and easily plan their route between stations. The difficulty of manually drawing these maps means that their use has been restricted and it is difficult to apply the metro map metaphor to other application areas beyond public transport networks.

This thesis introduces a method for drawing metro maps based on a graph model using a hill climbing multicriteria optimisation method. This method combines a number of criteria for positioning nodes (stations) taking into account aesthetic properties such as the angular resolution of edges, the octilinearity of edges, the length of edges and the straightness of lines. Another set of criteria handle the labelling of station names by considering the number of occlusions that the labels make and the positioning of the labels relative to other labels. A number of clustering and partitioning methods are also used in response to particular local minima in the search space.

The metro map layout method is demonstrated for a number of real-world examples, which are then evaluated empirically. This evaluation compares metro maps drawn with our method against metro maps currently in use as well as geographic maps of the metro network. This tests the hypotheses that maps drawn using this method are preferred by users and that they can be better for route-finding tasks in terms of the accuracy and time taken to complete those tasks.

This thesis concludes with a discussion describing how the metro map layout method can be applied to other application areas such as the visualisation of project plans and website navigation maps. Also described are details of ongoing and future research.

Dedication

This thesis is dedicated to the memory of my grandad and grandpa, whose inspiration and encouragement when I was young will always be remembered.

Contents

List of Tables

List of Figures

Acknowledgements

My biggest thanks go to my close family, especially my mum and my dad, for their unending support and encouragement during the long (and often painful) years of completing this thesis. Special thanks go to my sister, Laura, who found time between serving ice creams at Drayton Manor Park to help proof-read my work. I must thank my supervisor at the University of Kent, Dr. Peter Rodgers, for his time over the last four years and for giving me the opportunity to explore this research area in the first place. The input from Prof. Steven Walker and Juan-Carlos Martinez-Ovando at the Institute of Mathematics, Statistics and Actuarial Science at the University of Kent as I tried to get my head around the statistical analysis required for our empirical evaluation was invaluable. I have Dr. Fred Barnes to blame for being persuasive enough back in 2001 to even make me consider doing a Ph.D.!

A number of people have expressed interest in this work and have provided input. These include Remo Burkhard at the Universität St. Gallen in Switzerland, Doug Rose at FWT, Andrew Dow and Maxwell Roberts.

Finally, I would like to thank all the friends with whom I have shared the four years that I spent working on my Ph.D.. They made my time at the University of Kent something that I will remember for many years to come. They are too numerous to mention, but I am sure they know who they are!

Chapter 1

Introduction

Metro maps can be seen worldwide and have possibly become one of the most memorised cartographic items in the world [Dow05, Gar94, Ove03]. Ever since construction of the first railways began in the early 19th century, there has been a need to map the networks. This was particularly the case with metro networks, where people needed 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 increasingly difficult to read (see Section 2.2.1).

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 poster to advertise the ferry service between Harwich in England and the Hook of Holland was produced by a Dutchman by the name of Henrik Willem Mesdag [Wil47]. These maps were usually seen in advertising and promotional posters where the aim of using a simplified map of the network might have been to emphasise 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 published in 1933 [Gar94] (Figure 1.2), marked a significant departure from the more traditional geographic maps and built on the ideas introduced by earlier simple diagrammatic maps. Prior to this, geographic maps of the London Underground were used such as the example designed by Fred Stingemore and published in 1926 (Figure 1.1). Beck introduced more formalised design rules which have been followed right through to this day [Rob05], as can be seen in the 2005 map designed by Clockwork (Figure 1.3).

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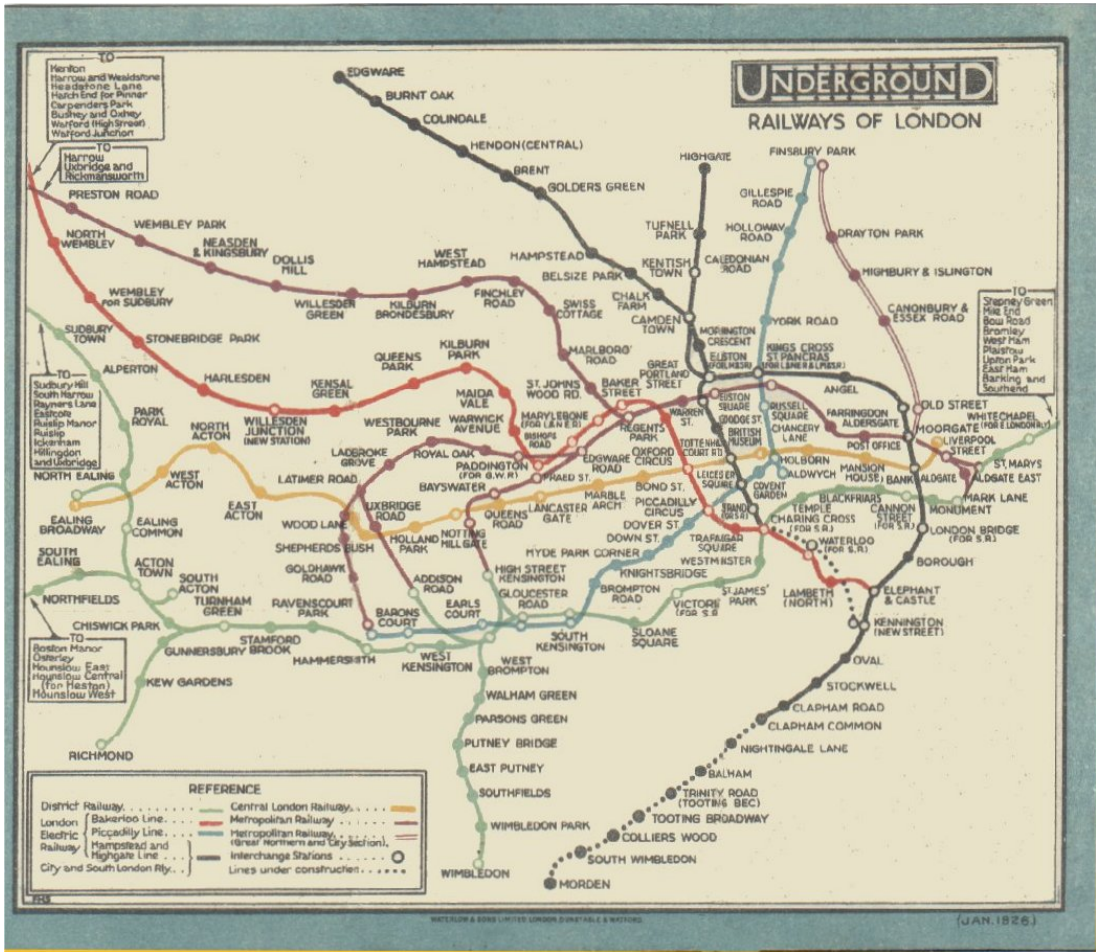
Figure 1.1: London Underground map published in 1926, designed by Fred Stingemore.

Figure 1.2: London Underground map published in 1933, designed by Harry Beck.
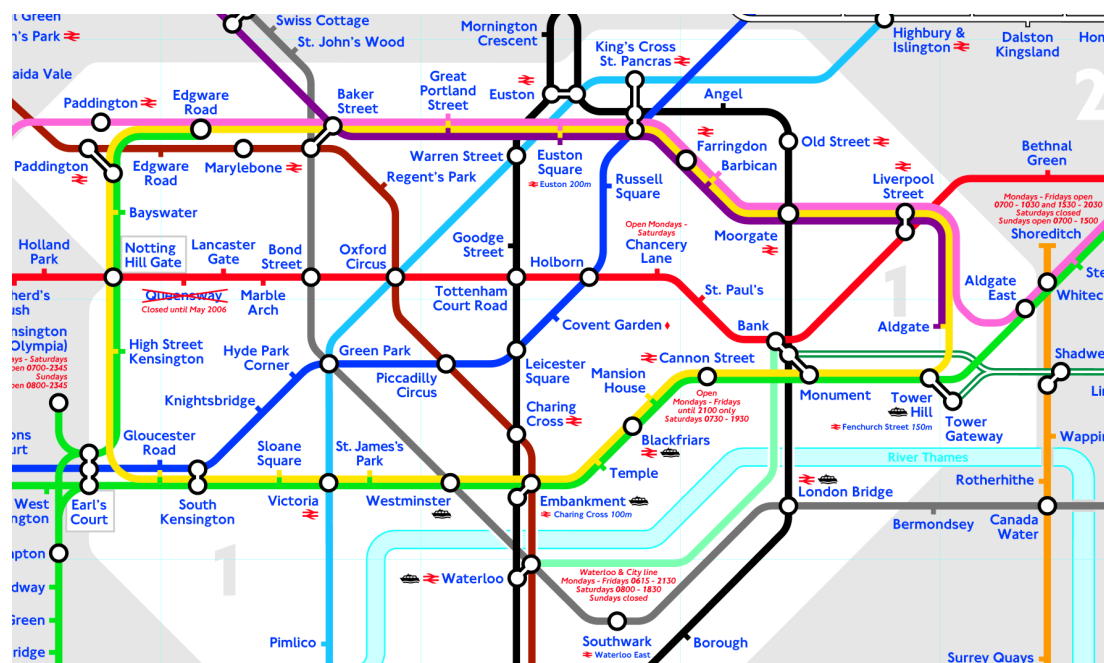
Figure 1.3: Extract from the London Underground map published in 2005 by Clockwork. Reproduced with permission from Transport for London.

has even been embraced as an iconic image of London. Following on from the success of Beck's version of the London Underground map, most metro networks now have a schematic map using a similar concept. Schematic maps have also been adapted for use with other public transport networks such as overground 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 their use.

At first glance, it might appear that metro maps are simple to design. However, this is seldom the case, and it often requires an experienced cartographer with considerable skill and design knowledge to be able to produce effective maps. There are many aspects which together contribute to produce an effective design: 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 of the medium being used to display the maps. Even what may appear to be trivial changes—for example, slightly increasing the size of labels—can drastically alter the overall appearance and effectiveness 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

The motivation for our research comes from the apparent difficulty in drawing metro maps. We are keen to explore effective methods for automatically drawing metro maps or sections of metro maps which can help map designers create alternative (and possibly improved) designs of existing maps and to provide targeted maps for travellers giving enhanced route information.

We believe that the metro map as a conceptual metaphor has value as a design form beyond its current use for navigation of public transport networks and can be applied to other application areas such as for the visualisation of project plans or website navigation. However, if these application areas are to be realised, some way for metro maps to be more easily produced is necessary. This thesis ultimately describes how we have implemented a novel method for the layout of metro maps, empirically evaluated the layout and then described how it can be applied to other application areas.

1.3 Contributions of this Research

The main contributions of this research are:

- A novel method for the layout of metro maps which uses a hill climbing multicriteria optimiser that combines the following elements:

  - the movement of individual nodes to more optimal locations based on a set of six node movement criteria which take into account such features as the length of edges, the angular resolution of edges around a node, edge crossings, the straightness of lines and the regularity of angles of edges;

  - the identification of clusters by finding overlength edges and partitioning the graph and the movement of these clusters using the same set of criteria as for individual nodes;

  - the repositioning of labels in more optimal locations based on a set of seven labelling criteria which incorporate label occlusions, label position, the consistency of label position, the proximity of labels to other nodes and the distance of a label from the node it belongs to.

- An empirical evaluation that provides statistical evidence to support our hypotheses that automatically-drawn metro maps are better for route planning tasks than geographical maps.

- The adaptation with examples of our metro map layout method to novel application areas where metro maps have not previously been used, particularly that of the visualisation of project plans and website navigation.

1.4 Thesis Outline

The structure of this thesis is as follows:

Chapter 1 (this chapter) introduces the metro map as a tool for visualisation and outlines our motivation and contributions of this 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. Also given is a comprehensive analysis of the characteristic features of metro maps which draws upon the historical evolution of the metro map as a design form. This chapter also defines the prevalent characteristics of the *metro map metaphor* that can also be used for abstract visualisation.

Chapter 3 describes the process by which we draw metro maps using hill climbing multicriteria optimisation. It details the process involved in selecting optimum positions for nodes including the various criteria and rules that we have implemented. The chapter also describes the criteria that we developed in order to position station labels.

Chapter 4 addresses the problem of moving many nodes in a graph at the same time by presenting methods in which clusters of nodes can be identified; we also show how a graph can be partitioned.

Chapter 5 describes how the method described in Chapters 3 and 4 was implemented. We explain how we determined effective criteria weightings as well as provide an analysis of particular local minima.

Chapter 6 provides a set of results generated using our metro map layout method. The chapter also contains a discussion of the method used with reference to a worked example.

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

Chapter 8 describes the modifications to our method that would be required for two application areas which apply the metro map metaphor to diagrams other than of public transport networks.

Chapter 9 summarises our conclusions and presents directions for further research as well as limitations of our research.

1.5 Publications

Four peer-reviewed publications have resulted from this research:

- Jonathan M. Stott and Peter Rodgers. Metro map layout using multicriteria optimization. In *Proceedings of the 8th International Conference on Information Visualisation (IV04)*, pp. 355-362. IEEE, July 2004 [SR04].

- Remo A. Burkhard, Michael Meier, Peter Rodgers, Matthias T. J. Smis, and Jonathan 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 (I-KNOW 2005)*, pp. 388-395. Know-Center, Austria, June 2005 [BMR+05].

- Jonathan M. Stott, Peter Rodgers, Remo A. Burkhard, Michael Meier, and Matthias T. J. 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 [SRB+05].

- Jonathan M. Stott and Peter Rodgers. Automatic metro map design techniques. In *Proceedings of the 22nd International Cartographic Conference*, p. 10. International Cartographic Association, July 2005 [SR05].

Chapter 2

Background

This chapter introduces the various background material which is relevant to this area of research. It begins with a list of definitions used throughout this thesis (Section 2.1), followed by a discussion of the characteristics of metro maps in Section 2.2. Our definition of the metro map metaphor is presented in Section 2.3.

The next sections describe a number of different methods for graph drawing (Section 2.4) and laying out schematic diagrams and cartographic generalisation (Section 2.5). Map labelling (Section 2.6) and the applications of the metro map metaphor to the visualisation of abstract concepts (Section 2.7) are also covered. Section 2.8 presents existing research in the area of metro map layout. The relevance of the background material to this thesis is highlighted throughout.

2.1 Definitions

The following definitions of terminology are used throughout this thesis.

Cartography The drawing of maps.

Diagram An illustration of the appearance of an object, retaining shape and relations of various parts, without being an exact representation of the object. A diagram can also be used to provide a concrete visualisation of an abstract concept. Examples of diagrams are illustrated in Figure 2.1 and include Gantt charts, Euler diagrams, flowcharts and electronic circuit diagrams.

Embedding An embedding is a particular instance of some structure, X, in another structure, Y, such that there exists a mapping $f : X \rightarrow Y$ that preserves the structure of X. In the context of two-dimensional graph drawing, an embedding is a mapping of nodes and edges to particular points and lines in a plane.

Generalisation A cartographic process involving the selection, simplification and/or combination of information when drawing a map.

(a) Gantt chart

(b) Euler diagram

(c) Flow chart

(d) Circuit diagram

Figure 2.1: Examples of different types of diagrams: a Gantt chart (a), an Euler diagram (b), flow chart (c) and a circuit diagram (d).
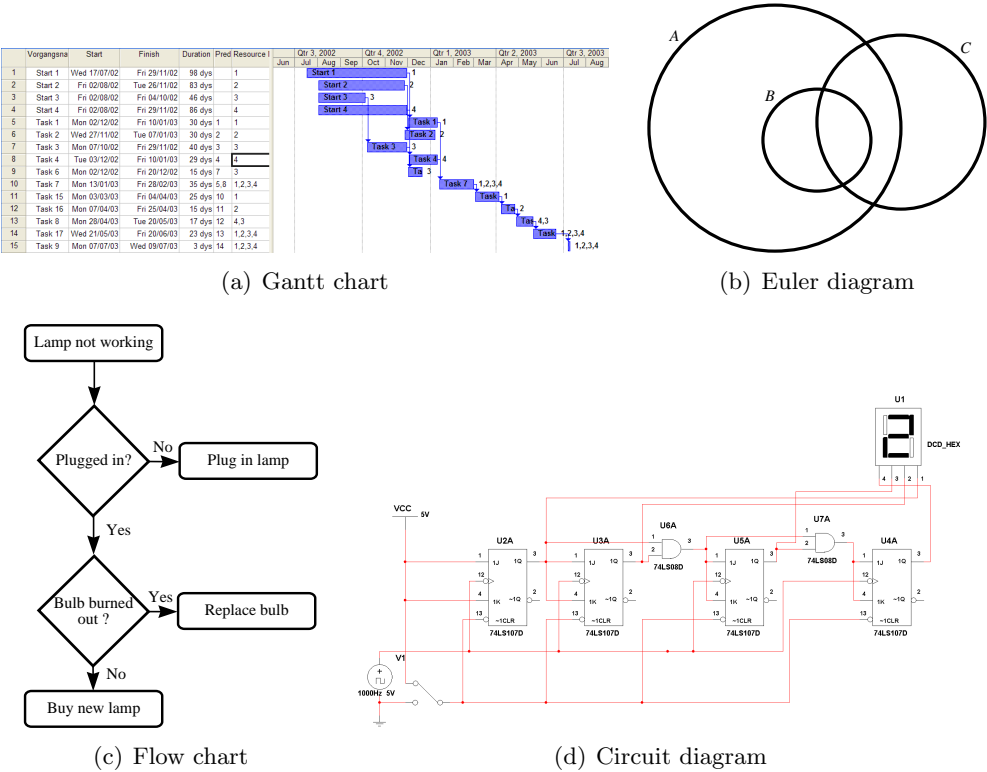
Graph A graph is a structure that expresses the connectivity of a set of objects. More formally, a graph, G, is a pair $G = (V, E)$ where V is a set of elements representing objects (*nodes*) and E is a set of unordered pairs representing the connectivity of nodes (*edges*). The *endpoints* of an edge, e, are the nodes belonging to that edge. The *degree*, $\rho(v)$, of a node v is the number of edges that have v as an endpoint. A *planar graph* is one which can be embedded in a plane such that no two edges intersect. Further definitions relating to graphs are presented in Section 2.4.1.

Graph drawing The process of finding embeddings of graphs.

Map A representation (model) of a geographic feature such that each part of the feature is drawn corresponding to some fixed scale or projection.

Metro map A schematic that represents a transport network. As a minimum, a metro map always consists of the following elements: *stations* which show places where passengers can board or alight from services, *metro lines* which show the routes taken by services, and *labels* which show the name of stations.

Metro map metaphor A conceptual metaphor that allows the understanding of one idea in terms of a metro map. More specifically, the application of the design characteristics of metro maps to concepts other than the illustration of transport networks. The metro map metaphor is defined in more detail in Section 2.3.

Schematic A diagram that represents the appearance of an object using abstract, graphic symbols.

Topography The surface features of a place or region.

Topology The study of the nature of space. In terms of the embedding of graphs or drawing of diagrams or schematics, topology is the consideration of the structure of the embedding or drawing and the way that constituent features (e.g. nodes and edges) are interrelated or arranged.

An important distinction must be made between the use of "map", "diagram" and "schematic". This is often confusing: for example, a "metro map" is a schematic and not a map in the geographic sense. "Diagram" and "schematic" are also often used interchangeably or even as the noun "schematic diagram". To avoid confusion, where possible, we have used the appropriate phrase when taking into account the above definitions and have highlighted places where the phrasiology in common use deviates from these defintions.

2.2   Characteristics of Metro Maps

Metro map diagrams have been around for almost as long as the networks themselves. Section 1.1 introduced some of the earliest examples of the metro map as opposed to previous geographic maps.

The requirements of metro maps evolved during the early years of public transport networks 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 before 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" [Dow05]. Commercial pressure may also have driven the change of 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 (particularly from outlying regions of urban networks).

One of the important features of metro map design is that the topology of the map should remain invariant, but the designer is afforded freedom to modify the location of stations and path of lines [Mor96b].

The next section describes how the schematic style for metro maps evolved.

2.2.1   Evolution of the Schematic Design for Metro Maps

It is important to understand the evolution of the schematic style for metro maps in order to appreciate the features which characterise the schematics over equivalent maps [AH06, Dow05, Gar94, Ove03]. 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. 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 2.2) [Hen29]. Unfortunately, the burgeoning networks soon resulted in a kind of map spaghetti and it became impossible to provide useful maps using the etching method. 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 excluded almost all surface topography. Examples of these minimalist maps include one of London's Metropolitan Railway in 1874 (Figure 2.3) and another of Berlin's railways (Figure 2.4).

However, these maps did not last very long in cities such as London, Berlin and New York as the networks continued to expand and even these "simpler" maps soon

Figure 2.2: Extract of part of an 1829 map of the Liverpool to Manchester Railway, drawn by G. Hennet [Hen29].
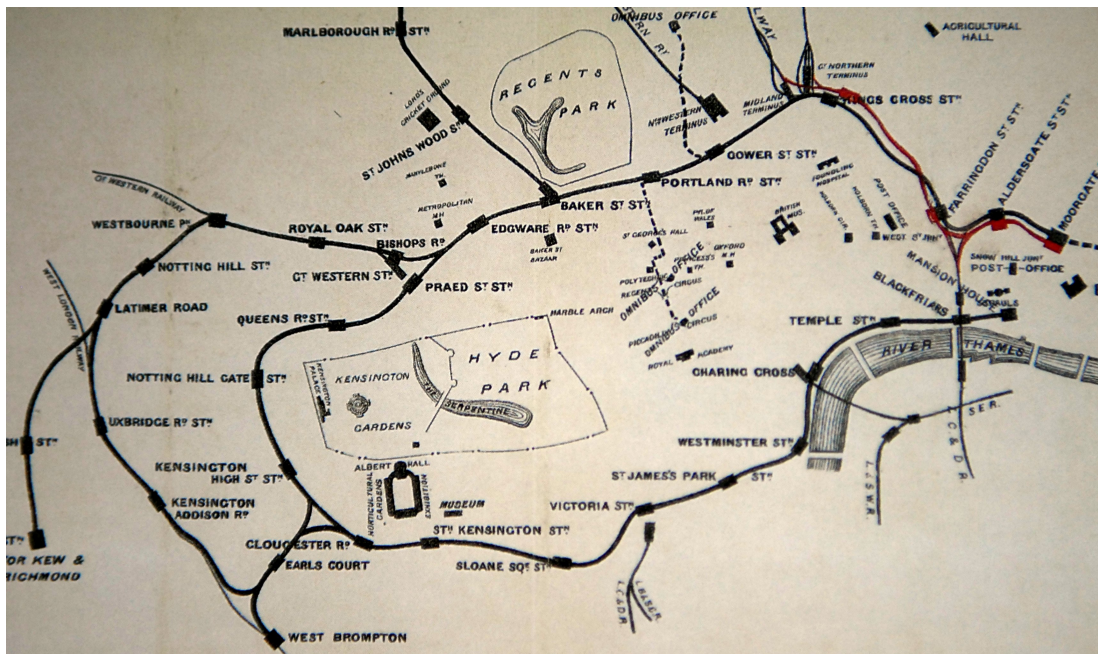
Figure 2.3: Map of London's Metropolitan Railway, published in 1874. This map illustrates an early example of the removal of much of the surface topography to produce a clearer map.
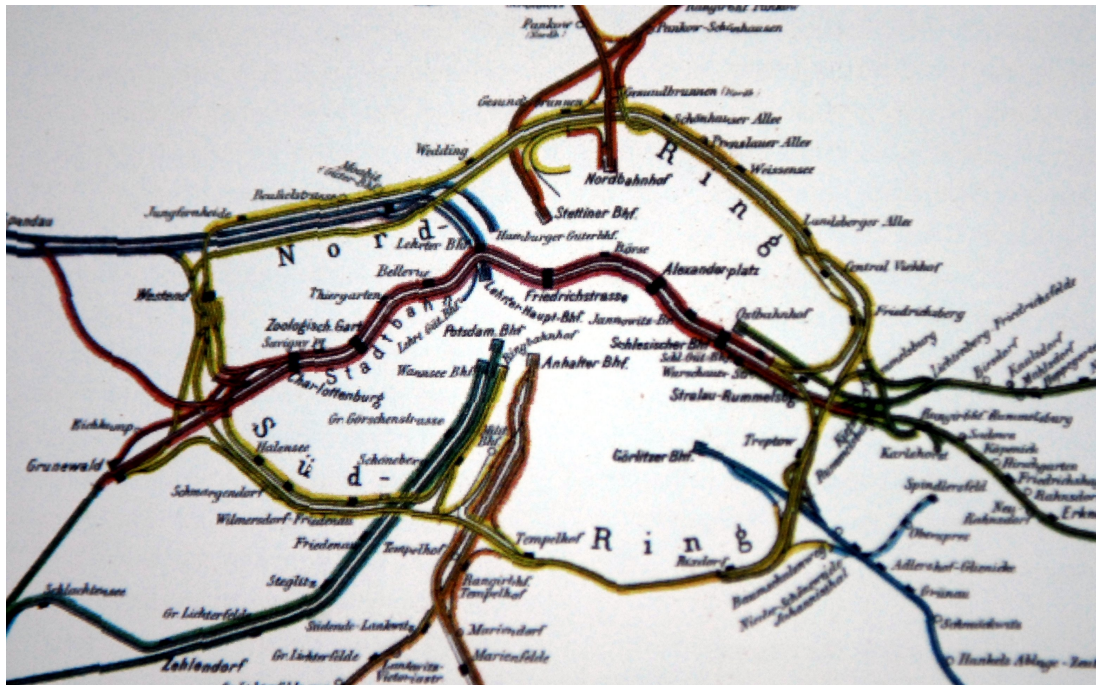
Figure 2.4: Late nineteenth-century map of Berlin's railways. This map has all surface topography removed to enhance the clarity of the railway.

became cluttered. The need arose to make more space in the maps, which could only be achieved by distorting the scale. This spatial distortion 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 schematic using distorted scale in this manner is that of the Metropolitan Railway, published in 1896 (Figure 2.5) 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 as the "convex effect". Harry Beck used this device in a large diagram of a complete network in his 1933 London Underground map (Figure 1.2): George Dow predates this in his 1929 LNER diagrams (Figure 2.6).

The next step in the evolution of the schematic form of the metro map came with the introduction of the practice of simplifying lines by removing most 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 schematics 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 schematics of 1929 (Figure 2.6) and positively seized upon by Beck for his seminal 1933 London Underground schematic (Figure 1.2). The
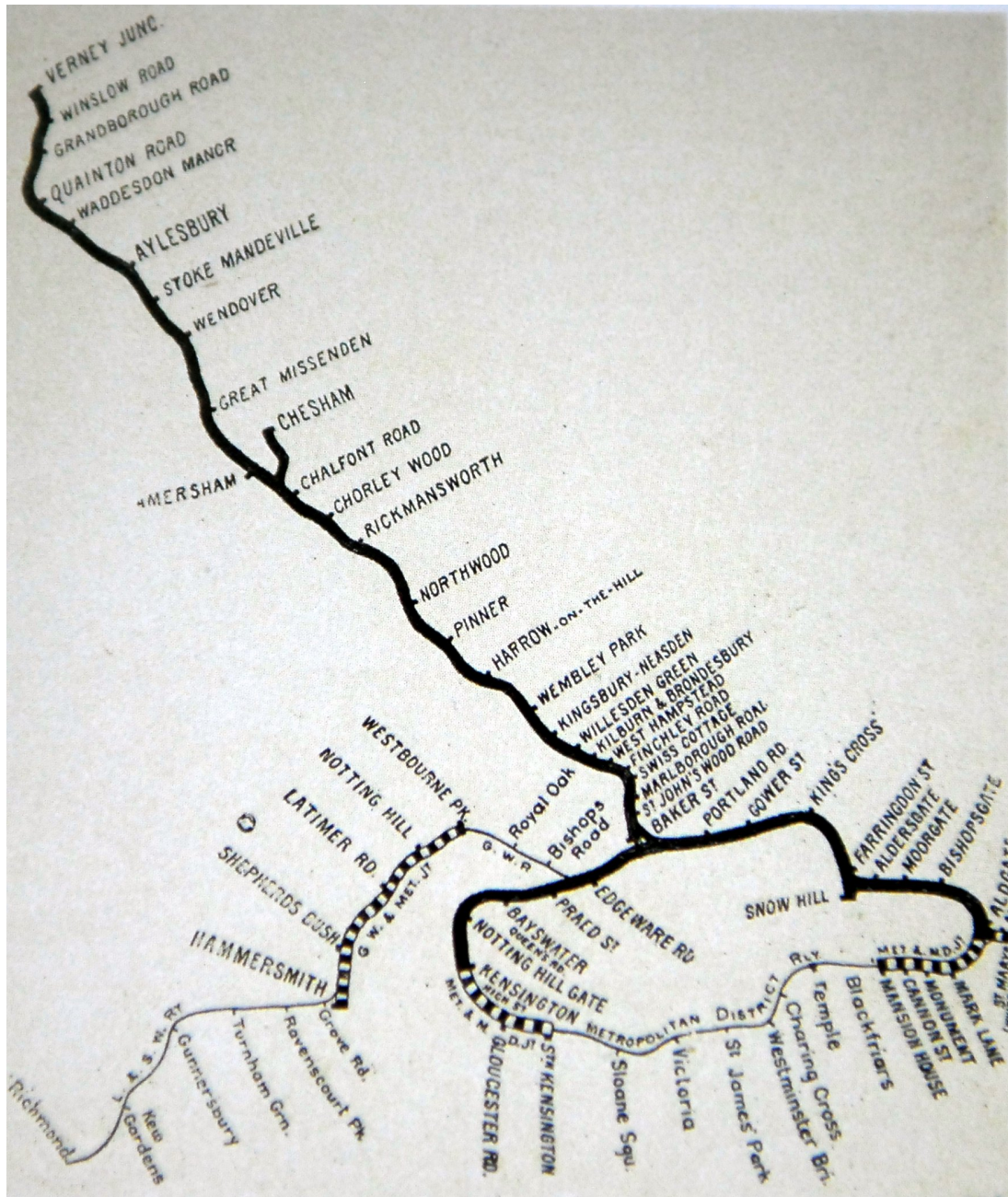
Figure 2.5: Schematic of the London Metropolitan Railway network, published in 1896. This map has large distortions in the scale—stations in the top-left corner of the map are up to six miles apart while some stations in the bottom-right corner are just a few hundred yards apart.
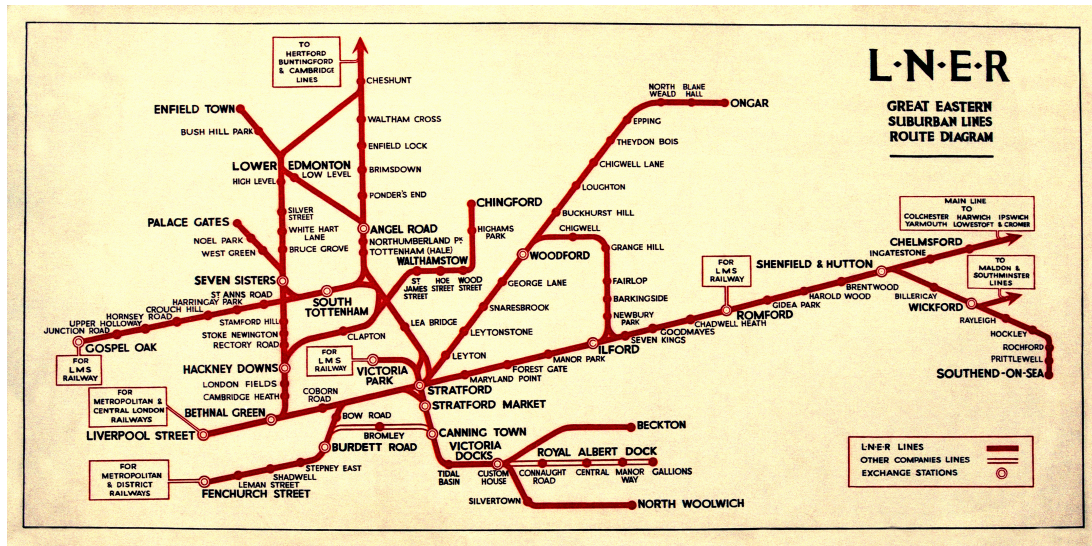
Figure 2.6: Schematic of the London and North Eastern Railway Great Eastern suburban network, published in 1929 and drawn by George Dow. Courtesy of Andrew Dow.

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 schematic, many others appeared following 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 design tweaks. It is surely testament to the forms introduced by both Dow and Beck that we still use schematics with the same features more than 70 years later.

More recently, Transport for London have produced a series of diagrams centred around individual bus stops [TfL07]. These *spider* or *octopus* diagrams feature a schematic showing all the bus routes that pass by a particular bus stop [Mor96a]. An example of a spider diagram showing bus routes from Monument in Central London is shown in Figure 2.7. The Transport for London spider diagrams include an area in the centre showing a street map which allows the location of individual bus stops to be shown in relation to the road network. As there will be many hundreds or thousands of bus stops, manual generation of these diagrams would be time consuming, so some automation of the process is necessary.
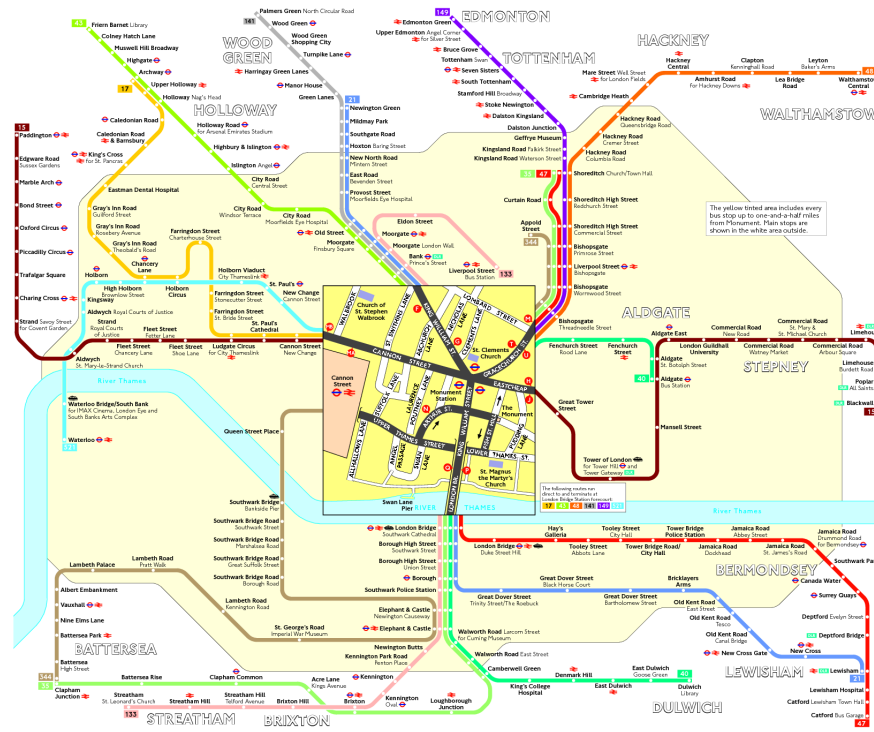
Figure 2.7: Spider diagram showing bus routes from Monument in Central London [TfL07].

2.2.2 Metro Lines

Depiction of metro 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. Metro lines are also usually distinguished on the map by using different colours. The colours of metro 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 and to emphasise the continuation of a line through a junction or station.

Examples of the features of metro lines can be seen in six map excerpts in Figure 2.8. 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° diagonals instead of 45° diagonals. Example (f) is unusual in that it does not 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.

2.2.3   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. 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 schematic metro 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 require. Examples (a), (c) and (d) in Figure 2.9 are good examples of this feature.

Station labels tend to be predominantly 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 is not 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 along 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 2.8(a). An example of diagonal labelling can be seen in Figure 2.9(f).
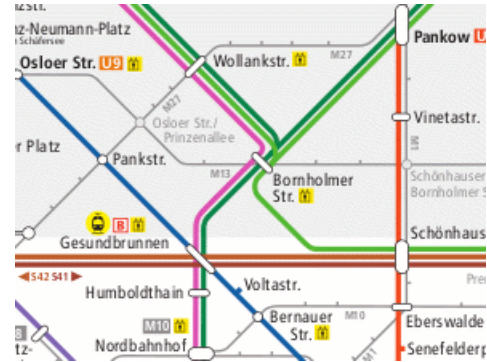
The biggest challenge when placing labels is to ensure that there is enough space for them without any cases where a label is drawn on top of (occludes) a line or a station. It is also desirable that the label is positioned so that it is unambiguously associated to just a single station. The example in Figure 2.10(f) shows a map where labels have been allowed to be drawn on top of lines and some of the labels could even be claimed to be ambiguous.

---

[1]Interestingly, proper case lettering on maps appeared quite late on, perhaps as late as 1936 in an LNER map [Dow05].

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

(a) London [TfL05]

(b) Berlin [BVG07]

(c) Munich [MVG07]

(d) Washington D.C. [WMA08]

(e) Madrid [MdM07]

(f) New York [MTA07]

Figure 2.8: Examples of metro 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.

(a) London [TfL05]

(b) Paris [RAT07]

(c) Oslo [OTB07]

(d) Stuttgart [VVS07]

(e) Tokyo [TM007]

(f) New York [MTA07]

Figure 2.9: 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.

2.2.4   Topographic Metadata

Topographic metadata are surface 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 either side of a river. 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 of fairly small networks. The general trend is for less metadata to be present as the size and complexity of the transport network increases.

Examples of maps with no metadata at all are illustrated in Figure 2.10, and include Madrid (a), Mexico City (b) and Stockholm (c). In these cases, metadata could have been included: Madrid could have featured metadata illustrating main roads (which the metro network mirrors closely in places) or large parks (such as the Parque del Retiro); Mexico City could also have used metadata showing main roads; Stockholm is built on several islands and the metro map could have illustrated the shoreline of the islands.

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 or coastline on the map (such as examples (d), (e) and (f) in Figure 2.10). 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.

2.2.5   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 different symbols. In the examples from London (Figure 2.11(a)) and Barcelona (Figure 2.11(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 2.12(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 2.12(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 2.13(a)) and slices taken through the line (Stockholm, Figure 2.13(b)).

(a) Madrid [MdM07]

(b) Mexico City [Mex08]

(c) Stockholm [SL008]

(d) Barcelona [TMB07]

(e) Boston [MBT07]

(f) Washington D.C. [WMA08]
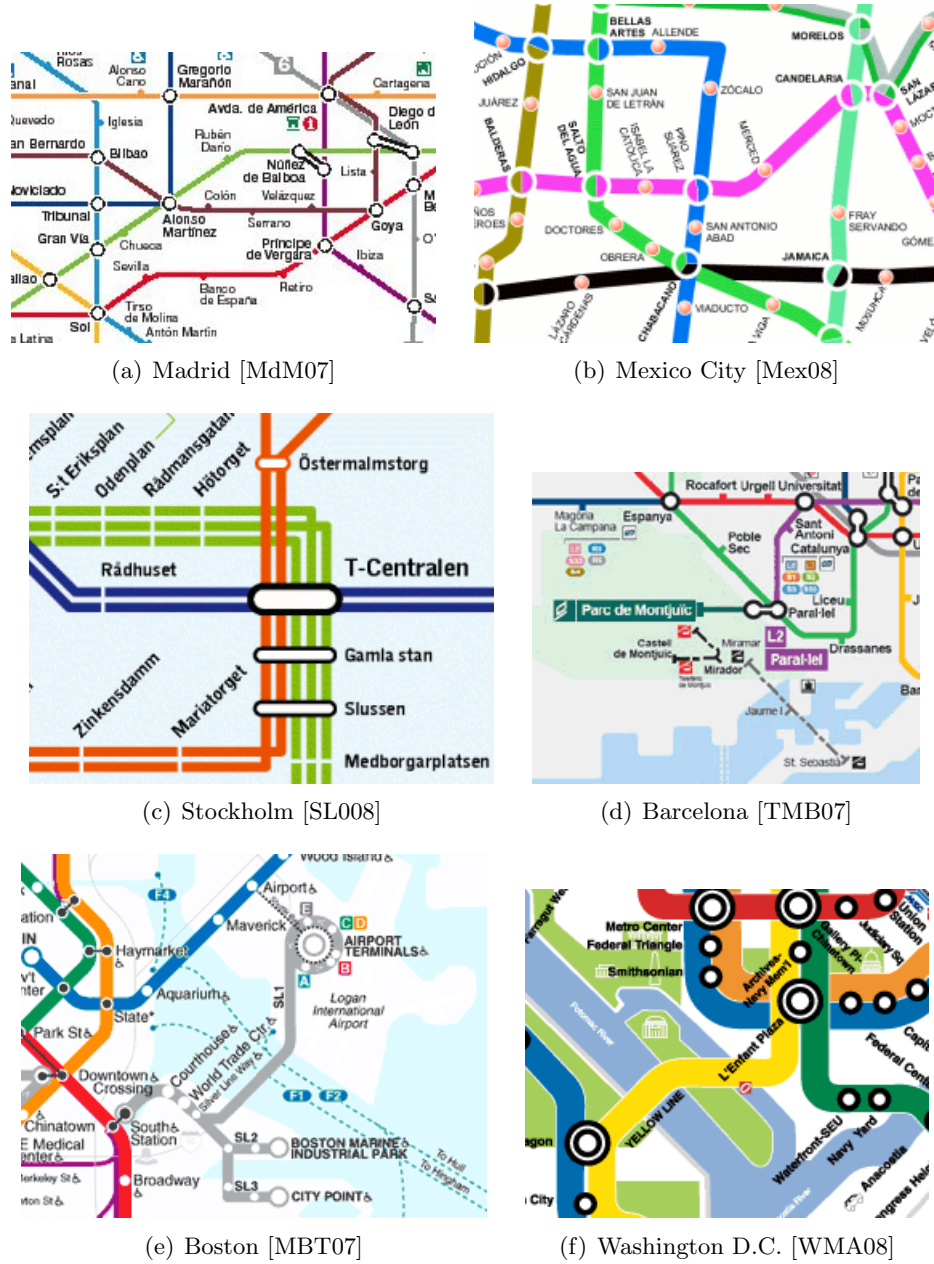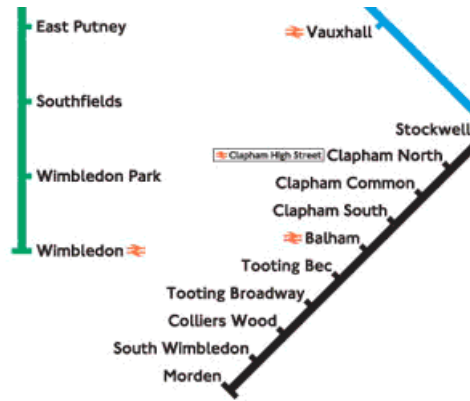
Figure 2.10: Examples of metro map topographic metadata on published maps. Examples (a), (b) and (c) have no topographic metadata at all. Examples (d), (e) and (f) show examples of topographic metadata including coastline, parks, rivers and landmarks.
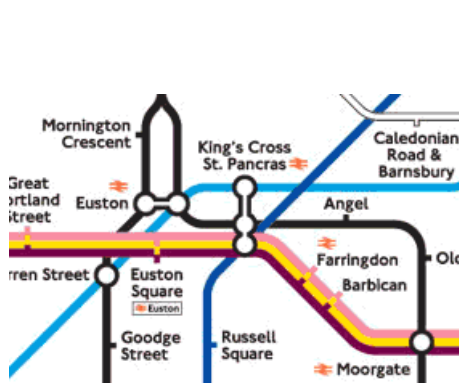
(a) London [TfL05]   (b) Barcelona [TMB07]

Figure 2.11: Examples of metro maps using ticks to represent stations on published maps.
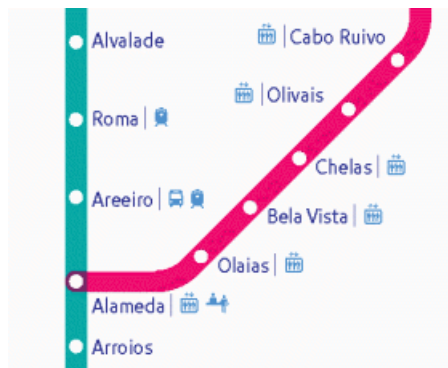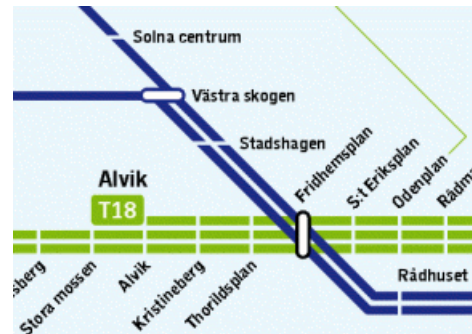
(a) London [TfL05]   (b) Hamburg [Hoc07]

Figure 2.12: Examples of devices used for distinguishing interchange stations on published metro maps.

(a) Lisbon [ML007]   (b) Stockholm [SL008]

Figure 2.13: Examples of metro maps using different devices for representing stations on published maps.

2.3 The Metro Map Metaphor

The previous section has highlighted a significant variety of characteristics for drawing metro maps. However, it is necessary to identify those characteristics that form the stereotypical view of how a metro map should appear. This will allow us to define more clearly what we mean when we talk of the *metro map metaphor*. While there will obviously be variation in how particular metro map designs appear, the following list of minimum requirements brings together the most commonly observed characteristics:

- **Metro lines** should be drawn using line segments that are either horizontal, vertical or at 45° diagonal. A metro line should be drawn using a single colour that allows the distinction of that line from others.

- **Stations** should be indicated on metro lines using a common symbol. Stations should be spaced evenly along metro lines—this implicitly requires that the scale of a metro map be undefined.

- **Labelling** of station names should be carried out using horizontal text. The possible locations for labels should be limited: labels on horizontal metro lines should be positioned immediately above or below the station symbol; labels on vertical lines should be immediately to the left or right of the station symbol; labels on diagonal lines should be above and to the left or right or below and to the left or right of the station label depending on the orientation of the line.

There is no requirement that a metro map include any **topographic metadata**. This is because the decision as to whether to include topographic metadata on a particular map depends on the context of that map meaning that there is no single rule that could fit all metro maps.

The selection of these characteristics comes about from our observations of real-world metro maps spanning the last eighty years. They are therefore based on historical convention and stereotypical perception rather than any formally validated rules determined empirically or otherwise.

This definition of the metro map metaphor drives the selection of the criteria for our metro map layout method described in Chapters 3 and 4 as well as the application of the metaphor to other application areas as described in Chapter 8.

2.4 Graphs and Graph Drawing

This section introduces graphs and the process of drawing graphs. As we will explain, graphs are a obvious representation for metro maps. A number of different methods for drawing graphs are considered, particularly where they appear to address some element of the metro map metaphor.
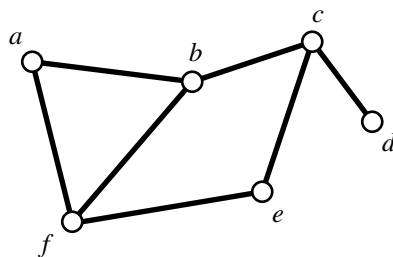
Figure 2.14: [An embedding of] a graph. The graph, $G = (V, E)$, contains a set of nodes, $V = \{a, b, c, d, e, f\}$, and a set of edges, $E = \{\{a, b\}, \{b, c\}, \{c, d\}, \{c, e\}, \{e, f\}, \{b, f\}, \{a, f\}\}$.

2.4.1   Graphs

A graph is a structure that expresses the connectivity of a set of objects. More formally, a graph, G, is a pair $G = (V, E)$ where V is a set of elements representing objects (*nodes* or *vertices*) and E is a set of unordered pairs representing the connectivity of nodes (*edges*). Figure 2.14 shows [an embedding of] a graph consisting of six nodes and seven edges. Graphs can be applied to a wide variety of real-world situations such as computer networks (the interconnections between computer systems), electronic circuits, social networks (illustrating friend or status relationships amongst people or animals) or transport networks (such as metro systems).

Edges can be *weighted* so that each edge carries some value or *weight*. For example, a graph showing distances between cities connected by roads could have weightings on each edge for the distance between two cities. An edge can also be *directed* to indicate a flow in just one direction between two nodes. A directed edge is analogous to a one-way street.

The particular properties of graphs are important to the understanding of what the graph is modelling. These properties can include adjacency and distance or the consideration of groupings of nodes and edges to form subgraphs, cycles or walks. In some situations it is important to know whether a graph can be drawn without any crossing edges: such graphs are planar graphs. The particular properties that are relevant to this thesis are described below.

Adjacency is the property concerning whether nodes or edges are directly connected. Two nodes, u and v are considered *adjacent* if they directly connected by at least one edge. An edge, x is said to be *incident* to a node, u, if u forms one of the endpoints of x. For example, in Figure 2.14, nodes a and f are adjacent whereas nodes a and e are not. The nodes b and c are both incident to the edge $\{b, c\}$. The *degree*, $\rho(u)$, of a node, u, is the number of incident edges to u. In Figure 2.14, the degree of node d is one and the degree of node f is three. The set of nodes that are adjacent to a particular node is called the set of *neighbours*. The neighbours of node b in Figure 2.14 are $\{a, c, f\}$.

The *distance* between two nodes, u and v, is the number of edges forming the shortest path between u and v. If u and v are the same node then the distance is zero. If there is no path between u and v (i.e. the graph is *disjoint*) then the distance is ∞. In Figure 2.14, the distance between nodes a and d is three and the distance between a and e is two.

A *subgraph* of a graph, G, is a graph formed from subsets of the sets of nodes and edges in G. Using Figure 2.14 as an example, the set of nodes $V^* = \{a, b, f\}$ and the set of edges $E^* = \{\{a, b\}, \{b, f\}, \{a, f\}\}$ forms a subgraph, $G^* = (V^*, E^*)$, of G. G is a *supergraph* of G^*.

A *walk* is a sequence of edges, starting from some node, u, and finishing on some other node, v, such that each subsequent edge shares a node. If u and v are the same node then the sequence forms a *closed walk*; otherwise it is an *open walk*. There is no requirement for every edge in a walk to be distinct: the case where the edges are distinct is called *trail*. A *path* is a trail that also contains only distinct nodes; a *cycle* is a closed path.

A *planar* graph is one which can be drawn without any edge crossings: a *plane* graph is one such embedding. Conversely, a *non-planar* graph is one which can not be drawn without edge crossings. Testing a graph for planarity can be done in $O(n)$ time where n is the number of nodes in the graph. Hopcroft and Tarjan present such a planarity-testing algorithm which uses a breadth-first search [HT74].

A *simple* or *strict* graph is one that contains only unweighted, undirected edges with no multiple edges between nodes or loops (where the two endpoints of an edge are the same node). Unless otherwise stated, a graph is normally considered to be simple.

2.4.2   Graph Drawing

Graph drawing is the process of finding a particular embedding of a graph [dBETT94a, dBETT94b, dBETT99, KW01]. In the context of two-dimensional graph drawing, an embedding is a mapping of nodes and edges to particular points and lines in a plane. This visualisation process is deeply rooted in the human interpretation of the graph as a structure and as such, the quality of the aesthetics of a particular drawing of a graph are very important. The readability of a graph is the key to allowing users to interpret the diagram accurately and clearly.

However, graph drawing is a generally difficult problem. Many aspects are NP-complete, such as determining the minimum number of edge crossings that a graph can be drawn with[3] [GJ83], finding a grid embedding of a graph that minimises the maximum edge length [MO85] or finding an embedding where all edges are the same

---

[3]The crossing number of a graph is related to the planarity of a graph: a planar graph has a crossing number of zero. Garey and Johnson show that determining the minimum crossing number for a graph is NP-complete [GJ83]—this is not the same as planarity testing.

length [Joh82]. By its very nature, the assessment of the quality of a particular drawing of a graph is entirely subjective. Therefore, acceptable heuristics are generally required to find good drawings of graphs in reasonable time, and measures of the aesthetic quality (either explicit or implicit) are required in order to be more objective about a particular drawing. Section 2.4.3 discusses what aesthetic criteria are used to drive graph drawing and how the importance of those criteria have been empirically assessed.

2.4.3 Graph Drawing Aesthetics

Many graph drawing algorithms make common assumptions that certain aesthetic criteria of the graph will 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]. A limited amount of empirical research has been conducted in order to quantify and justify these aesthetic criteria. This research is presented here.

In order to be able to evaluate any graph for its aesthetic quality, a number of criteria must first be identified and then defined quantitatively. Purchase [Pur02] provides a set of seven criteria and describes how they can be calculated such that they produce a normalised value between zero and one. The criteria defined are described below and illustrated in Figure 2.15.

1. **Minimising edge crossings.** This criterion addresses the intuitive situation where edge crossings can affect the readability of a graph (particularly if the angle between the edges is small) [RT81] (Figure 2.15(a)).

2. **Minimising edge bends.** In polyline edges (edges formed from more than one straight-line segment), the number of bends (joins between adjacent line segments) should be minimised [Tam87]. This criterion is shown in Figure 2.15(b).

3. **Maximising symmetry.** If the graph is innately symmetrical, the graph should be drawn to show this symmetry [GN98] (Figure 2.15(c)).

4. **Maximising the minimum angle between edges leaving a node.** Maximising the angle between adjacent edges leaving a node will reduce the risk that two edges are drawn close together. If the angle is very small, then it may be almost impossible to distinguish the edges (this is called angular resolution) [CSP96, GM98] (Figure 2.15(d)).

5. **Maximising edge orthogonality.** This criterion measures the deviation from an orthogonal angle (horizontal or vertical) for each edge in the graph [Tam87, PT97]. A completely orthogonal embedding would perform well under this criterion whereas an embedding with very few orthogonal edges would perform poorly. This is shown in Figure 2.15(e).

6. **Maximising node orthogonality.** Node orthogonality prefers nodes to be positioned such that they are on intersections of an integer square grid [Tam87, PT97] (Figure 2.15(f)).

7. **Maximising consistent flow direction for directed graphs.** In a directed graph, the "flow" of edges should be as consistent as possible such that they all appear to point in a similar direction. (Figure 2.15(g)).
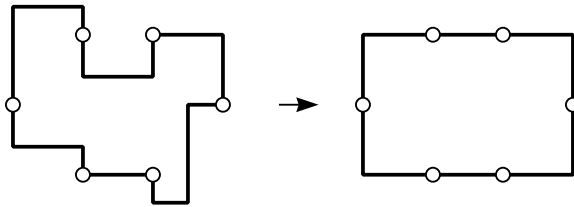
The first attempt at an empirical study into the effects of graph aesthetics on the general understanding of graphs was by Purchase, Cohen and James [PCJ95]. In this work they hypothesised that an increase in either the number of edge bends or edge crossings would lead to a decrease in the understandability of a graph and that an increase in the local symmetry in a graph would lead to an increase the understandability. In order to quantify the understandability of a graph, the performance of a person when answering a number of questions was considered. These questions involved such tasks as finding the shortest path between two given nodes, and finding the minimum number of nodes or edges that need to be removed in order to disconnect a graph. They conclude by saying that attempting to minimise edge crossings and edge bends has a significant positive effect on the aesthetic quality of a graph layout. They were unable to make any conclusive assessment of the effectiveness of the local symmetry hypothesis.

A later study by Purchase [Pur97] adds two more aesthetics, namely maximising the minimum angle between adjacent edges and fixing edges and nodes to an orthogonal grid. This study confirmed the previous findings regarding edge crossings, edge bends and symmetry but the two extra aesthetics did not result in a statistically significant improvement in the understandability of graphs. Both this study and the previous study considered each aesthetic in isolation by comparing graphs with extremes of the same aesthetic. Confounding factors in the graphs could also possibly confuse the results of analysing a particular aesthetic, for example, the graphs used in the latter study for comparing edge crossings appear have longer edges when there are more edge crossings. Later research studies consider and compare graphs that combine several aesthetic features to try to mitigate these weaknesses.

Purchase, Carrington and Allder present the results from three studies [PCA02] which evaluate graph layout based on several aesthetics. The first study looked at individual aesthetic criteria in simple, abstract graphs [Pur00]; the second looked at evaluating the output from several different graph drawing algorithms [Pur00]; the third evaluated aesthetic criteria in the context of the semantic preference for Unified Modelling Language (UML) diagrams [PAC01]. The first two experiments measured the syntactic performance on a number of abstract graphs by asking subjects to carry out tasks relating to the relationships between nodes rather than assessing any implied meaning in the graphs. The tasks used were:
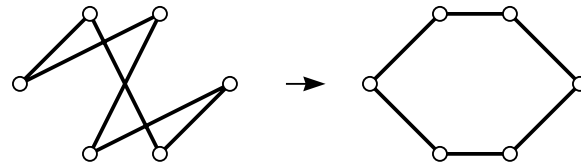
(a) Minimising edge crossings

(b) Minimising edge bends
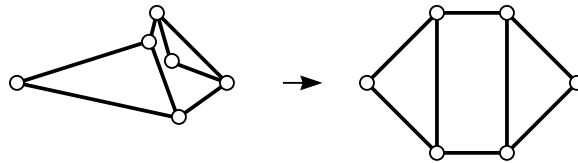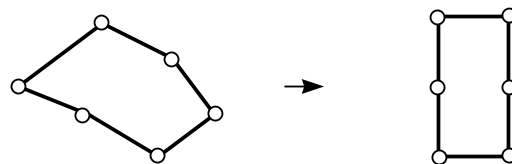
(c) Maximising symmetry
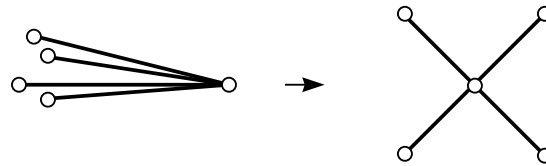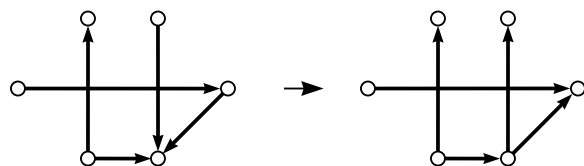
(d) Maximising the minimum angle between edges leaving a node
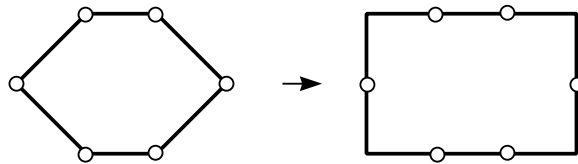
(e) Maximising edge orthogonality

(f) Maximising node orthogonality

(g) Maximising consistent flow direction for directed graphs

Figure 2.15: Examples of seven different criteria that can be used to measure the aesthetic quality of a graph. In each of the examples, two graph embeddings are shown: the left-hand one in each case shows a poor embedding in relation to the criterion and the right-hand one shows a good embedding.

- How long is the shortest path between two given nodes?

- What is the minimum number of nodes that have to be removed such that no path exists between two given nodes?

- What is the minimum number of edges that have to be removed such that no path exists between two given nodes?

The first study evaluated the same five criteria as in [Pur97] with metrics for each criterion being defined and normalised to lie between 0 and 1 (with 0 meaning that the aesthetic would make the graph harder to read and 1 making the graph easier to read). Fifty-five subjects were tested with a selection of graphs using the syntactic tasks described above in an online environment with performance measured in terms of response time and errors. The results of this experiment are consistent with previous studies in that support was identified for reducing the number of edge crossings, edge bends and increasing symmetry, with no support identified for maximising the minimum angle or increasing orthogonality.

The second study compared the results from eight different graph drawing algorithms. The eight algorithms compared were those by Fruchterman and Reingold [FR91] (Section 2.4.4), Kamada and Kawai [KK89] (Section 2.4.4), two variants of an algorithm by Tamassia [Tam87] (Section 2.4.5), Woods [Woo82], de Fraysseix *et al.* [dFPP90], Seisenberger [Sei91] and Tunkelang [Tun94] (Section 2.4.6). Eight variants of the same graph were produced using each of the different algorithms and were tested by fifty-five subjects using the syntactic tasks described above. Performance was again assessed in terms of response time and errors. Variance in response time was not statistically significant across each algorithm. A pairwise comparison showed that the graph drawn with the Seisenberger algorithm was prone to significantly more errors than the Fruchterman and Reingold, Kamada and Kawai, and Tunkeland drawings. The conclusions from this experiment were that it was difficult to assess the quality of one algorithm over another using tasks based on the syntactic understanding of drawings.

The third and final study evaluated UML class and collaboration diagrams with respect to the semantic preferences for drawings. Six aesthetics were considered across both types of UML diagrams: minimising edge bends, minimising edge crossings; orthogonality; width of the layout; text direction and font type. Two additional aesthetics were investigated for UML class diagrams, namely the notation used for inheritance and directional indicators. Two additional aesthetics were investigated for collaboration diagrams: adjacent arrows and arrow length. The usability of sixteen different diagrams for each type of UML diagram (with two diagrams to contrast each aesthetic) was assessed by asking for the subjects' preference of one diagram over another. Each subject also ranked all sixteen diagrams in order of preference. Seventy subjects participated in evaluating the class diagrams and ninety subjects evaluated the collaboration diagrams.

When taking confounding factors into account (for example, a more orthogonal diagram resulted in a greater number of edge bends), minimising edge crossings and increasing orthogonality were the most important aesthetics, while the width of layout and the font type were the least important aesthetics.

Ware *et al.* [WPCM02] contributes to this work by cognitively measuring the aesthetic quality of graphs based on the time taken for the shortest path to be found in each graph. They consider five primary factors (continuity; number of edge crossings; average edge crossing angle; number of branches; shortest path length) as well as two additional factors (total geometric line length and total number of edge crossings). Their experiment tested forty-three subjects. They conclude that the continuation of edges (where edges pass through nodes as straight as possible) contributes most to the quality of the graph when it is used for finding shortest paths. They also claim that their results show that the total number of edge crossings in the graph is less important than the number of edges that cross the shortest path. Finally, they observe that for very short paths with length less than three, the path is possibly perceived by the brain in approximately constant time.

2.4.4 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. As forces tend to apply broadly equally for all nodes, graphs drawn with a force-directed approach tend to have consistent edge lengths. This would allow even spacing of stations as required by our definition of the metro map metaphor (Section 2.3). Section 2.8.1 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 and is calculated in terms of the logarithm of the distance, d, between the nodes:

$$f_a = c_1 \log \frac{d}{c_2}. \tag{2.1}$$

A repulsive force is also applied between non-adjacent nodes using an inverse square law:

$$f_r = \frac{c_3}{\sqrt{d}}. \tag{2.2}$$

The forces should allow for a reasonable separation of the nodes that are non-adjacent while edges are kept to be of roughly similar length. Each iteration of the spring
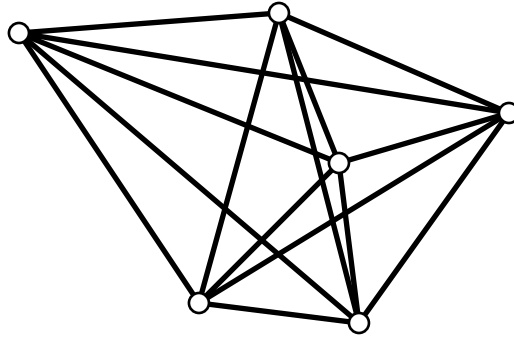
Figure 2.16: Randomised embedding of the complete graph with six nodes, K_6.
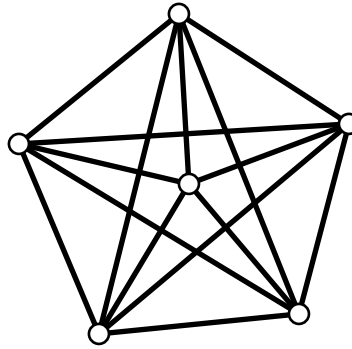
Figure 2.17: Embedding of K_6 using Eades' force-directed algorithm.

embedder requires calculating the net force, f, for each node:

$$f = c_4(f_a + f_r). \tag{2.3}$$

The four constants, c_1, ..., c_4 are determined beforehand.

The spring embedder typically starts with a random embedding of the graph and m iterations are applied until some equilibrium is reached (where the attractive force, a, balances the repulsive force, r). For larger graphs, a greater number of iterations is typically required, but most graphs achieve an equilibrium after around 100 iterations. Calculating the forces on each node in the graph takes time proportional to the number of the nodes in the graph: each iteration of the spring embedder runs with time $O(n^2)$. Figure 2.16 and Figure 2.17 show how the spring embedder lays out the complete graph with six nodes, K_6.

Fruchterman and Reingold [FR91] extend Eades' algorithm by basing the force calculations on an optimal distance between connected nodes, k, which depends on the number of nodes and the dimensions of the drawing area. The intention here is to ensure that nodes are evenly spaced across the drawing area such that there are no

significant voids of nodes and no significant clusters of nodes. k is calculated thus:

$$k = c\sqrt{\frac{wl}{n}} \tag{2.4}$$

where w and l are the width and length of the drawing area, n is the number of nodes in the graph, and c is some constant (determined experimentally). The same repulsion force as Eades' method was used but can be given in terms of k:

$$f_r = \frac{-k^2}{d} \tag{2.5}$$

but a different attraction force is used for efficiency purposes:

$$f_a = \frac{d^2}{k}. \tag{2.6}$$

As it stands, the algorithm runs in $O(n^2 + e)$ time (where n is the number of nodes and e is the number of edges), but approximating the effect of distant nodes allows this to be improved to $O(n + e)$. Unlike Eades, Fruchterman and Reingold were a lot less precise regarding the termination condition for their algorithm. They experimented with making the condition a function of n and e but they could offer no general justification for the number of iterations required for any particular graph.

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 $O\left(n^3\right)$ 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 consistently faster than both the Fruchterman/Reingold and Kamada/Kawai algorithms) [FLM95] and a method that ensures that edge crossings are preserved [Ber99].

2.4.5   Orthogonal Graph Drawing

Orthogonal graph drawing is mainly concerned with drawing graphs where edges are restricted to horizontal and vertical lines [EFK00, EFK01]. Our definition of the metro map metaphor in Section 2.3 means that edges should be drawn horizontally, vertically or with 45° diagonals. Orthogonal graph drawing algorithms are therefore a logical place to look for inspiration regarding this requirement. In many approaches, *polylines* (lines with more than one straight segment; the point between line segments is called a
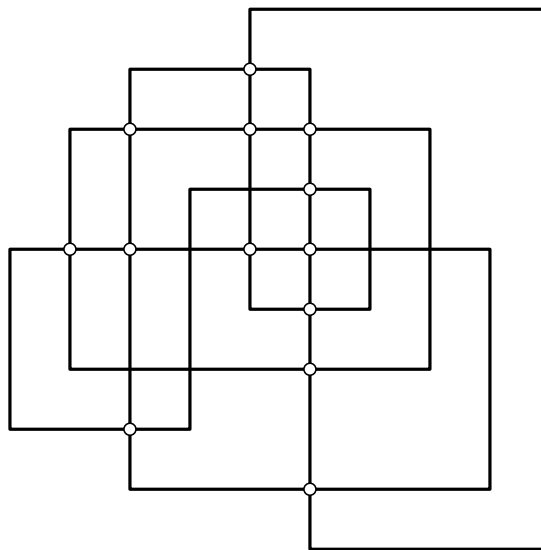
Figure 2.18: An embedding of an orthogonal graph.

bend) are allowed where a line with a single segment cannot be drawn orthogonally (see Figure 2.18). Orthogonal diagrams also have applications outside of graph drawing, particularly in the field of VLSI (very large scale integration) design [Ger99] 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. This means that an embedding is constructed starting with a "clean sheet" with the drawing algorithm specifying the relationships and structure of nodes and edges, rather than starting with a particular embedding and making modifications to it as is the case for force-directed algorithms. Orthogonal diagrams that allow a relaxation permitting some non-orthogonal edges are called *quasi-orthogonal* or *nearly-orthogonal* diagrams [KM98, GM98].

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 a way as to minimise the number of bends. Highly connected graphs are not catered for—the maximum degree of a node for an orthogonal drawing using this algorithm is four. The algorithm runs in $O\left(n^2 \log n\right)$ time where n is the number of nodes in the graph.

Tamassia first considers an *orthogonal representation* for a 4-planar[4] graph. An orthogonal representation describes the shape of the graph without considering any dimensions. The shape is described in terms of each face in the graph being a circularly

---

[4]A 4-planar graph is one which is both planar and has maximum node degree less than or equal to four.
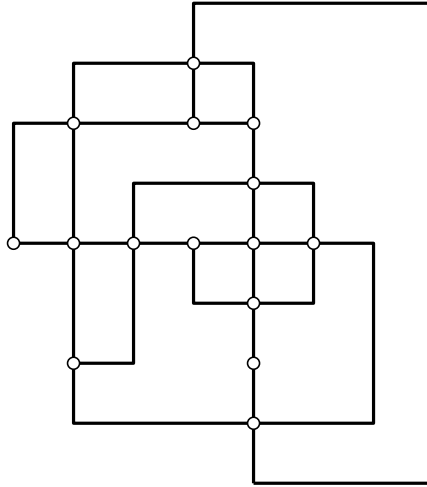
Figure 2.19: An embedding of a planar orthogonal graph.

ordered list of edges. For each edge, a binary string describes the shape of each segment (with '0' representing a bend of 90° and '1' representing a bend of 270°) and a value describes the angle between that edge and the next edge in the list (which will be one of 90, 180, 270 or 360). From this orthogonal representation a grid embedding can be computing by finding the lengths of all the edges. Tamassia's grid embedding algorithm can be distilled to the following steps shown in Algorithm 2.1.

Algorithm 2.1 Embedding a Planar Graph on the Grid (Tamassia) [Tam87]

1: $G \Leftarrow$ a 4-planar graph
2: $P \Leftarrow$ a planar representation of G
3: Construct a flow network $N(P)$
4: Find the minimum cost flow in $N(P)$
5: Find the optimal orthogonal representation, H, for P
6: Compute the lengths of edges in H to obtain an optimal grid embedding, Q, of G

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 their 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 could 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 the representation to allow for nodes
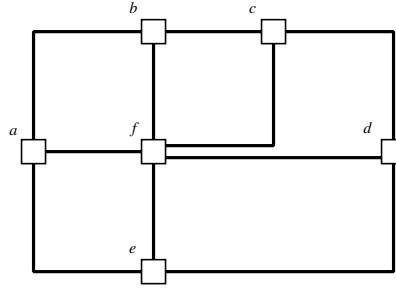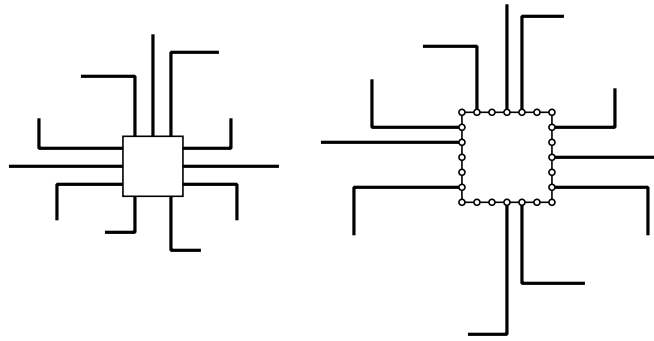
Figure 2.20: An embedding of a nearly orthogonal graph [FK95].

Figure 2.21: An example of replacing a large node with high degree with $8s$ smaller nodes.

with degree greater than four. They call this a *nearly orthogonal representation*, an example of which is shown in Figure 2.20 where node f has degree five. Fößmeier and Kaufmann show an algorithm which first creates a nearly orthogonal representation in $O\left(n^2 \log n\right)$ time. The representation is then used to create an embedding on the grid by computing the lengths of edges in a similar way to Tamassia. To be able to draw the graph using high degree nodes, large nodes are represented by $8s$ smaller nodes (where s is the length of the side of the node) as illustrated in Figure 2.21. 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 placing 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 on an underlying integer grid. The algorithm is able to produce an embedding of a graph in $O\left(e\right)$ 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 be no more than e bends as each edge has at most one

(a) U-turns

(b) Superfluous bends

(c) Poor placement of nodes with degree two

(d) Self crossings

(e) Stranded nodes

(f) Extra area
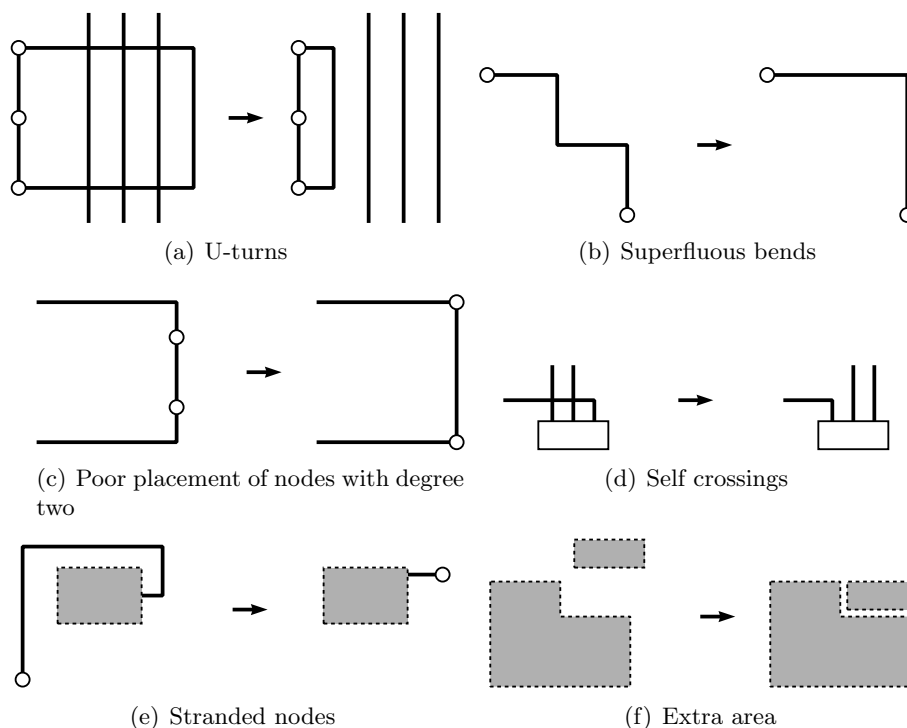
Figure 2.22: Examples of particular problems when drawing orthogonal embeddings of graphs. These problems are dealt with by refinement [SKT00].

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. They 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 from the sketch. Their algorithm runs in $O\left(n^2 \log n\right)$ time. They 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.

A 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. These improvements are illustrated in Figure 2.22. 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.

2.4.6 Optimisation Methods for Drawing Graphs

Optimisation is the study of problems where some function is either maximised or minimised by the systematic selection of values within an allowed range. In the context of graph drawing, this means that an *objective function* is used to determine the aesthetic quality of a graph based on a number of criteria or constraints with some process of modifying the layout of the graph such that the objective function is either maximised or minimised. The use of several criteria to produce an objective function is called *multicriteria optimisation.* Each successive iteration of an optimisation algorithm should produce a more optimal graph in most cases, but some algorithms occasionally allow less optimal graphs in order to avoid local minima.

A number of graph drawing methods based on optimisation exist including a heuristic optimisation approach [Tun94], approaches using genetic algorithms [KMS91, BBS97, RSOR98, Tet98] and the use of simulated annealing [DH96]. Optimisation methods which never allow degradation in the graph are referred to as *hill climbers* and have previously been used for drawing graphs [RSORS99] and Euler diagrams [FRM03].

Heuristic Optimisation

Tunkelang [Tun94] takes the approach of creating an aesthetic cost function which is then minimized using a local optimisation procedure. The algorithm is flexible due to the way in which the aesthetic cost function is modular, meaning that different criteria can be combined as appropriate for the graph being drawn. 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. This algorithm is assessed against the three aesthetic criteria and compared against algorithms by Davidson and Harel [DH96] and Fruchterman and Reingold [FR91]. A test set of thirty graphs is used, split into small and large graphs (around sixteen and sixty nodes respectively) and then into sparse and dense graphs (with average degree of three and four respectively). Tunkelang's algorithm is better with all three criteria for both small and large sparse graphs but does not perform as well as the other algorithms for the dense graphs except in respect to the edge crossings criterion.

Genetic Algorithms

Genetic algorithms borrow the concept of evolution from nature with the intention of maturing a population over a succession of generations. Typically, a population

of candidate solutions is maintained and an attempt to optimise them to form better solutions is made using the principles of selection, recombination and mutation. An initial population is normally chosen at random and then two individuals from that population are selected (possibly with a bias towards better, "pedigree" individuals) based on their *fitness* according to some function. Parts of these two individuals are then combined to form offspring that are then mutated in some way.

Branke *et al.* [BBS97] use a spring embedder (see Section 2.4.4) to perform a local optimisation after the mutation stage. As with other optimisation methods, an objective function based on aesthetic criteria is used to evaluate the fitness of individuals in the population. Recombination is performed using *crossover* where subgraphs of the two individuals being combined are swapped over. Care has to be taken when considering pairs of graphs that are qualitatively equivalent but may actually be shifted, rotated or inverted in the plane. Recombination of pairs like this can result in very poor offspring, so they try to minimise this problem by shifting and rotating one of the graphs so that it is as equivalent as possible to the other. This "competing conventions problem" is a more significant problem in early iterations until the population converges. They use mutation to make small changes to an individual.

Similar genetic algorithms with examples are presented by Tettamanzi [Tet98] and Rosete-Suárez and Ochoa-Rodríguez [RSOR98] where alternative crossover techniques are used.

Owing to the controlled stochastic nature of genetic algorithms, solutions are less likely to get stuck in a local minimum in the search space meaning that finding a more optimal solution is possible. However, the need to maintain a population of individuals can result in much greater running times. As a result, most examples of genetic algorithms for graph drawing are limited to graphs with fewer than 100 nodes.

Simulated Annealing

Simulated annealing is another approach to drawing graphs [KGJV83, Čer85]. It is based on physical processes of the way that liquids cool into a crystalline form (annealing) [MRR+53]. This uses the analogy that the minimum energy state of a physical system (the crystal state) is equivalent to the minimum energy of a simulated annealing system.

In annealing, the probability distribution $P(E)$ for the energy of each state E is a function of the temperature of the system T and the Boltzmann constant k, and is given by the Boltzmann distribution:

$$P(E) = e^{\frac{-E}{kT}}. \tag{2.7}$$

When simulating annealing, a series of sequential moves are made to the system and the probability of a system changing between energy states becomes

$$e^{\frac{-(E_2 - E_1)}{kT}},$$
(2.8)

where E_1 and E_2 are the energies of the old and new states respectively. In simulated annealing, the temperature of the system is artificial, so the assumption is made that $k = 1$.

Davidson and Harel demonstrate an algorithm which is based on simulated annealing [DH96] and illustrated in Algorithm 2.2. As with other force-directed drawing

---

Algorithm 2.2 Simulated Annealing (Davidson and Harel) [DH96]

1: $T \Leftarrow$ initial temperature
2: $\sigma \Leftarrow$ initial configuration
3: $E \Leftarrow \lambda(\sigma)$
4: **while** not terminated **do**
5:     $\sigma' \Leftarrow$ new configuration based on σ
6:     $E' \Leftarrow \lambda(\sigma)$
7:     **if** $E' < E$ or $random < e^{\frac{E - E'}{T}}$ **then**
8:       $\sigma \Leftarrow \sigma'$
9:       $E \Leftarrow E'$
10:    **end if**
11:    $T \Leftarrow \gamma T$
12: **end while**

---

algorithms, they start with a random embedding of the graph, σ, and iteratively generate an improved embedding, with the improvement being measured with an objective function, λ (as in other optimisation methods for drawing graphs). The simulated annealing process produces an erratic improvement with a probability that subsequent iterations are less optimal than the previous iteration. The probability of a less optimal configuration being used is determined by $random < e^{\frac{E - E'}{T}}$ where $random$ is a random real number between 0 and 1. This has the advantage that it is possible to escape from local minima in the search space, thereby increasing the probability that a more optimal solution may be found. The rate at which the temperature decreases is determined by γ and is usually in the range $0.6 \leq \gamma \leq 0.95$. The termination condition is set such that a finite number of iterations are performed. Their algorithm runs in time at most $O(n^2 e)$ where n and e are the number of nodes and edges in the graph respectively. When compared to other force-directed methods for graph drawing, results are reasonably comparable but the running time of the algorithm is generally poor, especially for large graphs (over sixty nodes).
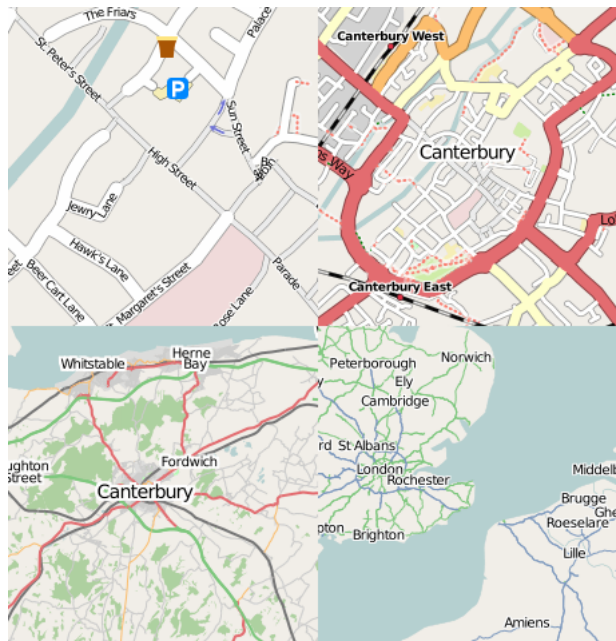
Figure 2.23: Examples of cartographic generalisation required for changing scale [OSM07]. All four map extracts are centred on the same place (Canterbury, England) but the increasing scale as you "zoom out" requires a generalisation of features in order to avoid the map becoming unreadable.

2.5 Schematic Diagrams and Automatic Cartographic Generalisation

2.5.1 Generalisation

Cartographic generalisation is the process of taking a map and modifying it by removing information whilst ensuring that enough detail remains to serve the purpose of the map. Generalisation in this context usually involves selecting the information to display, simplifying information or combining information from a number of features into a single feature. In cartography, this process is necessary for producing maps at different scales or where abstraction of the map is required without changing scale.

When producing maps of differing scale, smaller scale maps will show much less information than larger scale maps to avoid the map becoming too cluttered and illegible. Figure 2.23 shows four map extracts centred on the same place (Canterbury, England), each with increasing scale. The top-left extract is at the smallest scale and shows street names on small roads in the city centre, but as the scale increases, first the street names, then the smaller roads themselves disappear. By the time the largest scale map is drawn, Canterbury is not even marked on the map.

The decision as to which features are removed when a map is generalised is by no means arbitrary. Take the example of the roads in Figure 2.23: on a large scale map it

just is not possible to show all the minor roads because the resolution of the map does not allow this, and even if it was possible it would still be nearly impossible for a user of the map to distinguish these features visually. However, as the scale increases, we remove first the minor roads but leave the main roads and motorways, then many of the main roads are progressively removed. In other words, the order that features are removed as scale increases is dependent on their relative importance. The same applies to the labels for place names: the towns of Whitstable, Herne Bay, Canterbury and Fordwich are less important than London and Rochester. In this case, importance can be determined by considering the population of a town or city or the grade (motorway, major, minor) of roads.

An excellent online example of generalisation is that of Google Maps [Goo07] where it is possible to view maps from a scale of roughly 1:357,000,000 right down to 1:2,700 at 18 different scales.

2.5.2 Automatic Cartographic Generalisation

The advent of computers provided a great resource for cartographers to automate many of the tedious processes involved in creating maps. This became essential when geograpical information systems (GIS) began to appear [HCC98]. The process of selecting which features to show on a map of a particular scale has received a great deal of interest from researchers, including the publication of several books which provide an overview of the research area [BM91, MS92, MLW95].

Generalisation can be broken down into a number of different areas depending on which features are being generalised. *Line generalisation* is one of the most significant areas (and the one most relevant to metro map layout), but other areas include settlement *selection*, where particular point features are selected for inclusion at a specific scale or *combination*, where irregular features (such as trees or houses) are combined into single contiguous features (such as forests or cities). It may also be necessary to generalise scale or to base the map on some other scale basis other than geographical area. A conceptual framework for automated map generalisation is given by Brassel and Weibel [BW88]. Cartograms are an example of this where the apparent size of a particular feature is proportional to a particular value associated with that area [Den99, Cam01]. An example of a cartogram would be a map of the world where the area of each country is proportional to some metric derived for that country, such as the gross domestic product or average life expectancy.

For metro map layout, it is important that the topology is invariant (for example, that stations or interchanges are not removed), so selection or combination are not relevant here. However, line generalisation forms a significant part of metro map layout and so we will focus on that in the next section.
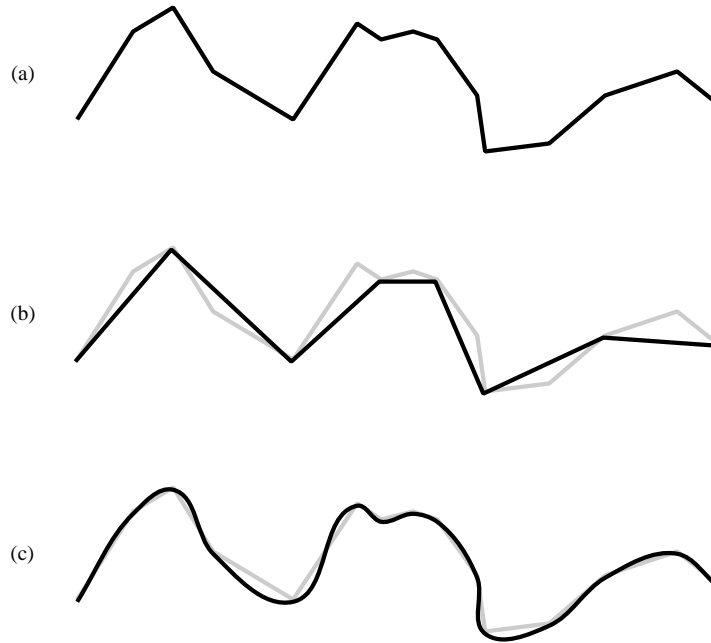
Figure 2.24: Illustration of line generalisation. Line (b) shows an example of a simplification of line (a); line (c) shows an example of a smoothing of line (a).

2.5.3 Automatic Line Generalisation

Line generalisation is the process of taking a line and reducing its geometric complexity. This usually involves simplifying and/or smoothing the line such that the overall form of the line is maintained. Simplification of a line will result in a new line that has fewer geometric points (fewer bends in the line) while smoothing of a line involves replacing straight line segments with curving line segments. These generalisation methods are applicable to both maps and to schematic diagrams. Figure 2.24 shows an example (b) of a line that has been simplified by removing every other vertex and an example (c) of a line that has been smoothed using Bezier curves.

In the context of metro maps, we are not interested in smoothing lines. We are, however, interested in simplifying lines so that metro lines can be drawn as straight as possible. The line generalisation algorithms considered in this section all consider the simplification of lines.

The Douglas-Peucker heuristic [DP73] is very often used as a basis for other line generalisation heuristics and algorithms. The heuristic works by taking a path between two points, $P_0, P_1, \ldots, P_n$, and recursively splitting the line at the point which is furthest from the direct line from P_0 to P_n. An illustration of how the Douglas-Peucker heuristic works is given in Figure 2.25. A generalised path can be found in $O(n \log n)$ time, but as this is a heuristic, it is not necessarily the most optimal solution [HS98]. The original Douglas-Peucker heuristic can occasionally introduce circumstances where the topology of the diagram is modified by inadvertent self-crossings. Saalfield modifies
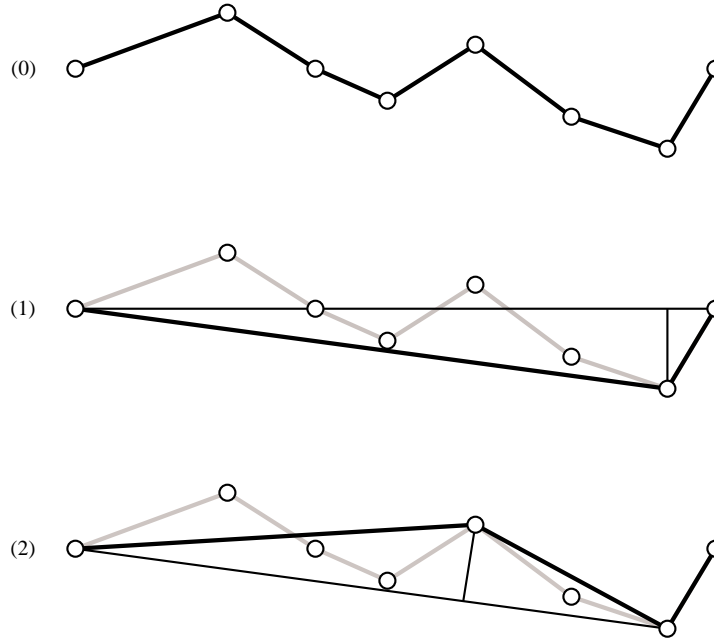
Figure 2.25: An example of line generalisation using the Douglas-Peucker heuristic [DP73]. The original line is recursively subdivided at the node which is furthest from a line between the two end-points until all nodes are within a certain error criterion.

the Douglas-Peucker heuristic to ensure the finished simplifications are topologically consistent [Saa99].

Graph theory is frequently used as a basis for line generalisation [MB93, TR95] as roads or other line features can be represented as a set of edges connecting pairs of nodes. Approaches that use graph theory as a basis include those by Imai and Iri [II86a, II86b, II88] who consider the shortest path (using a breadth-first search) in a directed, acyclic graph representing the network to be simplified. The generalised lines are based on a subset of the points that make up the complex line. Their algorithm runs in $O\left(n^3\right)$ time. Melkman and O'Rourke describe a similar algorithm modified with error criteria (explained below) which runs in $O\left(n^2 \log n\right)$ time [MO88]. Other algorithms which can be applied to line generalisation in this way include those presented by Bose *et al.* [BCC$^+$06], Chan and Chin [CC96], Chen and Daescu [CD03] and Toussaint [Tou85]. Van Kreveld and Peschler describe how road network maps can be generalised using selection based on a number of geometric, typological and semantic requirements [vKP98, Wei96].

In general, an approximation of a line must satisfy some kind of error criteria. A number of error criteria are used such as the parallel-strip (or infinite beam) criterion [CD03, ET94, Tou85], the tolerance zone criterion [BCD$^+$02, CDH$^+$05, II86a, II88, MO88] or the uniform metric [AV00, Goo95]. These three criteria are illustrated in Figure 2.26. The criteria might also be attempting to preserve the distance between the
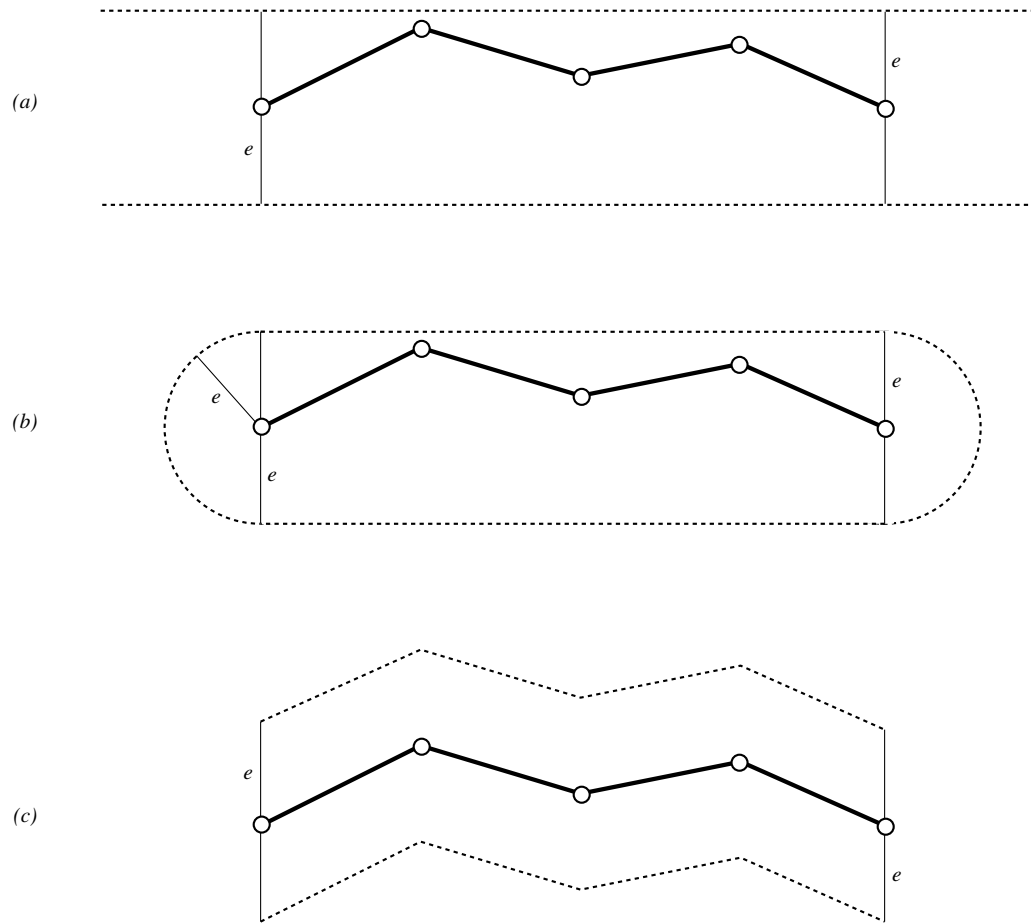
Figure 2.26: Illustration of different line generalisation error criteria. Bounds of errors are shown with dashed lines; e is the acceptable magnitude of the error. Line (a) shows the parallel-strip (or infinite beam) criterion; line (b) shows the tolerance zone criterion; line (c) shows the uniform metric criterion.

two endpoints of the line [GNS07] or minimising the area between a line and an approximation to the line [BCC$^+$06]. Line generalisation algorithms tend to either minimise the error criteria (the min-ϵ problem) or to ensure that the line contains the minimum number of segments, but that the error criteria do not exceed some tolerance (the min-# problem). Error criteria that are too strict will allow for little or no generalisation of the line.

2.5.4 Schematic Diagrams and Route Maps

Schematic route maps are used to depict a route between two points on a map. A common application of route maps is for planning driving directions between two locations. Route maps can depict these directions by massively exaggerating the scale of the start and end points to show the route through city roads and shortening long inter-city motorway sections. This normally involves a process of generalisation so that
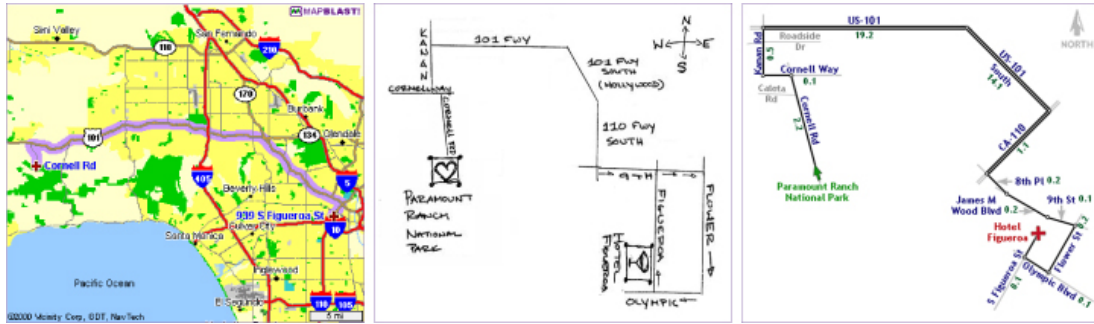
Figure 2.27: Three route maps rendered to show the same route [AS01]. The left-hand image uses a standard geographic map; the middle image is a schematic sketch while the right-hand image is a computer-generated schematic.

the essential topological information for following a route is preserved (such as the intersections and approximate direction), while at the same time exaggerating distances to make the schematic cleaner and easier to read [Ave02, Cab04]. An example of a schematic route map is shown in Figure 2.27. These route maps are particularly useful for route-planning applications and especially with today's modern in-car satellite navigation devices, where the aim is to communicate a route to the driver as quickly as possible even if this means exaggerating some distances.

Agrawala and Stolte describe the LineDrive system that uses three types of generalisation to produce route maps [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 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 to the horizontal or vertical axis.

- Shape generalisation. The exact meandering of a road is not important, so the shape of the road is straightened or smoothed 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. Avelar and Huber [AH01] show a similar method but model their route maps on the characteristics of public transport networks.

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 schematic route maps (particularly intersections), and use their taxonomy to provide an empirical assessment of schematic graphs.

Figure 2.28: Ordering of multiple metro lines passing through a node. In the left-hand example, the ordering of the three metro lines is not preserved as they pass through the node, whereas in the right-hand example, the ordering is preserved.

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). Cabello *et al.* [CdBvDvK01] presents a relatively efficient combinatorial algorithm for the generation of schematic maps which takes into account a number of requirements such as choosing the minimum separation of nodes and not moving nodes. Their algorithm runs in $O\left(n\log^3 n\right)$ time. Later research by Cabello and van Kreveld [CvK02, CdBvK02] implements and evaluates this algorithm.

Elroi describes how the orientation of lines in simplified schematic diagrams can be restricted by placing paths on a grid [Elr88a, Elr88b]. Avelar and Müller present an algorithm for ensuring that a schematic map is topologically correct [AM00]. In schematic diagrams used for wiring diagrams or public transport networks it is important to be able to recognise easily the continuation of wires or metro lines as they pass through components or stations. This problem is illustrated in Figure 2.28. Benkert *et al.* [BNUW07] consider the problem, where several lines or edges run through a node, of ensuring that the relative order remains the same as they do so. Their algorithm also attempts to minimise the number of edge crossings and runs in $O\left(n^2\right)$ time.

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.

These methods are generally successful for the problems that they try to solve. However, the problems discussed in route map schematic diagrams are generally smaller and less complex than would be expected for a metro map. For example, a route map showing a route from a a start point to a destination tends to be linear and involve very few loops; most of the examples presented in the literature for route maps show no more than ten or twenty interchanges.

2.6 Labelling

In many fields, such as cartography, applying labels to 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 features as roads, railway lines or rivers) and area labels (for labelling such things as oceans or countries). In cartographic terms,
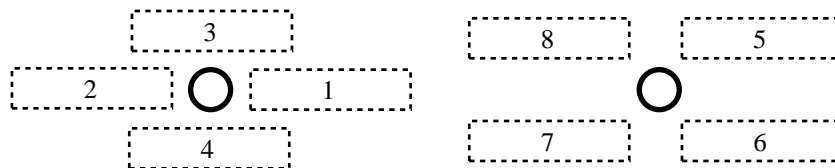
Figure 2.29: Search space for labelling.

these are point-, line- and area-feature labels. For metro map layout, we are mainly concerned with the labelling of point features—in this case stations [WS07].

Map labelling can be a very tedious task, with cartographers only able to place labels manually at a rate of approximately twenty to thirty labels per hour [CJ90]. In particular, the problem of finding a labelling solution where no labels overlap and where no point features are overlapped by labels is NP-complete [KI88, MS91, FW91].

A detailed survey of existing research into point feature label placement is given by Christensen, Marks and Shieber [CMS95]—the main methods used for label placement are discussed in this section, but are briefly discussed in general terms here.

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 [vKSW98]). A typical labelling space is shown in Figure 2.29. 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. Imhof [Imh75] recommends the placement of labels in English to be above and to the right of the point feature[5]. The preferences for label positions in a labelling space are normally dependent on the application—for example, the positions of labels for stations on a metro map is likely to depend on the orientation of metro lines passing through the stations.

In the following sections, the general method used for measuring the quality of labelling solutions is given, followed by a summary of different categories of algorithms that attempt to solve the point-feature labelling problem.

2.6.1 Measuring the Quality of a Labelling Solution

There are a number of principles that are normally followed when labelling point-, line- or area-features. Inhof [Imh75] describes six such principles:

- Labels should be easy to read and easy to locate;

- The association between a label and the feature to which it belongs should be clear;

---

[5]This is because the English language generally uses more ascending characters (b, d, f, h, etc.) than descending characters (g, j, p, q, etc.). In other languages the frequency of use of ascending and descending characters will be different so the preference for label positions may also be different to English.

- Obscuring labels by other labels or other map features should be avoided;

- The context of a label should be clear. For example, labels should describe territories, connections, importance or differentiation between objects;

- The font type of labels should reflect the classification and hierarchy of the feature being labelled (so labels for important features are depicted in a stronger type than less important features);

- Labels should not be evenly dispersed nor be densely clustered.

Deciding how to place labels in the labelling space requires the use of an objective function as the position of each label will affect the quality of the labelling [Imh75]. Describing the objective function in terms of a number of rules is a common approach used by several labelling algorithms that will be discussed later [AF84, FA87, Jon89, CJ90, WB91, DF92].

As a basis for an objective function, Yoeli [Yoe72] uses three criteria for determining the quality of a labelling for point features:

- How much overlap there is between labels and other features of the map;

- A ranked order of preferred label positions;

- The number of points left unlabelled.

These criteria are illustrated by the example map shown in Figure 2.30. In this example, the same map is shown labelled well (the left-hand version) and labelled poorly (the right-hand version) when taking these criteria into account. By providing a metric for each of these criteria, a value can be calculated to provide an objective assessment of the quality of any particular labelling. The values of this objective function for different labellings can then be compared to determine which one is better. The goal of any label placement algorithm or heuristic is therefore the minimisation of the objective function.

2.6.2 Exhaustive Search Labelling Algorithms

These algorithms are based on ultimately performing a complete search of every possible label position [AF84, Nom87, FA87, Jon89, CJ90, DF92]. Each of these algorithms vary only slightly and all use a backtracking methodology so that if a label cannot be placed, the search returns to the last successfully labelled point and tries a different location.

Ahn and Freeman [AF84] were some of the first researchers to investigate the automated placement of labels for point-, line- and area-features. As we are particularly interested in point-feature labelling for metro map layout, we consider only their point-feature labelling algorithm here. They position labels of each feature type based on the degree of freedom afforded by each feature. For example, area-features have the least
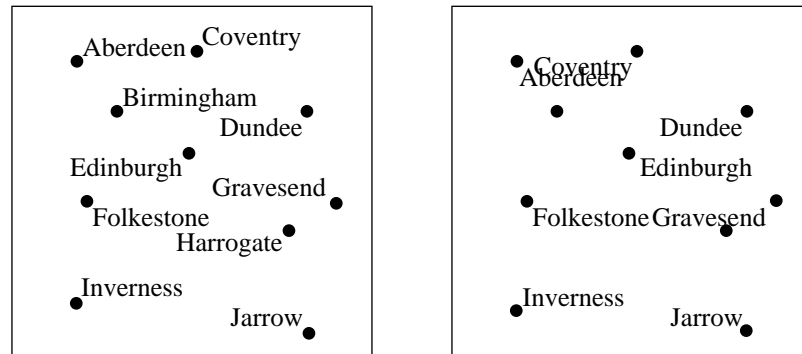
Figure 2.30: Examples of good and bad labellings of the same map. The left-hand version is fully labelled with no overlaps; the right-hand version is missing two labels and has some overlaps.

freedom for placement as the text for the labels tends to be spread out and in larger type than other labels and are therefore positioned first. Line-features have the greatest freedom for placement as the label can be positioned at any point on the line and are therefore positioned last. Their algorithm for point-feature labelling is as follows:

1. A graph is first constructed such that nodes represent the point features and edges connect any two nodes where the label spaces for the point features overlap. Figure 2.31 illustrates this with the right-hand part showing the graph constructed for the labelling spaces in the left-hand part.

2. An unprocessed node with degree greater than zero is selected and each connected node is processed in a breadth-first manner:

   (a) A free-space list of free label positions is constructed. Any potential label positions that already contain labels or other map features are rejected from the free-space list if there is not enough space for the label without causing overlaps.

   (b) A state-space search is conducted where the initial state consists of no labels being placed and the goal state consisting of all labels being placed. The search uses a modified A* search [HNR68] to determine the goal state. The modification involves checking the labels that have the least degree of freedom (such as a larger label) first.

   (c) The algorithm backtracks if no possible location can be found for a particular label.

3. Step 2 is repeated until all nodes with degree greater than zero are processed.

4. Nodes with degree zero can be processed independently as they do not overlap any other point-feature label.
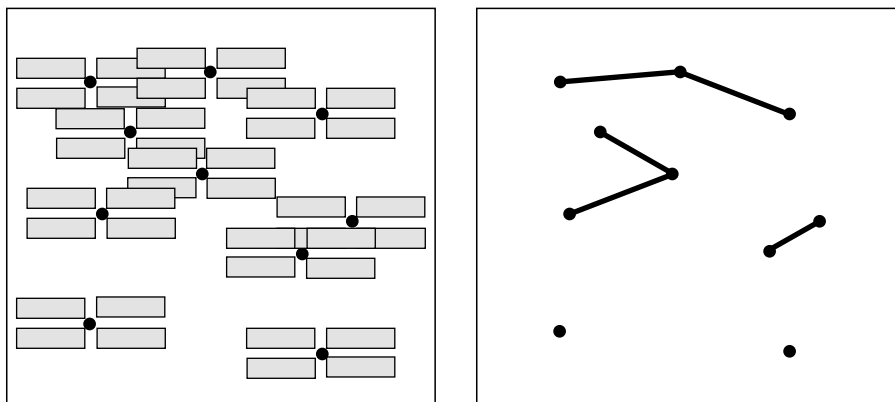
Figure 2.31: Illustration of the Ahn and Freeman algorithm for labelling [AF84]. The left-hand part shows all the possible label positions (the labelling space) for a small example. The right-hand part shows the overlap graph with nodes connected if there was an overlap in the labelling space.
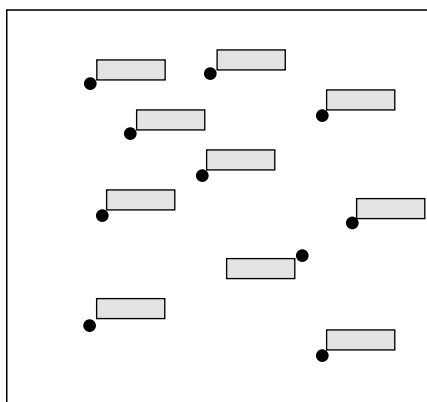
Figure 2.32: Illustration of the Ahn and Freeman algorithm [AF84] for labelling showing a completed labelling for the label space shown in Figure 2.31.

  5. Figure 2.32 shows the completed labelling for the labelling space shown in Figure 2.31.

Exhaustive search is by its nature not ideally suited to situations with large numbers (around 100 or more) of point features. This would be reasonable in all but the largest metro maps where the number of stations tends to be fewer than 100—only the largest metro maps have more than 100 stations. For these larger problems, heuristics can be applied to reduce the size of the search space or to optimise the way in which the exhaustive search algorithm explores the search space [Kor88].

2.6.3   Greedy Search Labelling Algorithms

Exhaustive searches suffer from having too large a search space to explore using the backtracking method. A greedy search can improve on this from a time perspective by
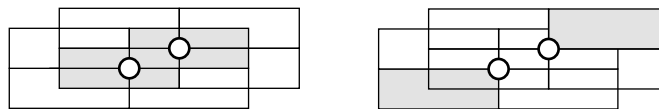
Figure 2.33: Example of a local minimum when labelling using a gradient descent algorithm. The example on the left shows the particular local minimum with positioned labels shown in grey and overlapping each other. Moving either one of the positioned labels does not result in fewer overlaps—the objective function remains unchanged. The example on the right shows the optimal solution to avoid this minimum, but can not be achieved without moving both labels simultaneously.

compromising on the quality of the labelling. Conflicts where labels would overlap can be resolved either by leaving out that label [LP86], leaving it in but making do with the overlap, or obtain the assistance of an expert user [Yoe72].

2.6.4 Gradient Descent Labelling Algorithms

Rather than attempting to label all the features in the map in a single go, it is possible to start with a (random) labelling and make incremental improvements (optimisations) by altering the positions of individual labels. This requires the computation of the objective function for all the labels in the map and an iterative process which calculates the function for each potential label position in the label space for a particular label. The objective function is based on some measure of the aesthetic quality of the labelling as described in Section 2.6.1. The label is then moved to a new location which improves (minimises) the objective function. This method for labelling is analogous to the hill climbing optimisation method for graph drawing as described in Section 2.4.6. The gradient descent method is quite effective [CMS95], but problems with local minima can mean that a more optimal solution is not explored.

Local minima are the largest drawback for gradient descent algorithms. Figure 2.33 shows one such example of a local minimum. In this case, the example on the left has two labels already positioned (shown in grey) so that they overlap each other. It is not possible to move either one of these labels to result in fewer overlaps as each label intersects all four possible label positions—the objective function would remain unchanged. It is necessary to move both labels at the same time to find a more optimal solution with no overlaps. Heuristics can be developed to overcome particular local minima.

2.6.5 Force-Based Optimisation for Labelling

Rather than using a finite set of label positions, it is also possible to use an infinite set of positions and then use a gradient descent algorithm where the position selection for labels is modified. The advantage in this case is that some potential local minima can
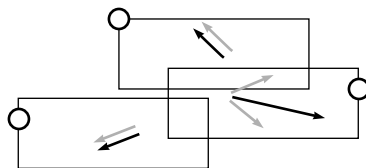
Figure 2.34: Example of overlap vectors for labelling using Hirch's algorithm. The grey arrows indicate the individual overlap vectors from each label overlap; the black arrows indicate the aggregate overlap vectors for each label.
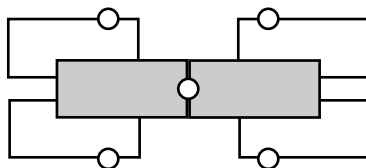
Figure 2.35: Example of a local minimum when using Hirch's labelling algorithm. The label for the central point-feature can end up oscillating between the two locations shown in grey.

be avoided as there is a greater freedom of movement afforded for each label. Clearly the disadvantage is that the search space is now infinitely large, so a heuristic is required to decide where labels can be positioned.

Hirsch [Hir82] describes a method which uses the concept of an overlap vector which can be used to "force" two labels apart. The vector in this case is analogous to a repulsive force of two objects in a physical system. It is also similar to the analogy used in force-directed graph drawing (Section 2.4.4 and [Ead84]). An overlap vector for a large overlap will have a greater magnitude than for a small overlap. Calculation of the vector for a single label is then based on the aggregate of all overlap vectors for each overlap for that label. This is illustrated in Figure 2.34.

Two problems exist with methods using these force-based approaches:

- Errors in calculating the overlap vectors can arise when the vectors have relatively large magnitudes. Large magnitudes become likely when the overlap between two labels is substantial (for example, when one label completely or almost completely obscures another label). An extreme example of this is when two labels occupy exactly the same space: in this case, the magnitude of the overlap vectors becomes infinite and the direction of the aggregate vectors becomes arbitrary.

- Local minima in the search space can still become manifest. Typical local minima with this method include examples where labels oscillate from one side of a point to the other as illustrated in Figure 2.35.

2.6.6   Stochastic Search for Labelling

To handle the local minima cases when using a gradient descent algorithm, some element
of stochasticity can be introduced. This means that each subsequent movement need
not always result in an improvement. Simulated annealing is a common stochastic
optimisation method (introduced in the context of graph drawing in Section 2.4.6) where
the element of randomness of the system decreases in line with an annealing schedule.
Methods for labelling using simulated annealing have been shown by Kirkpatrick *et
al.* [KGJV83], Černý [Čer85] and Edmondson *et al.* [ECMS86]. In these examples, the
problem is usually not with the process of simulated annealing, but with the computation
of the objective function to determine the quality of a particular labelling solution.

2.6.7   Labelling with Linear Programming

Linear programming is a mathematical method which minimises or maximises a particu-
lar objective function by taking into account a set of constraints. Zoraster [Zor86, Zor90]
implemented an algorithm for labelling point features of a map using 0-1 integer pro-
gramming (ZOLP). Finding the optimal solution for ZOLP is NP-hard, so Zoraster
incorporates a number of heuristics to improve results. Zoraster also describes a num-
ber of modifications required to mitigate the risk of ending up in local minima.

2.6.8   Comparison of Map Labelling Methods

The previous sections highlighted a number of different methods for map labelling of
point features. Christensen *et al.* [CMS95] analysed some of these methods empirically
to see how they performed on maps with varying numbers of point features. They
analysed the following methods:

- Simulated annealing (see Section 2.6.6);

- Zoraster [Zor90] (linear programming) (see Section 2.6.7);

- Hirsch [Hir82] (force-directed) (see Section 2.6.5);

- Gradient descent (see Section 2.6.4);

- Greedy (see Section 2.6.3);

- Random

Their experiment plotted fixed-size labels (30 x 7 units) for point features in a region
with size 792 x 612 units. These dimensions were chosen so as to fit a typical map
printed on an 11 x 8.5 inch page. The number of point features (the problem size)
varied from 50 through to 1500 and 25 random layouts were produced for each number
of point features. The quality of the output after running each method was determined

by counting the number of overlapping labels and averaging across all layouts for that method at that problem size. A second experiment was carried out in a similar manner but this time using a real-world map of the state of Massachusetts in the United States of America.

Their conclusions for both experiments were that simulated annealing provided significantly better results when compared to the other five methods. Of the four other non-random methods, the Zoraster and Hirsch methods performed better for smaller maps (fewer than 1000 point features for the random maps and 150 point features for the Massachusetts map[6]) while the gradient descent and greedy methods performed better for larger maps.

2.6.9 Labelling Graphs

While most of the research on labelling is concerned with cartographic maps, it can also be applied to labelling graphs. An embedding of a graph can be considered as a very generalised map with nodes being treated as point features, edges being treated as line features and faces being treated as area features. Kakoulis and Tollis [KT98] show a method for labelling graphical features specifically for graphs. Their approach first 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. They show results of their algorithm for labelling both hierarchical graph embeddings[7] and orthogonal graph embeddings[8]. Their results show that their algorithm produces a better labelling for orthogonal embeddings than for hierarchical embeddings. They also show that the size of the labels is a significant factor in being able to produce a successful labelling of a graph.

2.7 Applications of the Metro Map Metaphor

While the majority of examples of diagrams using the metro map metaphor are of public transport networks, there are a number of existing applications that use the metaphor to illustrate other abstract concepts. Burkhard and Meier [BM04, BMR+05] show how the metro map metaphor can be applied to the visualisation of project plans (Figure 2.36). They created a large wall-mounted poster showing the plan for a project involving a number of groups of people with lines representing each group and stations representing milestones in the project. The poster was displayed in a prominent location

---

[6]These figures cannot be compared directly owing to the difference in scale of the random and Massachusetts maps.

[7]A hierarchical graph embedding is one that constrains nodes so that they lie on equally-spaced horizontal lines called layers.
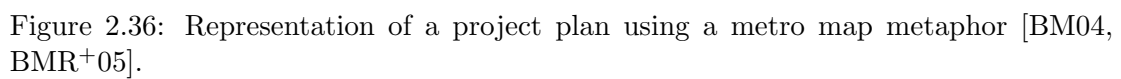
[8]An orthogonal graph embedding is one where edges are composed entirely from horizontal or vertical segments.

in a communal area in the organisation. They found that people were more readily attracted to the poster and that interest in the project plan was increased. Most people surveyed found the metaphor to be easy to understand but said that they would prefer a more interactive version of the map that was available online. The map was hand-drawn and it was noted that a static display would not be suitable for a project plan that frequently changed without some automation of the drawing process as drawing the map by hand would take too long.

An "open source map" produced by the publisher O'Reilly in 2003 [ORe] illustrates the relationships between different computing technologies (Figure 2.37). The map shows each technology (such as "Perl", "UNIX" or "Java") as a separate line with stations representing publications by O'Reilly. Another map of cancer pathways was created by Hahn and Weinberg and designed by Claudia Bentley [HW02] (Figure 2.38). In this map, the progression of the biological steps involved in cancer are shown. As this is a progressive map, lines are augmented with the direction of the pathways and whether that step is activative or inhibitive.

Other examples of the use of the metro map metaphor include the visualisation of the "trains of thought" that run through a Ph.D. thesis [Nes04] (Figure 2.39), the organisation of web-based learning resources [BGH02] and the illustration of guided tours on the web [SGSK01]. Enterprise portals can also be visualised using the metro map metaphor [Zie04] where pages, functions and links in the portal are represented by stations and lines in a metro map.

The usefulness of the metro map as a metaphor is somewhat limited to simple examples by the time required to manually produce these maps. As such they are generally only useful for applications that do not change frequently. This limitation could be removed by quality methods for the automatic drawing of metro maps from abstract data.

Figure 2.36: Representation of a project plan using a metro map metaphor [BM04, BMR$^+$05].

Figure 2.37: Open source route map [ORe].
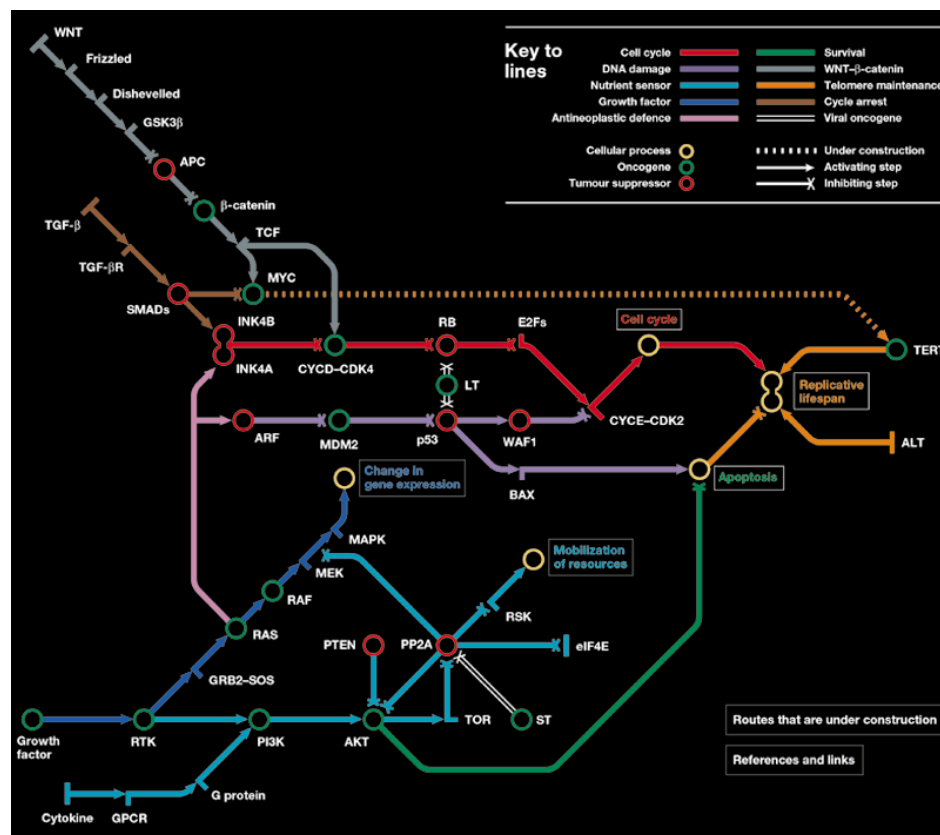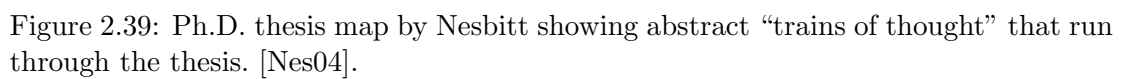
Figure 2.38: A subway map of cancer pathways by Hahn and Weinberg (designed by Claudia Bentley) [HW02].

Figure 2.39: Ph.D. thesis map by Nesbitt showing abstract "trains of thought" that run through the thesis. [Nes04].

2.8   Existing Automatic Metro Map Layout Research

This section introduces and describes existing research that directly tackles the metro map layout problem. Existing approaches include the use of a force-based method (Section 2.8.1), a mixed-integer programming method (Section 2.8.2), path simplification (Section 2.8.3) and centrality-based scaling (Section 2.8.4). None of the research presented here has empirically evaluated the output.

2.8.1   Force Directed Metro Map Layout

Hong *et al.* use a force-directed approach to laying out metro maps [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 and replacing them with a single edge. This reduces the number of nodes and edges in the graph that need to be considered during the drawing process[9]. They also include a final step of labelling the graph which uses a combinatorial approach to try to achieve a labelling with as few a number of overlaps as possible.

They analyse their methods with regards to a set of four 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 $45°$.

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 considered at all. 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, especially 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 the four criteria. Figure 2.40 shows their result of drawing the Sydney CityRail map and Figure 2.41 shows their London Underground result. Their best method is their fifth method which seems to produce the most

---

[9]Adequate space still needs to be left on the single edge so that the removed nodes can be replaced at the end of the drawing process. The nodes are replaced so that they are equally-spaced along the edge.
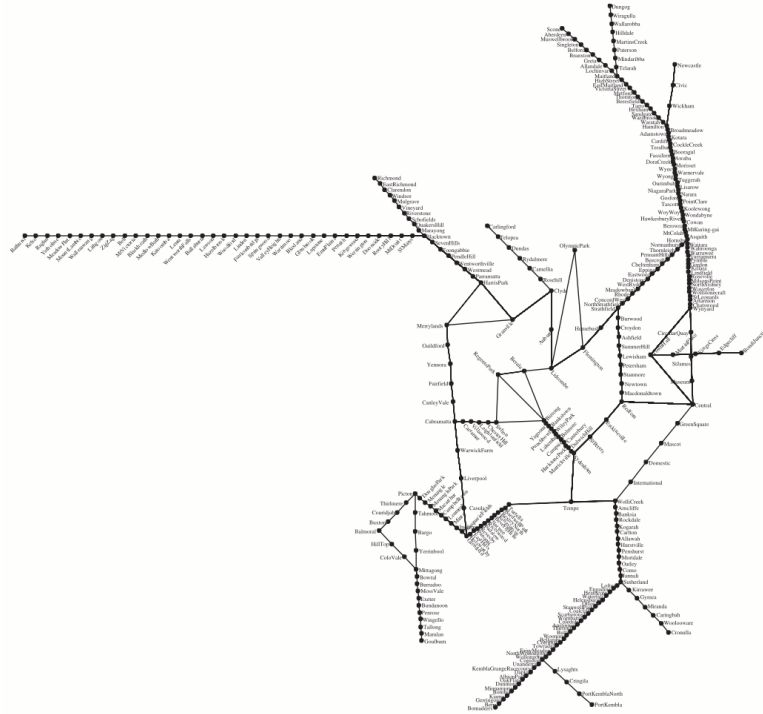
Figure 2.40: The Sydney CityRail map as drawn by Hong *et al.* [HMdN06].

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. Also, they do
not consider drawing whole metro lines; each pair of connected nodes are connected by
only a single edge when in many real-world examples, many edges (different metro 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 (see Section 2.2.3). 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.

2.8.2 Mixed-Integer Program for Metro Map Layout

Nöllenburg describes a method of drawing metro maps using mixed-integer linear pro-
gramming [Nöl05]. Linear programming (LP) is a combinatorial optimisation method
which aims to minimise or maximise a particular linear objective function subject to a
set of linear constraints [CR99a, CR99b]. Mixed-integer linear programming (MIP) ex-
tends linear programming and introduces the notion of restricting variables to be within
certain discrete integer ranges.
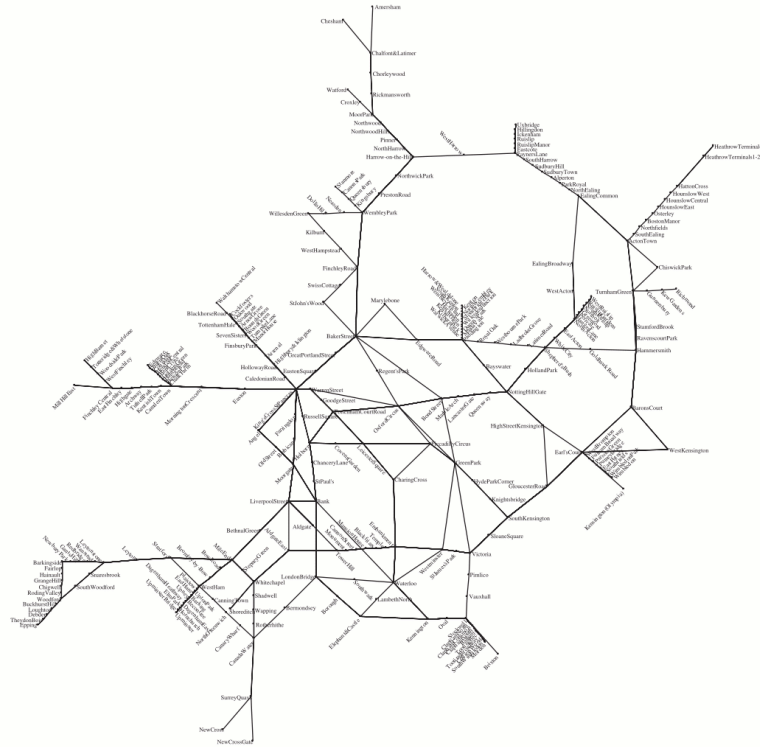
Figure 2.41: The London Underground map as drawn by Hong *et al.* [HMdN06].

As with other metro map layout methods, Nöllenburg models a metro map as a graph. Four *hard* constraints (H1, ..., H4) and three *soft* constraints (S1, S2 and S3) are defined. The hard constraints relate to the linear constraints of an MIP and the soft constraints relate to the linear objective function. The set of hard constraints is as follows:

- **H1.** The output drawing must preserve the topology of the input drawing.

- **H2.** Every edge in the drawing must be composed of octilinear segments (horizontal, vertical or 45° diagonal).

- **H3.** Every edge in the drawing has a minimum length and edges cannot be shorter than this length.

- **H4.** Every edge in the drawing has a minimum distance from non-incident edges. This effectively gives rise to MIP constraints that lead to planarity.

The set of soft constraints is as follows:

- **S1.** Each line should have as few bends as possible.

- **S2.** The total length of all edges in the drawing should be small.

- **S3.** The relative position of neighbouring nodes should be preserved as much as possible.

These constraints are combined in a mixed-integer linear program. As many real-world examples of metro maps are fairly large (the London Underground map has around 300 stations), the number of variables and constraints in the resulting MIP can be enormous[10]. To counter this, Nöllenburg describes using a number of heuristics to reduce the size of the MIP, particularly with regard to the constraints relating to planarity (H4) which can form up to 90% of the MIP. These heuristics are:

- **3-link heuristic.** This heuristic replaces long lines of nodes with degree two with a line containing at most two intermediate nodes (allowing up to two bends in 3-link lines). This extends the ideas introduced in [HMdN06] and [SR04] which replaces entire lines of degree two nodes with a single straight-line edge and can result in a significantly smaller graph and therefore a smaller MIP.

- **Face method heuristic.** This heuristic works by ignoring adjacent edges of the same face as there is no way they can intersect (assuming that the face has a finite area).

- **Convex hull heuristic.** This heuristic helps reduce planarity constraints (H4) by not checking the distance between non-incident edges that are on opposite sides of the outside face. The heuristic works by constructing a convex hull around the input layout and partitioning the space by joining adjacent vertices of the hull with dummy edges. These smaller faces are then used when checking for intersecting pairs of edges.

- **Pendant edge heuristic.** This heuristic comes about from an observation that most intersecting edges involve pendant edges (edges that lead to a node with degree one—for example, a branch line on the metro system). Under this heuristic, intersections between pairs of edges are only checked when at least one of the edges is a pendant edge.

Finally, Nöllenburg explains how the requirement of labelling stations with their names can be incorporated into the MIP. The case when labelling nodes along lines that were collapsed by the 3-link heuristic is handled in some detail by considering a

---

[10]For example, the MIP for the London Underground map still contains over 1.2 million constraints and over 300,000 variables even after applying the two optimisation heuristics described. The Montreal metro is small with 65 stations but the MIP for this map still has 38,000 constraints and 10,000 variables. Neither of these examples includes labelling: Nöllenburg illustrates labelling with the S-Bahn Rhein-Nekar map consisting of 108 stations. In this map, the number of constraints and variables increases by over seven times when doing labelling compared with not doing labelling with 750,000 constraints and 185,000 variables for the MIP including labelling.
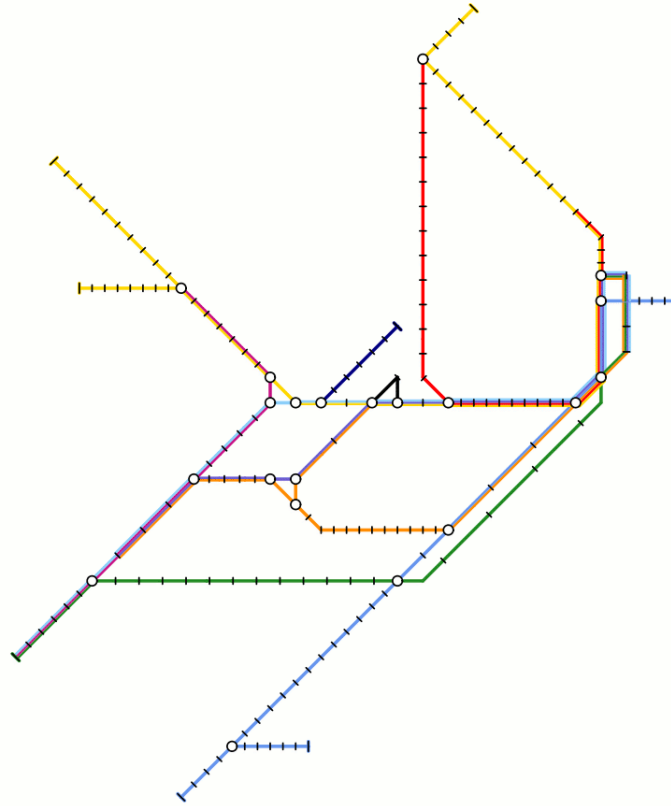
Figure 2.42: The Sydney CityRail map as drawn by Nöllenburg [Nöl05].

parallelogram-shaped area that fits around all the labels along a straight segment of a line.

There are several examples of the output of the MIP for real-world metro maps. Figure 2.42 shows the finished layout of the Sydney CityRail network and Figure 2.43 shows the finished layout for the London Underground map. The Sydney map highlights the effectiveness of their handling of long lines of degree-two nodes using the 3-link heuristic as the spacing of nodes is very regular. The London map is clearly more challenging by way of the highly interconnected central area. This seems to cause the spacing between adjacent nodes to become irregular when taking the whole map into account. Interestingly, some prominent features of the published map (Figure 1.3) are retained, including the characteristic sideways-bottle shape of the Circle (yellow) Line and the horizontal Central (red) Line. There are also some acute angles in some of the lines, particularly in the Victoria (light blue) Line around Euston in the north of the map. Neither of these examples include station labels. This may be because both of these maps are relatively large and the resulting MIP including labelling constraints would be too large to optimise. Nöllenburg shows the results for labelling the S-Bahn RheinNeckar map but state that the layout took 20 hours to produce, rather than 43 seconds for the equivalent unlabelled map.
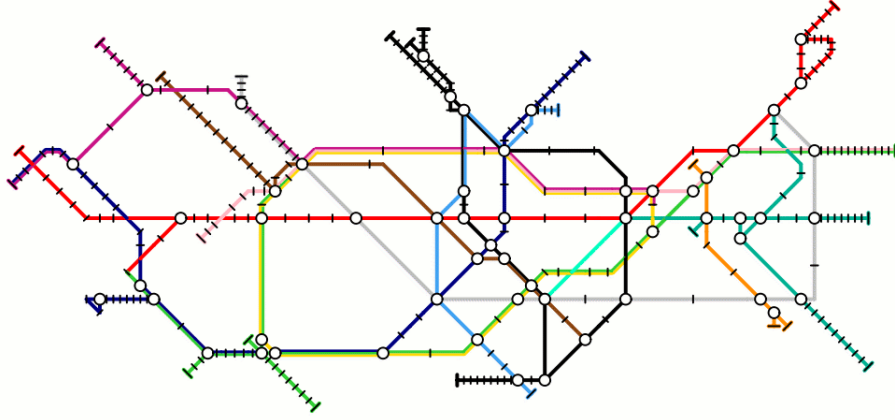
Figure 2.43: The London Underground map as drawn by Nöllenburg [Nöl05].
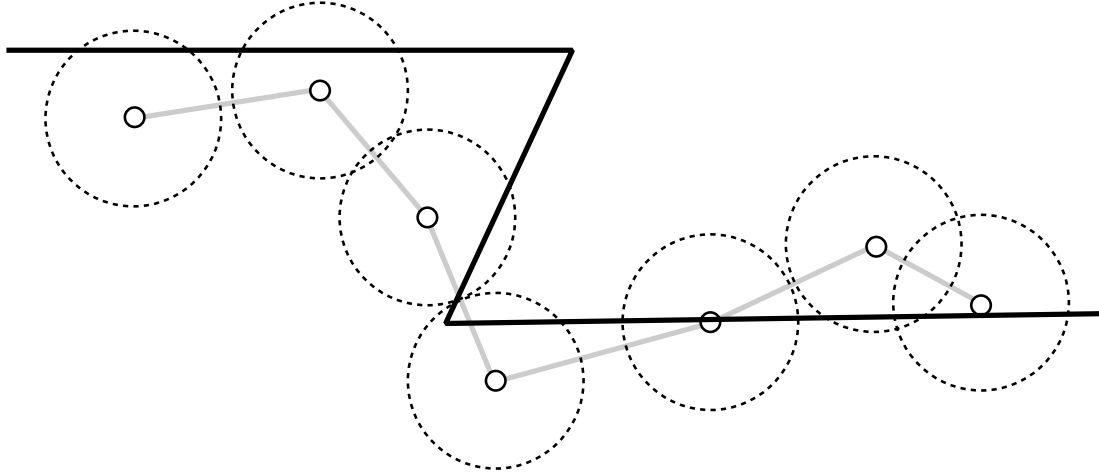
Figure 2.44: Path simplification by stabbing ϵ-circles. The line being simplified is shown in grey with ϵ-circles shown with dotted lines. A simplified line that stabs all the ϵ-circles is also shown.

2.8.3 Path Simplification for Metro Map Layout

Merrick and Gudmundsson describe a method of path simplification which restricts the number of directions that segments of lines can take [MG07]. They call their method *C-directed path simplification* where C is the set of possible directions for line segments. Their method draws on previous line generalisation work discussed in Section 2.5.3. The simplification method imposes the restriction that the simplified line must intersect (stab) the ϵ-*circle* for each point on the input path. The ϵ-circle is analogous to the error criterion used in other line generalisation algorithms. The difference here is that the ϵ-circle is a closed disc with radius ϵ centred on each point of the input path. An illustration of ϵ-circles is shown in Figure 2.44. Their algorithm runs in $O\left(|C|^3 n^2\right)$ time.

Merrick and Gudmundsson explain how their C-directed path simplification algorithm can be extended to the layout of metro maps. As with other metro map layout methods, they use a graph model. Their path simplification algorithm must be extended to handle multiple intersecting lines. To do this, they consider the *importance* of each of the lines in the graph. The importance could be manually defined, but in this case, they use a heuristic function based on the number of nodes on the line that intersect other lines (interchange stations). They then proceed with the most important line and simplify each line in order of decreasing importance. When a line intersects other lines that have already been simplified, they split it into a set of lines so that each has at most one fixed node. To even out the spacing of nodes along lines, they are prepared to allow errors greater than ϵ if this produces an even spacing between nodes. They also introduce two further modifications to their algorithm: the maximum angle that a line is allowed to bend and a minimum link length in the simplification. However, they do not consider the important task of labelling each node with its name.

They illustrate their algorithm by showing results for the Sydney CityRail map and the London Underground map (Figures 2.45 and 2.46 respectively). In the case of their London result, they first used a centrality-based scaling algorithm (see Section 2.8.4) to reduce the relative density of the centre of the map [MG06]. Both the results satisfy the restriction of using only a limit of eight directions, meaning that every edge is octilinear. There are several examples of the topology of the original map not being maintained, particularly in the London map (for example, the loop in the line in the top-right corner of the London map is reversed), but they do not claim that topology will be preserved by their method. The lack of any labelling and any colour to represent metro lines significantly reduces the usefulness of the maps. However, the key advantage of their algorithm is that it produces results in a relatively quick time.

2.8.4   Centrality-based Scaling

One common characteristic of metro maps is that the central area of the map tends to be highly dense with a lot of interconnections, while the extremities of the map become less dense. This is due to the metro systems being focused around the centre of the town or city which they serve. Merrick and Gudmundsson use a method of centrality-based scaling to apply an increased scale to the centre of the map relative to the extremities [MG06]. This is analogous to the "convex effect" described in Section 2.2.1.

The scaling algorithm uses the concepts of *time-distance mapping* [SI03] and *centrality* [WF94]. Time-distance mapping scales a map of a transport network such that the Euclidean distance between nodes is determined by the time taken to travel between those nodes. Centrality assigns a measure for each node in the graph of the localised density of the graph at that point. A number of measures of centrality are detailed by Merrick and Gudmundsson and include *degree*, *betweenness* and *hubness*. A number
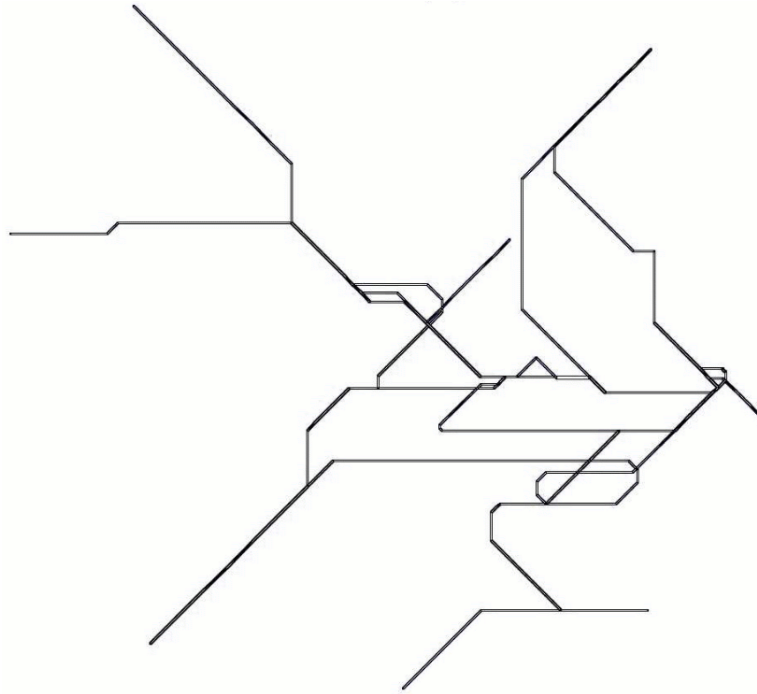
Figure 2.45: The Sydney CityRail map as drawn by Merrick and Gudmundsson [MG07].
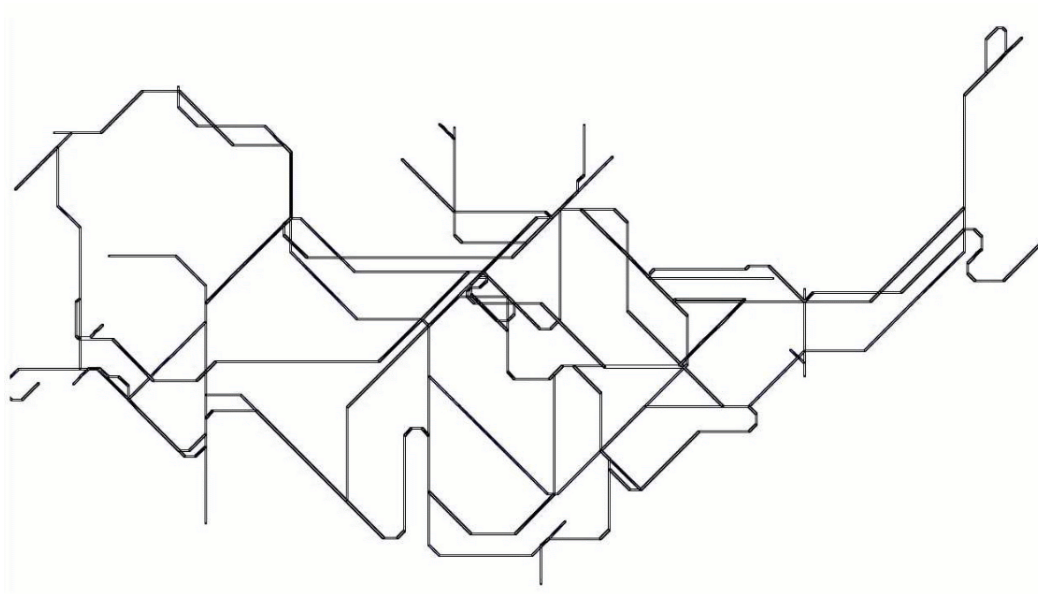
Figure 2.46: The London Underground map as drawn by Merrick and Gudmundsson [MG07].

of these measures of centrality are taken from research on social networks where it is necessary to have measures of the relative importance of each node in the social network.

Degree centrality applies a measure to each node based on the number of incident edges (the degree) to the node. Degree centrality is therefore a local measure of how connected a node is—in a metro map, a node with high degree represents am interchange station on several metro lines. For a node v, degree centrality, $C_{D(v)}$, can be calculated as

$$C_{D(v)} = \frac{\rho(v)}{|V| - 1} \tag{2.9}$$

where $\rho(v)$ is the degree of v and $|V|$ is the number of nodes in the graph.

Betweenness centrality [Fre77] measures the proportion of all possible paths through a graph that pass through a particular node. This gives a measure of the importance of a node in relation to the flow of a network—for a metro map, a node with high betweenness would represent a station through which a high number of services would pass. Betweenness centrality, $C_{B(v)}$ can be calculated using

$$C_{B(v)} = \sum_{s \in V} \sum_{t \in V} \frac{\sigma_{s,t}(v)}{\sigma_{s,t}}, s \neq t \neq v \tag{2.10}$$

where $\sigma_{s,t}$ is the number of paths from node s to node t and $\sigma_{s,t}(v)$ is the number of such paths that also pass through node v.

Hubness centrality is derived from the definition of *hubs* and *authorities* by Kleinberg [Kle99] for ranking the importance of web pages. An authority is a resource that forms a primary source of information on a topic while hubs are resources that act as guides in referring to authorities. For example, a web search engine would act as a hub and search results that appear at the top of the list are most likely to be authorities. This results in a similar measure of centrality to degree centrality but takes into account a wider view than just neighbouring nodes. Hubness centrality, $C_{H(v)}$, can be calculated using

$$A^T A C_{H(v)} = \lambda C_H \tag{2.11}$$

where A is the adjacency matrix for the graph and λ is the highest positive eigenvalue of A.

The scaling algorithm then adapts the time-distance mapping method of Shimizu and Inoue [SI03] to incorporate the measures of centrality. Time-distance mapping alters the lengths of edges in a graph to represent the weighting of an edge (such as the time taken for a train to travel between two adjacent stations) by the Euclidean distance between the endpoints of the edge. The desired length of an edge between nodes u and v is then calculated by

$$d'(u, v) = \alpha \left(1 + \left(\beta \frac{C(u) + C(v)}{2} \right)^\gamma \right) \tag{2.12}$$

where $C(u)$ and $C(v)$ are centrality measures, α and β describe the scale of the output and γ defines the rate at which edge length decreases in respect to the centrality. Merrick and Gudmundsson determined appropriate values for α and β using "extensive experimentation" for each input graph. They do not explain this process of experimentation, but use the values $\alpha = 10.0$ and $\beta = 50.0$. γ was set at either 0.5, 1.0 or 2.0 giving a square-root, linear or quadratic scaling for each centrality. The effect of the centrality measure becomes more pronounced as γ increases.

The centrality-based scaling algorithm is illustrated by the Sydney CityRail and London Underground maps. The London Underground map in Figure 2.46 used centrality-based scaling before applying Merrick and Gudmundsson's path simplification algorithm.

2.9   Conclusion

This chapter provided a summary of the various research areas that are relevant for the research presented in the rest of this thesis. Of particular relevance was the discussion around the characteristics of metro maps in Section 2.2 which led to our definition of the metro map metaphor in Section 2.3. This definition forms the key basis for our method and was the motiviation behind many of the criteria that we chose.

We take forward much of the research on graph drawing aesthetics as discussed in Section 2.4.3. Our multicriteria optimisation method is inspired by the various research on optimisation methods for drawing graphs (Section 2.4.6). While we do not explicitly utilise the other graph drawing or schematic and cartographic generalisation techniques, they served the purpose of informing our ideas and decisions. The existing research on labelling (Section 2.6) informed our intention to use a similar optimisation method to discrete gradient descent as described in Section 2.6.4.

The final two sections presented the state of the art regarding our specific application area: that is, drawing metro maps (Section 2.8) and applying the metro map metaphor to other application areas (Section 2.7).

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 2.2. 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 (clusters) simultaneously is required. Moving clusters of nodes is described in Chapter 4.

3.1 Aims

There are a large number of characteristics that are apparent on existing metro maps. These characteristics are discussed in detail in Section 2.2 which lead to the definition of the metro map metaphor in Section 2.3. We will take these characteristics into consideration as part of our layout method which leads to the following aims:

- **Metro lines**

  - Metro lines will be drawn to take advantage of parallel lines and lines of common angle using angle generalisation—more specifically, lines should be drawn *octilinearly* (that is, horizontally, vertically or with 45° diagonals).

  - Metro lines should have stations spaced at regular intervals.

  - Metro lines should appear to pass straight through stations without changing direction.

  - The scale of the map should be generalised.

  - The topology of a geographic map should be maintained.

- **Labelling**

  - The maps will have station labels with horizontal text.
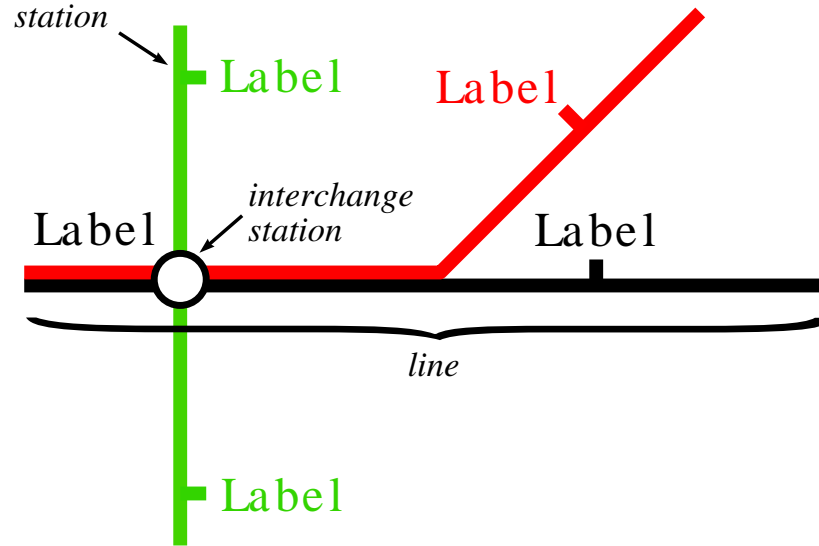
Figure 3.1: Metro map features.

- Labels should not occlude other labels, lines or stations.

- The number of potential positions for a label will be limited.

- Labels will be positioned such that they take the orientation of the line, the position of labels for neighbouring stations and the proximity of other unrelated stations into account.

These aims will be covered by the node movement and label movement criteria described in Sections 3.5 and 3.9.

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 an edge to represent a single connection between two stations. In some cases, there may be several edges connecting two nodes where two or more metro lines run together. We use the term *metro 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). Edges also have metadata in the form of a colour that identifies which line they are part of. These features are illustrated in Figure 3.1.

We decided to use a graph model to represent metro maps, mainly because a metro map is clearly equivalent to a graph. A graph also gives us the programmatic flexibility to make tasks such as finding neighbouring nodes or incident edges relatively straightforward.
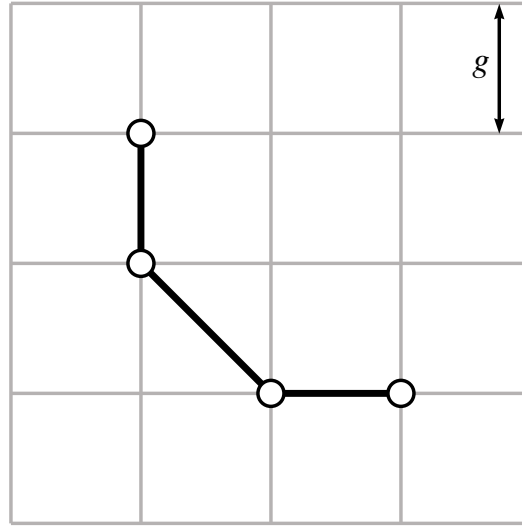
Figure 3.2: Grid used for embedding the metro map graph where g represents the grid spacing.

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 in an orthogonal manner. The spacing between adjacent intersections in the grid is denoted by g. Making the search space discrete 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

Multicriteria optimisation was introduced in terms of graph drawing in Section 2.4.6. Algorithm 3.1 shows an overview of the process for a straightforward, generic hill climbing multicriteria optimisation method. The important parts of multicriteria optimisation are the objective function used to assess the quality of the entity being optimised (calculateCriteria()) and the manner in which an attempt is made to make some improvement (line 4). The termination condition for multicriteria optimisation is simply the point at which no further improvement can be made (line 6)—there is no need to continue beyond this point as the method is completely deterministic and c will never change.

Existing methods for drawing metro maps using a spring embedder, linear programming and path simplification were discussed in Section 2.8. We therefore considered a number of other optimization methods such as simulated annealing (Section 2.4.6) and genetic algorithms (Section 2.4.6). However, we found that the simpler method using hill climbing multicriteria optimisation was more appropriate for this application.

Algorithm 3.1 Multicriteria Optimisation

 1: $c_0 \Leftarrow$ calculateCriteria()
 2: $running \Leftarrow$ **true**
 3: **while** $running$ **do**
 4:    attempt to make some improvement
 5:    $c \Leftarrow$ calculateCriteria()
 6:    **if** $c \not\prec c_0$ **then**
 7:      $running \Leftarrow$ **false**
 8:    **else**
 9:      $c_0 \Leftarrow c$
10:    **end if**
11: **end while**

Simulated annealing adds an element of non-determinism in order to escape from local minima in the search space. However, this stochastic behaviour meant that a larger number of iterations would be necessary to reach a minimum in the search space. As such, we generally found that it was quicker and easier to cater for specific local minima (for example, by introducing clustering and partitioning) 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.

The likelihood of finding a sub-optimal local minimum in the search space is the main disadvantage of hill climbing multicriteria. As the method is completely deterministic, once a local minimum has been encountered the method is stuck there. As such, specific enhancements are required in order that it is more likely that a more optimal local minimum is found by the optimiser. However, these enhancements can be tailored to suit the particular application which means that it is possible that a more optimal local minimum is found more quickly than the non-deterministic methods of simulated annealing or genetic algorithms.

3.3.1   Metro Map Layout with Multicriteria Optimisation

Our process for drawing metro maps is built on the basic multicriteria optimisation method in Algorithm 3.1. Algorithm 3.2 illustrates an overview of our complete multicriteria optimisation method. Here, a graph representing a metro map, G, consists of a set of nodes, V, a set of edges, E, and a set of node labels, L. The section from line 6 to line 12 handles moving nodes (see Sections 3.5 to 3.8), the section from line 13 to line 20 handles moving clusters of nodes (see Chapter 4) and the section from line 21 to line 27 handles moving node labels (see Sections 3.9 and 3.10).

The process runs as long as improvements to the graph are still being made. The termination condition is given at line 29. In practice, we used a predefined number

Algorithm 3.2 Metro Map Layout

1: $G \Leftarrow (V, E, L)$
2: snapNodes(V)
3: $m_{T0} \Leftarrow$ calculateNodeCriteria(V) + calculateLabelCriteria(L)
4: $running \Leftarrow$ **true**
5: **while** $running$ **do**
6:    **for** $v \in V$ **do**
7:       $m_{N0} \Leftarrow$ calculateNodeCriteria(V)
8:       $m_N \Leftarrow$ findLowestNodeCriteria(V)
9:       **if** $m_N < m_{N0}$ **then**
10:          moveNode(v)
11:       **end if**
12:    **end for**
13:    $P \Leftarrow$ clusterOverlengthEdges$(V, E)\cup$clusterBends$(V, E)\cup$clusterPartitions(V, E)

14:    **for** $p \in P$ **do**
15:       $m_{N0} \Leftarrow$ calculateNodeCriteria(V)
16:       $m_N \Leftarrow$ findLowestNodeCriteria(V)
17:       **if** $m_N < m_{N0}$ **then**
18:          moveCluster(p)
19:       **end if**
20:    **end for**
21:    **for** $l \in L$ **do**
22:       $m_{L0} \Leftarrow$ calculateLabelCriteria(L)
23:       $m_L \Leftarrow$ findLowestLabelCriteria(L)
24:       **if** $m_L < m_{L0}$ **then**
25:          moveLabel(l)
26:       **end if**
27:    **end for**
28:    $m_T \Leftarrow$ calculateNodeCriteria(V) + calculateLabelCriteria(L)
29:    **if** $m_T \not< m_{T0}$ **then**
30:       $running \Leftarrow$ **false**
31:    **else**
32:       $m_{T0} \Leftarrow m_T$
33:    **end if**
34: **end while**

of iterations rather than relying on this specific termination condition as the improvements in later iterations become very small and only make a subtle effect on the overall aesthetic appearance of the graph.

3.4   Initial Embedding

An important consideration which greatly affects the finished maps is that of the initial layout of the map. There are two main 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. Therefore, it 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.

3.4.1   Embedding on the Grid

The first stage of the metro map layout method is to ensure that all nodes are centred on grid intersections (line 29 of Algorithm 3.2). We will be starting with a geographic layout (or a close alternative using a topologically-correct sketch of the map [BEKW02]) so some process of *snapping* nodes to the nearest grid intersections is needed.

Finding 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. In other words, each node is snapped to the nearest grid intersection that is not already occupied by another node.

Nodes are processed in order of the distance that they are from a grid intersection such that those nodes closest to a grid intersection are moved first and those nodes furthest from a grid intersection are moved last. In the case where two ore more nodes are an equal distance from the same grid intersection, an arbitrary choice is made as to which one to move first. 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 cases, 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. 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 nearest grid intersection first (as it is closest) and B' is positioned on the next nearest grid intersection.

3.5   Node Movement

Movement of nodes depends on 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,** c_{N1}**.** The angular resolution of incident edges at each node is maximised. See Section 3.6.1.

- **Edge Length Criterion,** c_{N2}**.** The edge lengths across the whole map should be approximately equal. See Section 3.6.2.

- **Balanced Edge Length Criterion,** c_{N3}**.** The length of edges incident to a particular node should be approximately equal. See Section 3.6.3.
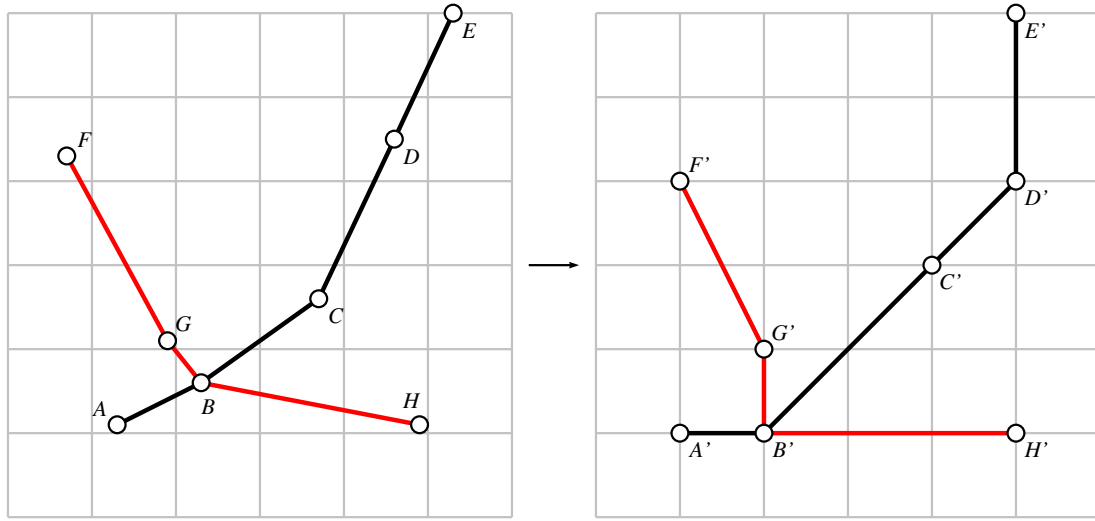
Figure 3.3: Moving nodes to grid intersections. The left-hand graph is the initial layout; the right-hand graph is the same graph but with nodes moved to grid intersections. In the cases where there is already a node occupying a grid intersection, the next nearest grid intersection is used—this is the case with nodes B and G in this example.

- **Edge Crossings Criterion, c_{N4}.** The number of unnecessary edge crossings should be reduced. See Section 3.6.4.

- **Line Straightness Criterion, c_{N5}.** Edges that form part of a line should, where possible, be collinear either side of each node that the line passes through. See Section 3.6.5.

- **Octilinearity Criterion, c_{N6}.** Each edge should be drawn horizontally, vertically or diagonally at 45°. See Section 3.6.6.

The purpose of these criteria is that they should be minimised by the hill climbing multicriteria optimisation method. Each criterion forms part of the objective function that the optimiser uses to determine the aesthetic quality of the graph.

Our basis for the selection of criteria comes from existing research that evaluates aesthetic criteria in relation to graph drawing, as discussed in Section 2.4.3. Not all common aesthetic criteria are appropriate (for example, symmetry does not play a part in many metro maps) and some criteria need to be modified (such as the octilinearity criterion being a relaxation of orthogonality) to fit the metro map metaphor. The line straightness and balanced edge length criteria were included to meet specific requirements of the metro map metaphor or in response to particular local minima in the search space.

As well as these six criteria, we have implemented four node movement rules which are strictly enforced during the layout process. We used rules in addition to criteria as there are some cases in which the criteria are unable to guard against introducing
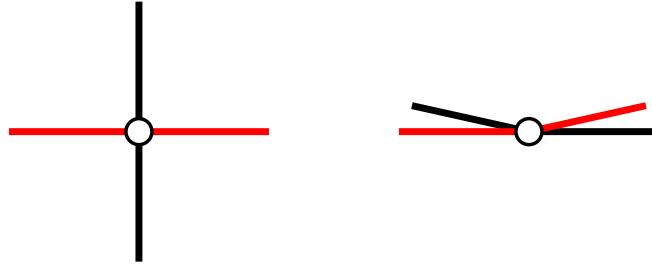
Figure 3.4: Examples of optimal angular resolution (left) and poor angular resolution (right).

undesirable node movements that would otherwise be difficult (or inefficient) to handle with additional criteria. The four node movement rules are:

- **Bounding Area Restriction Rule.** Restrict the movement of nodes to be within a certain bounding area. See Section 3.7.1.

- **Geographic Relationships Rule.** Enforce the geographic relationships between pairs of nodes (for example, that one node should be north of another node). See Section 3.7.2.

- **Occlusions Rule.** Avoid the introduction of occlusions of other edges and nodes. See Section 3.7.3.

- **Edge Ordering Rule.** Preserve the ordering of edges incident to a node. See Section 3.7.4.

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. The section for each criterion also relates how that criterion applies to the aims for our metro map layout method as set out in Section 3.1.

3.6   Node Movement Criteria

3.6.1   Angular Resolution Criterion, c_{N1}

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 between two pairs of edges but very large angles between the other two pairs of edges.
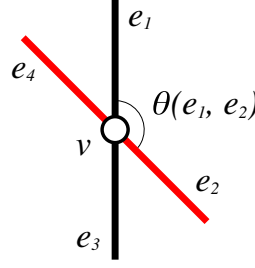
Figure 3.5: Calculating the angular resolution criterion.
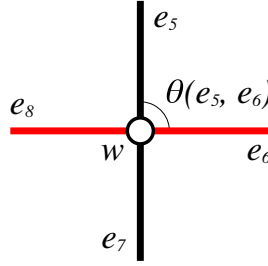
Figure 3.6: Calculating the angular resolution criterion.

This criterion does not map directly to any of the aims described in Section 3.1 but is related to the aim that metro lines should appear to pass straight through stations.

The angular resolution criterion, c_{N1}, ensures that there is as large an angle as possible between adjacent edges incident to a node. The criterion is calculated using

$$c_{N1} = \sum_{v \in V} \sum_{\{e_1,e_2\} \in E_v} \left| \frac{2\pi}{\rho(v)} - \theta(e_1, e_2) \right| \tag{3.1}$$

where $\rho(v)$ is the degree of the node v and $\theta(e_1, e_2)$ is the angle in radians 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 $\frac{\pi}{2}$ (90°). The angle θ between edges e_1 and e_2 is $\frac{3\pi}{4}$ (135°), so for that particular pair of edges, the absolute difference between the actual and ideal angles is $\left| \frac{2\pi}{4} - \frac{3\pi}{4} \right| = \frac{\pi}{4}$ (45°). For the other pairs of edges, the difference is also $\frac{\pi}{4}$, so for the area around v, $c_{N1} = \frac{4\pi}{4} = \pi$ (180°).

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 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_{N1} for the area around w is 0.

3.6.2   Edge Length Criterion, c_{N2}

A common feature of metro maps is that stations should be spaced evenly along lines and that the spacing 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 2.2.2). The regular spacing of stations and the implicit use of scale generalisation are covered by two of the aims for our method set out in Section 3.1.

The edge length criterion, c_{N2}, addresses the need to have regular spacings between stations. It is based on some preferred multiple, l, of the grid spacing g. The purpose of the criterion is to penalise edges that are longer than or shorter than lg. The edge length criterion is calculated by

$$c_{N2} = \sum_{e \in E} \left| \frac{|e|}{lg} - 1 \right| \tag{3.2}$$

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_{N2} for that edge will be greater than zero.

To illustrate how the edge length criterion works, consider the example shown in Figure 3.7. There are three edges in this small graph, AB, BC and BD, with lengths $2g$, $\sqrt{18}g$ and $6g$ respectively. If we assume that $g = 1$, we can see how c_{N2} 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 initially be longer than lg. As l tends towards ∞ the value for c_{N2} tends towards the number of edges in the graph.

While this works fine for horizontal or vertical edges, diagonal edges pose an interesting problem, as may have been noticed for edge BC in Figure 3.7. Where diagonal edges are concerned, it is very likely that a position on the grid where the edge length criterion can be zero simply does not exist. Figure 3.8 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 $g\sqrt{2}$. 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.
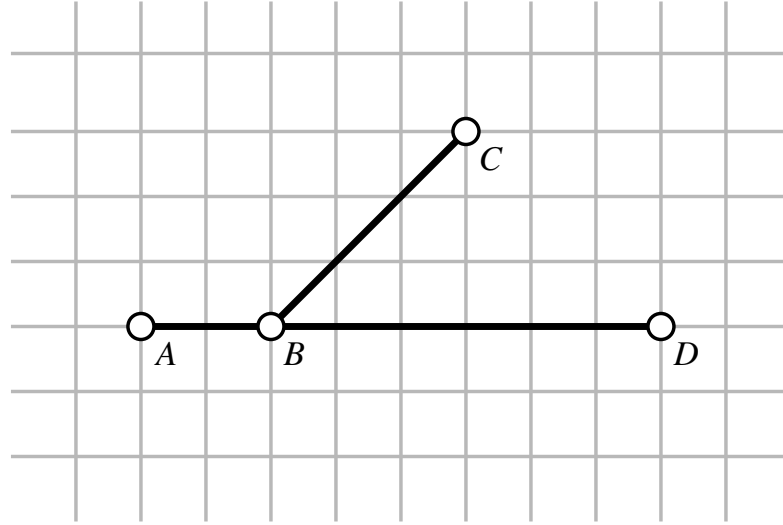
Figure 3.7: Example used to illustrate the edge length criterion.

Table 3.1: Examples of how the edge length criterion varies with different values of l (l is the preferred multiple of grid spacings for the length of an edge). The edges AB, BC and BD are shown on Figure 3.7 and have lengths 2, $\sqrt{18}$ and 6 respectively. In calculating c_{N2} for each of these edge lengths, we have assumed that $g = 1$. c_{N2}^{XY} means the calculation of c_{N2} for just the part of the graph between nodes X and Y.

l	c_{N2}^{AB}	c_{N2}^{BC}	c_{N2}^{BD}	c_{N2}
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
7	0.72	0.39	0.14	1.24
8	0.75	0.47	0.25	1.47
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
∞	1	1	1	3
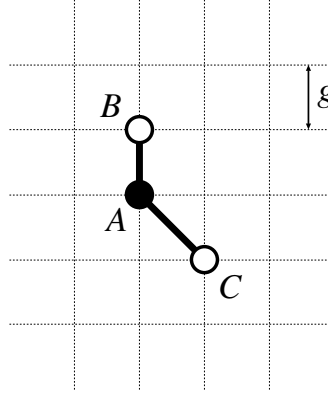
Figure 3.8: Problem with diagonal edge lengths and the edge length criterion.
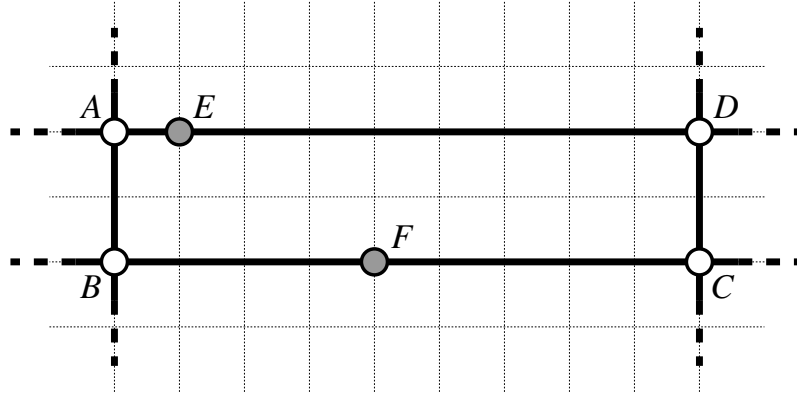
Figure 3.9: Balanced edge lengths.

3.6.3   Balanced Edge Length Criterion, c_{N3}

One of the characteristics of metro maps is that there are many nodes with two in-cident 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 in Section 3.6.2 does not handle this particular feature of the maps effectively and can result in a very common local minimum in the search space. The balanced edge length criterion overcomes this particular limitation.

Figure 3.9 shows an example whereby there are two nodes, E and F, 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

$$\left|\frac{|e_{AE}|}{lg} - 1\right| + \left|\frac{|e_{ED}|}{lg} - 1\right| = \left|\frac{1}{1} - 1\right| + \left|\frac{8}{1} - 1\right| = 7 \tag{3.3}$$

and for node F

$$\left|\frac{|e_{BF}|}{lg} - 1\right| + \left|\frac{|e_{FC}|}{lg} - 1\right| = \left|\frac{4}{1} - 1\right| + \left|\frac{5}{1} - 1\right| = 7. \qquad (3.4)$$

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_{N3}, is simply the sum of the absolute difference between the lengths of the two incident edges of every degree-two node in the graph, i.e.

$$c_{N3} = \sum_{v \in V, \rho(v)=2} ||e_1| - |e_2|| \qquad (3.5)$$

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

Using the nodes E and F from Figure 3.9 as an example, the balanced edge length criterion for E is $|1 - 8| = 7$ and for F is $|4 - 5| = 1$.

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 perfectly.

3.6.4 Edge Crossings Criterion, c_{N4}

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 adversely affect the readability of the map [PCJ95].

Calculation of the edge crossings criterion, c_{N4}, is performed by checking each edge to see whether any other edges cross it: c_{N4} 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 adjacent nodes. In this case, even though several edges are actually being crossed, only a single crossing is counted for the purposes of this criterion. Counting the number of edge crossings in a graph involves checking pair of edges to see if they intersect. With most metro maps having no more than a few hundred edges, the time taken to compute the number of edge crossings is fairly short, so we did not explore any options for improving this.

Two efficiency enhancements can be made for this criterion:
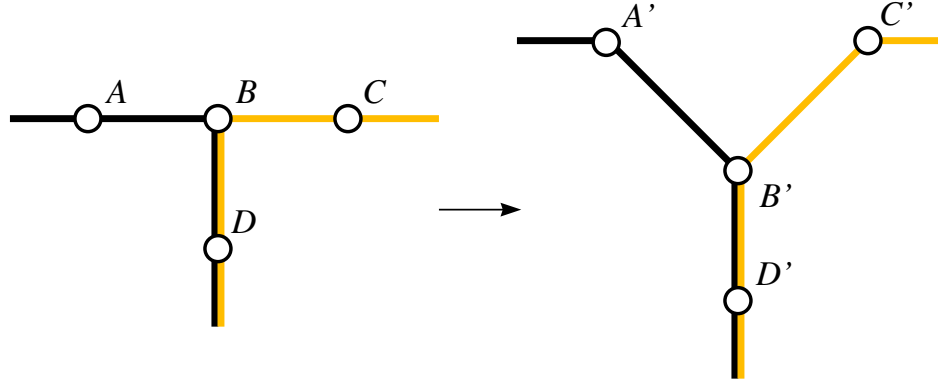
Figure 3.10: Examples of poor line straightness (left) and improved line straightness (right).

- 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.

- 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.)

3.6.5   Line Straightness Criterion, c_{N5}

One of the important features of metro maps is that metro 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 (i.e., 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 graph on the left can have the line straightness improved by moving nodes. This criterion addresses the aim of having metro lines passing through stations in as straight a line as possible (see Section 3.1).

To ensure this type of improvement, we introduce the line straightness criterion, c_{N5}, calcuated by

$$c_{N5} = \sum_{(v \in V)} \left(\sum_{e_1, e_2 \in E} \theta(e_1, e_2) \right) \tag{3.6}$$

where $\theta(e_1, e_2)$ is the smallest 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 the end points of e_1 and e_2 are collinear, $\theta(e_1, e_2) = 0°$; 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 $\theta(e_1, e_2)$
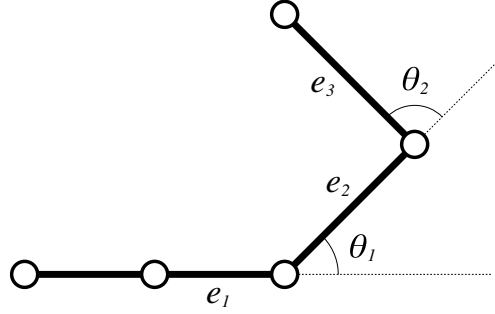
Figure 3.11: Example of the calculation of the line straightness criterion.

is large more than turns where $\theta\,(e_1, e_2)$ is small or zero. To avoid any ambiguity in what angle is being measured, the metro line must be followed from one end to the other, being consistent with the point at which the angles are measured.

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 calcuate the line straightness criterion for 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_{N6}

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. This criterion specifically addresses the first of our aims for drawing metro lines as described in Section 3.1.

The octilinearity criterion, c_{N6}, has the effect of penalising edges that are not some multiple of $45°$ and is calculated using

$$c_{N6} = \sum_{\{u,v\} \in E} \left| \sin 4 \left(\tan^{-1} \frac{|y(u) - y(v)|}{|x(u) - x(v)|} \right) \right| \tag{3.7}$$

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_{N6}^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 evaluate to the same value because they are both $26.57°$ away from the nearest multiple of $45°$[1].

---

[1] This is somewhat unintuitive visually but can be checked with simple trigonometry.
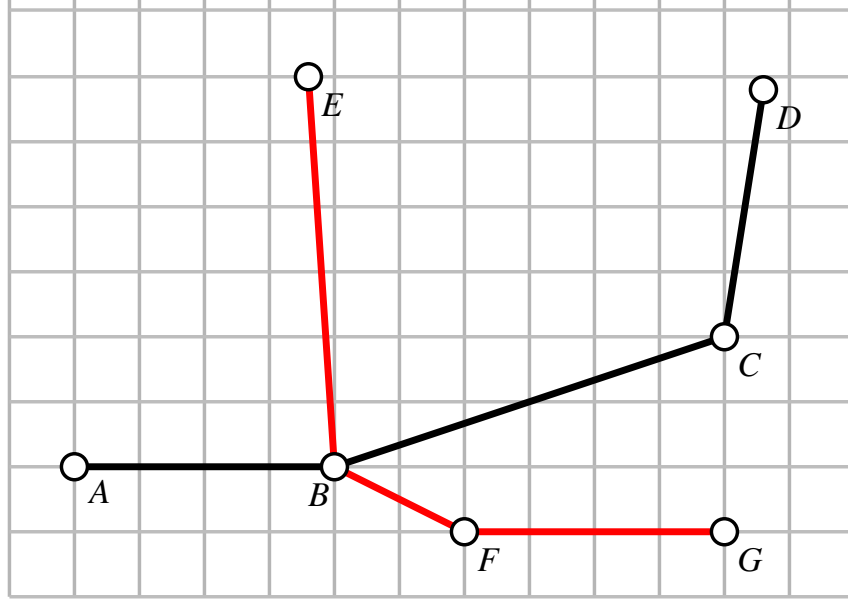
Figure 3.12: Example of the calculation of the octilinearity criterion.

Table 3.2: Examples of octilinearity criterion calculations with reference to Figure 3.12. Edges BC and BF evaluate to the same value because they are both $26.57°$ away from the nearest multiple of $45°$.

Edge, $e = \{u, v\}$	c_{N6}^e
$\{A, B\}$	$\left\|\sin 4 \left(\tan^{-1} \frac{0}{4}\right)\right\| = 0$
$\{B, C\}$	$\left\|\sin 4 \left(\tan^{-1} \frac{2}{6}\right)\right\| = 0.96$
$\{C, D\}$	$\left\|\sin 4 \left(\tan^{-1} \frac{3.8}{0.6}\right)\right\| = 0.586$
$\{B, E\}$	$\left\|\sin 4 \left(\tan^{-1} \frac{6}{0.6}\right)\right\| = 0.388$
$\{B, F\}$	$\left\|\sin 4 \left(\tan^{-1} \frac{1}{2}\right)\right\| = 0.96$
$\{F, G\}$	$\left\|\sin 4 \left(\tan^{-1} \frac{0}{4}\right)\right\| = 0$

3.6.7   Node Movement Criteria Weightings

Each criterion was weighted by an independent weighting. The intention of these weightings is twofold. First, 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 (normalised). This is important to ensure that one criterion does not completely overwhelm the other criteria. Second, by 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. Conversely, a lower weighting can be used to reduce the effect of a particular criterion. If an application of the method does not require a particular criterion, the weighting can be set to zero. This allows flexibility in deciding which criteria should contribute to the characteristics of the map for various applications.

The sum of the weighted criteria for node movement, m_N, is given by

$$
\begin{aligned}
m_N &= w_{N1}c_{N1} + w_{N2}c_{N2} + w_{N3}c_{N3} + w_{N4}c_{N4} + \\
&\quad w_{N5}c_{N5} + w_{N6}c_{N6} \\
&= \sum_{i=1}^{6} w_{Ni}c_{Ni}.
\end{aligned} \tag{3.8}
$$

The values for w_{Ni} can be modified by the user depending on the characteristics of the particular metro map being drawn. The weightings that we used were determined through a process of trial and error. This process first involved setting the weightings such that the weighted values are effectively normalised (to cancel out differences in magnitudes) and then using particular examples to determine how each weighting should be modified so that it has the desired effect. This process is discussed with examples in more detail in Section 5.3.

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:

$$
\begin{aligned}
m_N &= 19.2962 + 21.0240 + 25.1063 + 40.0000 + 23.5620 + 21.4848 \\
&= 150.4733.
\end{aligned} \tag{3.9}
$$

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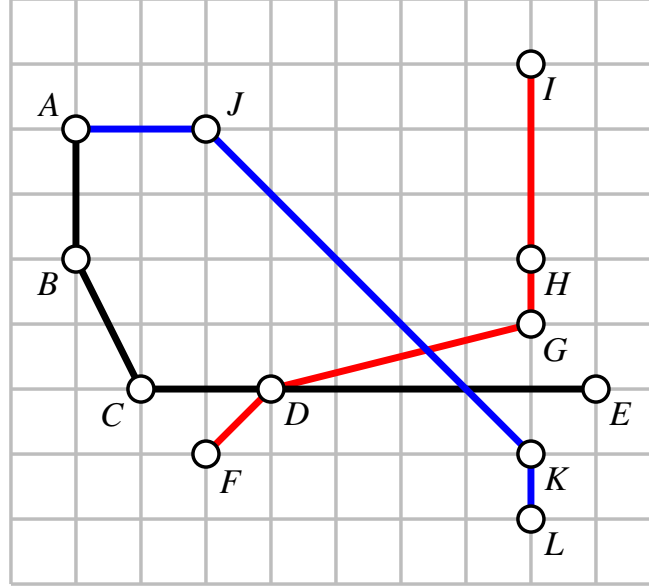
Figure 3.13: Simple example of node movement criteria for a complete graph. The values for each criterion for this graph are shown in Table 3.3.

Table 3.3: Typical node movement criteria weightings.

Criterion	c_{Ni}	Weighting, w_{Ni}	$w_{Ni}c_{Ni}$ for graph in Figure 3.13
c_{N1}	19.2962	1.0000	19.2962
c_{N2}	7.0080	3.0000	21.0240
c_{N3}	16.7375	1.5000	25.1063
c_{N4}	2.0000	20.0000	40.0000
c_{N5}	3.9270	6.0000	23.5620
c_{N6}	1.7904	12.0000	21.4848
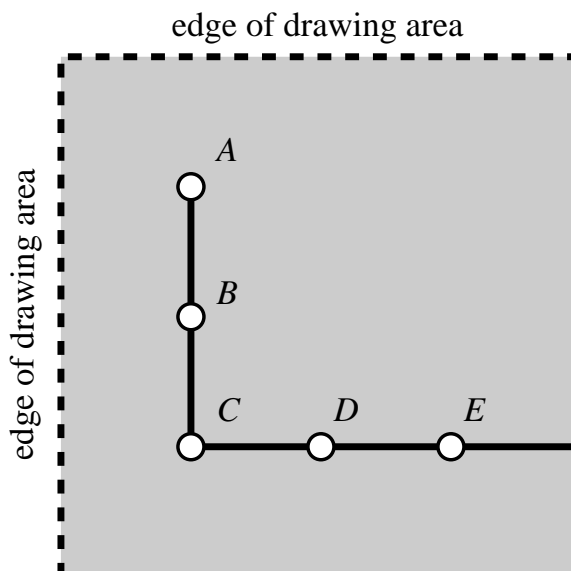
Figure 3.14: Example of the restriction of node movements to stay within the drawing area boundary. Nodes C, D and E form part of a longer straight line and ideally nodes A and B would move to be parallel to them, but the edge of the drawing area restricts this movement.

movement of a node to a new location if any of the rules are broken. Specific examples justifying the reason for each rule that we used are detailed later in this Section.

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 drawing maps on a finite drawing area, it is sometimes a reqirement that any node movement does not cause any node to move outside the drawing area. This is not always necessary—in practice, it should be possible to ensure that the drawing area is always large enough to contain the graph and any potential growth of the overall dimensions of the graph.

To illustrate a case where this rule applies, 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 collinear with them. However, moving nodes A and B in this way would put them either wholly or partially outside the drawing area.

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

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 would not be 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 so that they are 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.

3.7.3 Handling Node and Edge Occlusions

In Section 3.6.4 we introduced a criterion which penalises edge crossings in the graph. While this criterion is useful for removing unwanted edge crossings, it is usually the case that we do not want new edge crossings to be introduced. We also want to make
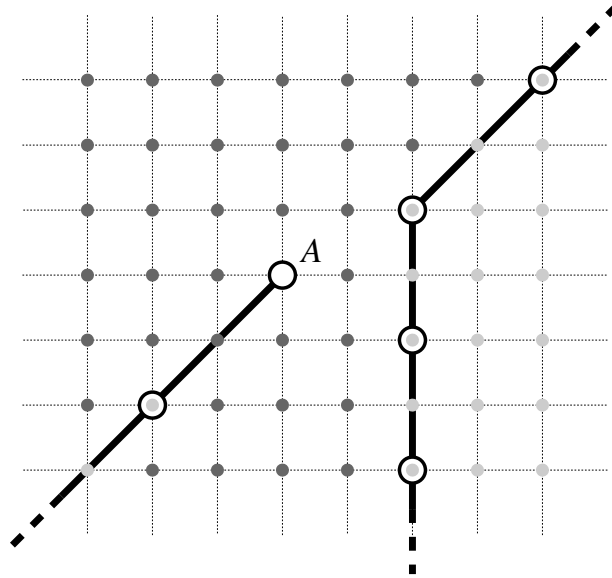
Figure 3.16: How node and edge occlusions can restrict the search space. Light grey dots show positions where node A cannot be moved to because either a node or edge occlusion would result. Dark grey dots show valid positions where no occlusions would happen.

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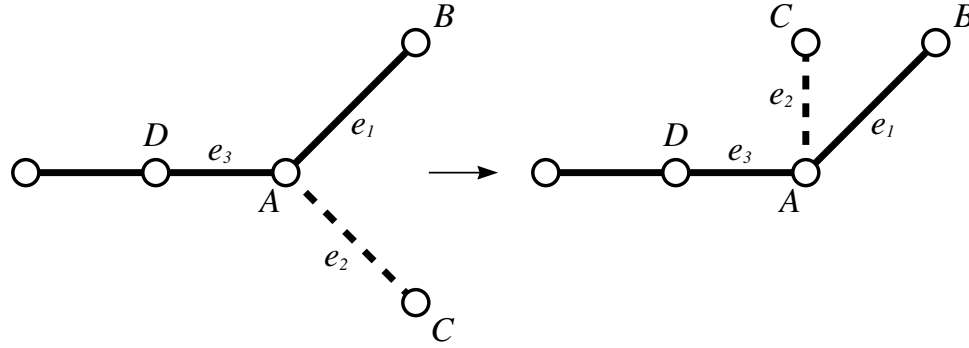
Figure 3.17: Preservation of edge ordering. Without preserving the ordering of edges, node C would be able to move as shown, changing the topography of the map.

meaning that the node movement criteria need only be calculated for the remaining 36 locations.

3.7.4 Preservation of Edge Ordering

The geographic relationships rule (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.

This rule becomes redundant when considering graphs that are not constrained by their topology. For example, the topology a graph representing an abstract, non-geographic network is not relevant. In these cases, this rule can be ignored or the graph can have the topology defined by providing an initial embedding.

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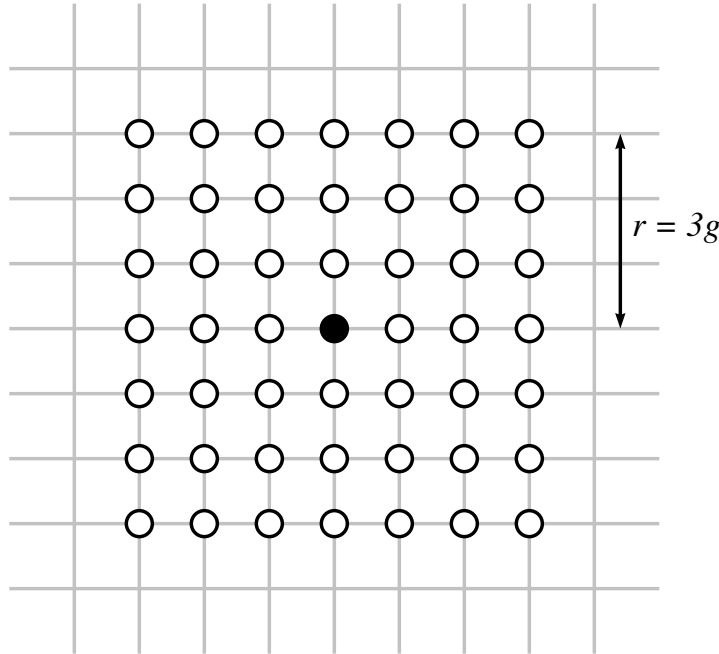
Figure 3.18: The node movement radius. The black node in the centre is being moved and $r = 3g$: all the potential new locations in this radius are shown with white nodes.

a position to move a node to: the sum of the weighted criteria, m_N (Equation 3.8); 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 maximum distance to move the node is reduced with each iteration using cooling (see Section 3.8.1); 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. For example, if nodes in the densest part of the graph are separated by approximately $3g$, a value of $r = 10$ would be appropriate if it was desired to consider nodes moving up to three times the approximate distance between nodes. 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.

When considering potential movements for a node, the initial value of m_0 is first calculated using Equation 3.8. 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
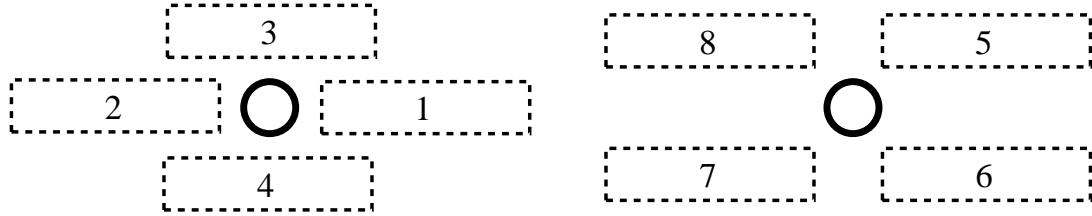
Figure 3.19: Search space for labelling the metro map.

locations, T, is remembered for each potential movement where the new value of m is less than the initial value of m_0. When there is a single location in T with the smallest value of m, the node is moved to that location. In the case of no potential movements being discovered, the node is left in its original location. If there is more than one location with the same smallest value of m, we select the first improved location that was found.

3.8.1   Cooling

Cooling is a process borrowed from physical systems where the rate of change is initially high but decreases over time. Annealing is one such process and was described in Section 2.4.6. The rate of cooling is described as the *cooling schedule* and may be linear, logarithmic, irregular, etc. In terms of our method, cooling means that the radius within which nodes can move (r) is decreased at each successive iteration.

We used a linear cooling schedule such that the value of r is reduced to one by the last iteration. For example, if at the first iteration $r = 10$ and we are running five iterations, r will be reduced by two at each iteration so that by the last iteration $r = 2$. We experimented with a number of different cooling schedules such as a logarithmic schedule or an irregular schedule, but the difference between results using each different schedule was negligible. It therefore made sense to use the least computationally expensive linear cooling schedule.

3.9   Labelling

Labelling is an integral part of metro maps and hence it should form an integral part of our multicriteria optimisation method. To this end, a number of criteria are introduced for label placement. Each of these criteria address the aims for labelling as described in Section 3.1.

In order to reduce the number of potential locations for labels and to allow a preference for one position over another, we limit the number of positions using a labelling space. Figure 3.19 shows our chosen labelling space, which allows eight different label positions.
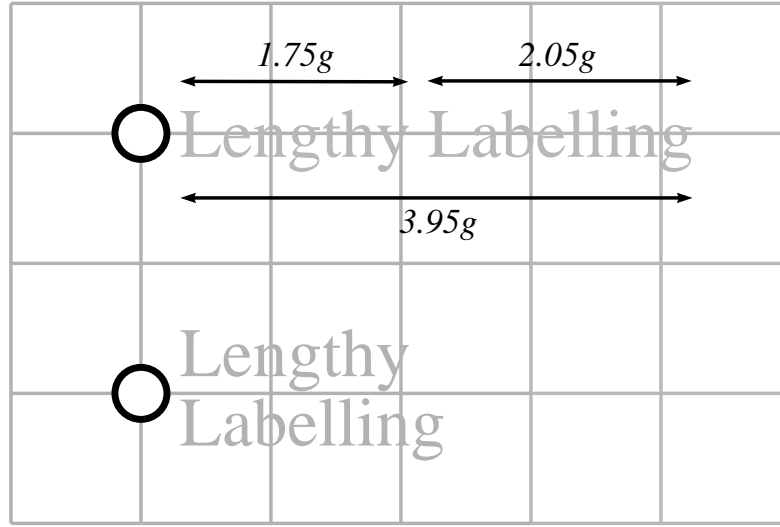
Figure 3.20: Splitting a long, multi-word label of multiple lines. The word "Lengthy" is about $1.75g$ and "Labelling" is about $2.05g$ causing the label to be about $3.95g$ in total. The bottom label shows how the label "Lengthy Labelling" can be split over two lines.

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. We decide whether to split a long label over multiple lines by causing a line break in the label if the length of the label exceeded $0.75lg$. This value was chosen because the distance between nodes tends towards lg: it would be better, if labels were to fit between other 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 3.20 shows an example where a long label with two words can be split over two lines.

3.9.1 Labelling Criteria

We have implemented a total of seven labelling criteria:

- **Label Occlusion Criteria,** c_{L1}, c_{L2}, c_{L3}. Three criteria that take into account the number of nodes, edges and other labels that intersect/occlude labels. See Section 3.9.2.

- **Label Position Criterion,** c_{L4}. Places a preference on label positions in the labelling space by putting a value on each position. See Section 3.9.3.

- **Label Position Consistency Criterion,** c_{L5}. Gives preference to labels along a line in the map that consistently appear on the same side of the line. See Section 3.9.4.

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 (c_{L1})	4	c, e, f, g
Label-node (c_{L2})	1	e
Label-edge (c_{L3})	7	a, b (twice), e (twice), f, g

- **Node Proximity Criterion, c_{L6}.** Considers labels as they come into close proximity to unrelated nodes with the intention of discouraging labels from being positioned too close to other unrelated nodes. See Section 3.9.5.

- **Perpendicular Tick Criterion, c_{L7}.** Attempts to ensure that the tick (and therefore the position of the label) for a particular node is perpendicular to the line. See Section 3.9.6.

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, c_{L1}, c_{L2}, c_{L3}

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. This tends to occur in areas of the graph with the greatest density of nodes and edges. In these cases, the label may not fit in any position around a node such that it does not occlude an edge without first significantly altering the layout of that part of the graph. For dense areas of the graph, it may not be possible to find any improvements to the position of nodes to resolve any label occlusions.

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 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 labels exhibit occlusions. Notice that, although only there are only two examples of occluding labels, the count for the label-label occlusion is four.
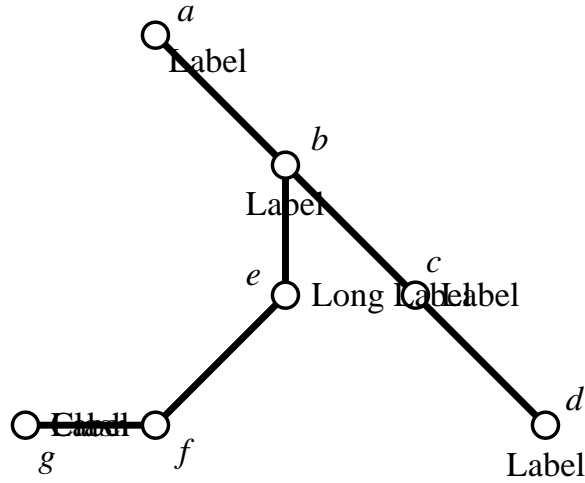
Figure 3.21: Illustration of the label occlusion criteria.

3.9.3 Label Position Criterion, c_{L4}

As discussed in Section 3.9, 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 is not necessarily an unreasonable limitation as many existing metro maps also restrict the 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 makes 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 each different position in the labelling space is assigned a value relating to the preference for that position. Table 3.5 shows the set of values for each position in the labelling space. The label position criterion is then defined as the sum of the position values for each label in the graph.

By setting the values, 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

Table 3.5: Label position values. The positions refer to the positions shown in Figure 3.19.

	position	value
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

node (position 1) has a value of 1.0 and is therefore the most preferred location, while a label which is oriented to the north-west of a node has value 1.8 and is the least preferred location. The values chosen were determined through trial and error 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 the values we have assigned to each position. Where there was not a particular preference, the values were chosen so that they were similar or even identical (for example, there is no difference between the values for north and south labels[2]). This then allows other labelling criteria to have a greater effect on the label position.

3.9.4 Label Position Consistency Criterion, c_{L5}

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 switch 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 equal to 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.

---

[2]This is because we sometimes want labels to alternate either side of a horizontal line where there is not enough horizontal space between adjacent labels to fit each label. This is discussed with examples in Section 3.9.
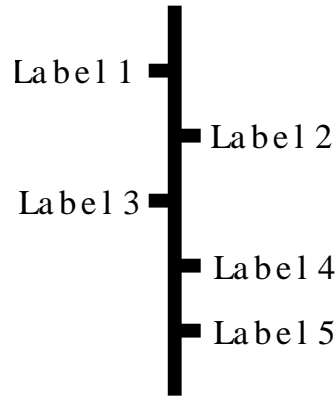
Figure 3.22: Label position consistency.
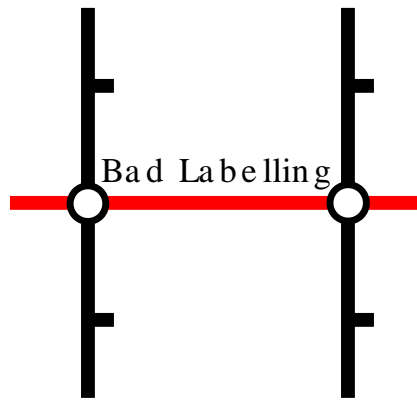
Figure 3.23: An example of ambiguous labelling.

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).

This criterion applies to both horizontal and vertical lines. However, for horizontal lines, the width of each label may mean that they cannot all fit along one side of the line. In this case, there may be no option but to have each label swap from one side of the line to the other. As such, the weighting for this criterion must be determined so that it doe not overwhelm the label-label occlusion criterion.

3.9.5   Node Proximity Criterion, c_{L6}

The node proximity criterion addresses the problem whereby a label for an individual node could be 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. To counter this problem, we introduce a node proximity criterion which works by forcing a label away from a node to which it does not belong.
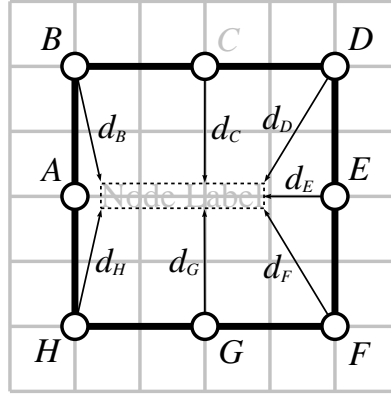
Figure 3.24: Calculation of the node proximity criterion.

The node proximity criterion, c_{L6}, is given by

$$c_{L6} = \sum_{k \in L} \sum_{n \in V, k_n \neq k} \frac{1}{d(k,n)^2} \qquad (3.10)$$

where $d(k,n)$ is a function giving the distance from the closest point on the bounding box of label k to node n. Notice that we are interested in $n \in V, k_n \neq k$, that is all nodes in the graph except the one for which the label (k_n) is the label we are considering (k). In other words, we do not take into account the distance between a label and node that that label belongs to.

In practice, because most nodes in the graph will be some way from the label in question, they will contribute very little to c_{L6}. We can therefore approximate the contribution of nodes with $d(k,n) > x$ to zero. We use a value of 1 for x.

The example in Figure 3.24 illustrates how the criterion is calculated for a particular node. In particular, it shows how the distance function $d(k,n)$ relates to the bounding box for the label.

3.9.6   Perpendicular Tick Criterion, c_{L7}

One of the disadvantages of using a combination of criteria to decide on the position of a label is that positions may be chosen that do not 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 3.25 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
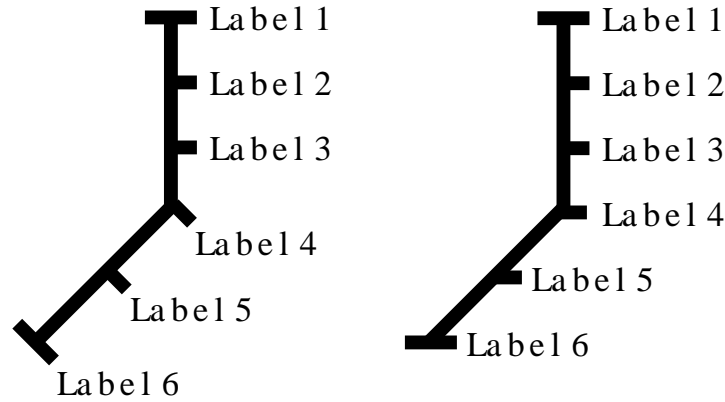
Figure 3.25: Examples of perpendicular tick labels (left) and non-perpendicular tick labels (right)
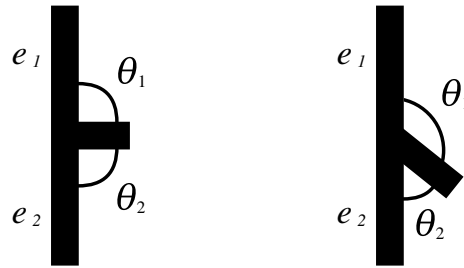
Figure 3.26: Calculating the perpendicular tick criterion

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.

The example shown in Figure 3.25 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 would 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.

Calculation of the perpendicular tick criterion is fairly straightforward. Figure 3.26 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

$$c_{L7} = \sum_{v \in V} |\theta_1 - \theta_2| \,. \tag{3.12}$$

In the examples in Figure 3.26, the value of c_{L7} 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_{L7} is $135° - 45° = 90°$. This shows that the label is not perpendicular to either e_1 or e_2.

3.9.7   Labelling Criteria Weightings

Each criterion is independently weighted depending on the importance associated with that criterion. The values for weightings were determined through trial and error with various examples in a similar manner to the way that we determined node movement criteria weightings (Section 3.6.7). Two weightings will be similar or the same for criteria that have similar magnitudes when unweighted and when the two criteria do not depend on each other (for example, the label position and label position consistency criteria are interdependent, while the three occlusion criteria are not dependent). More detail on specific examples that guided our choice of weightings are given in Section 5.3.

The sum of the weighted criteria for labelling, m_L, is given by

$$
\begin{aligned}
m_L &= w_{L1}c_{L1} + w_{L2}c_{L2} + w_{L3}c_{L3} + w_{L4}c_{L4} + \\
&\quad\ w_{L5}c_{L5} + w_{L6}c_{L6} + w_{L7}c_{L7} \\
&= \sum_{i=1}^{7} w_{Li}c_{Li}.
\end{aligned} \tag{3.13}
$$

As with the node movement critera weightings, the values for the w_{Li} can be modified by the user depending on the characteristics of the particular metro map being drawn.

Table 3.6 shows a typical set of weightings for each of the labelling criteria and the values as applied to the example in Figure 3.27. This example has several labels in sub-optimal positions (such as "Label B" and "Label E") in order to illustrate the calculation of the labelling crtiteria. 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 optimisation algorithms that local minima exist in the search space, so no one set of weightings would be applicable in every single case.
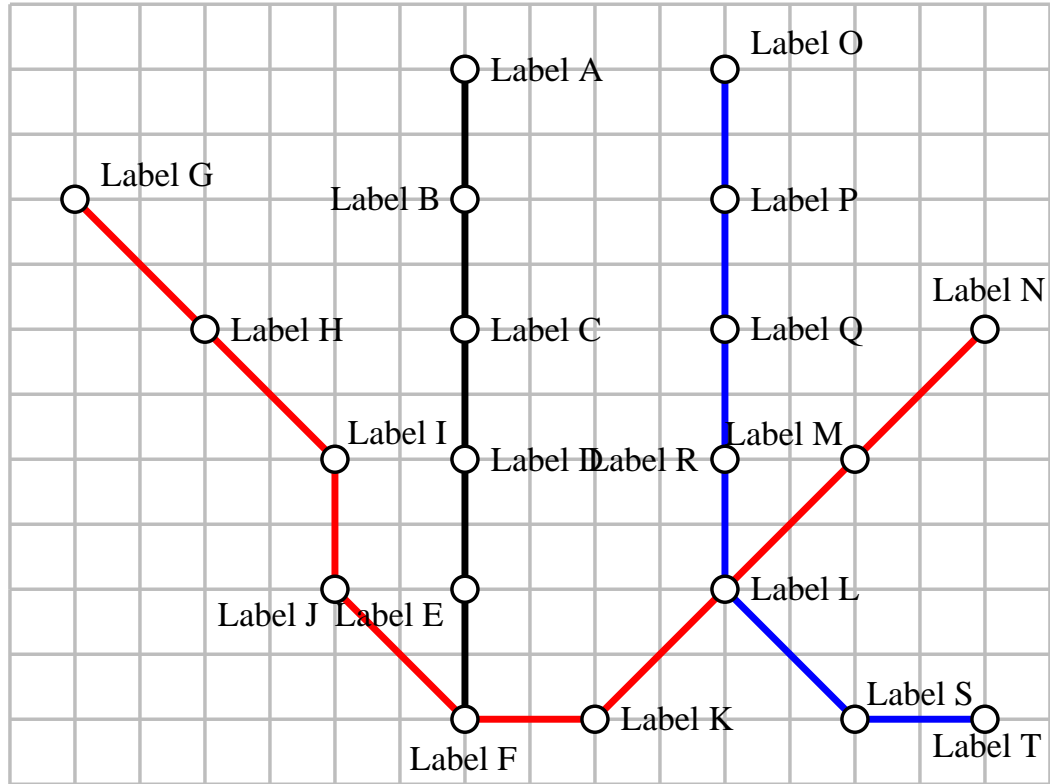
Figure 3.27: Simple example of label criteria for a complete graph. The labels are shown in a sub-optimal position to illustrate the calculation of the criteria. The values for each criterion for this graph are shown in Table 3.6.

Table 3.6: Typical label position criteria weightings.

Criterion	c_{Li}	Weighting w_{Li}	$w_{Li}c_{Li}$ for graph in Figure 3.27
c_{L1}	2.0000	100.0000	200.0000
c_{L2}	0.0000	100.0000	0.0000
c_{L3}	2.0000	100.0000	200.0000
c_{L4}	24.0000	3.0000	72.0000
c_{L5}	10.0000	12.0000	120.0000
c_{L6}	70.1923	1.0000	70.1923
c_{L7}	7.8540	10.0000	78.5400

The total of the weighted criteria in Figure 3.13 is

$$
\begin{aligned}
m_L &= 200.0000 + 0.0000 + 200.0000 + 72.0000 + 120.0000 + 70.1923 + 78.5400 \\
&= 740.7323. \tag{3.14}
\end{aligned}
$$

3.10 Moving Labels

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.

3.11 Combining Node Movement and Labelling Criteria

The total of the weighted node movement criteria, m_N, and labelling criteria, m_L, is combined to produce a total sum for the entire graph. This total, m_T, is given by

$$
\begin{aligned}
m_T &= m_N + m_L \\
&= \sum_{i=1}^{6} w_{Ni} c_{Ni} + \sum_{j=1}^{7} w_{Lj} c_{Lj}. \tag{3.15}
\end{aligned}
$$

This is the value that is used when assessing the aesthetic quality for the examples later in this thesis. Notice that we do not apply an additional weighting to m_N or m_L—this was not necessary as the weightings for each individual criterion were determined on the basis of the node movement and labelling criteria being summed together. The determination of individual criterion weightings is discussed in detail with examples in Section 5.3.

3.12 Conclusion

This chapter started by setting out the aims of our method for the automatic layout of metro maps. The process of iteratively improving an initial embedding of a metro map was described using multicriteria optimisation combining several aesthetic criteria for nodes, edges and labels.

In the next chapter, we show how clustering and partitioning the graph can help resolve a number of problems involving local minima by allowing more than one node to be moved simultaneously.

Chapter 4

Clustering and Partitioning

Chapter 3 introduced a method for laying out metro maps using multicriteria optimisation. In this method, just individual nodes at a time were considered for moving. Owing to the nature of the optimisation method, it is likely that local minima are found rather than a more optimal solution. Many of the most common local minima that we encountered were due to long edges and bends in lines. Rather than introduce more criteria into the single node movement phase, we decided to change the way that the graph was modified by allowing several clustered nodes to move together. Specific examples of the local minima that motivated each clustering method are described later in this chapter.

Clustering is the process of finding subsets (*clusters*) of nodes in a graph based on some rules that define the boundaries of a cluster. *Partitioning* is a similar process in that clusters (or *partitions*) of nodes are discovered. Partitioning is different to clustering in the way that the partitions are defined (by splitting the graph into two partitions rather than selecting a subset of nodes to treat as a cluster)—the result of both partitioning and clustering is a set of clusters containing nodes that can be moved together.

We introduced three methods for clustering:

- Clustering based on overlength (or underlength) edges (see Section 4.1).

- Clustering based on bends in lines (see Section 4.2).

- Clustering based on partitioning the graph into two parts that can be moved closer together (see Section 4.3).

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. The relative position of nodes forming a cluster is maintained while the cluster is being moved. Individual nodes can therefore be treated as clusters containing just one node.
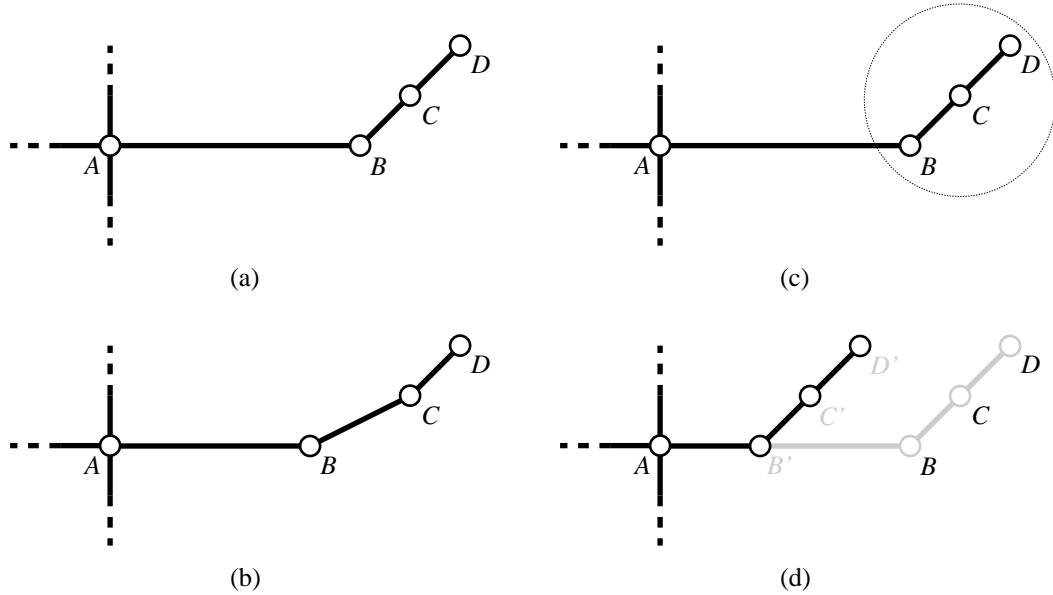
Figure 4.1: Clustering overlength edges. The edge AB is too long (a), but it is not possible to reduce the length of this edge (b) without moving nodes B, C and D (c) at the same time (d).

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 introduces 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 criterion 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 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 move all three nodes B, C and D to their optimal positions as shown in (d). It is also undesirable to have to increase criteria weightings significantly 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 partitions would require fewer criteria calculations when potential locations for the nodes are being tested.

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 breadth-first search from node C will find the
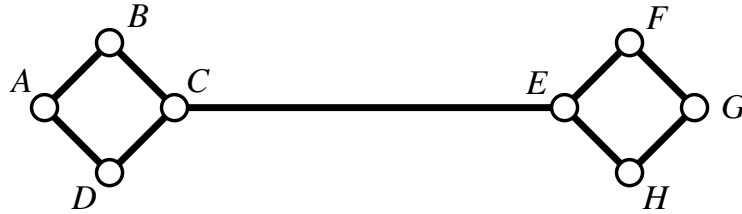
Figure 4.2: Example of clustering with a single overlength edge. An attempt to cut the graph based on the edge CE will produce two clusters of nodes: $\{A, B, C, D\}$ and $\{E, F, G, H\}$
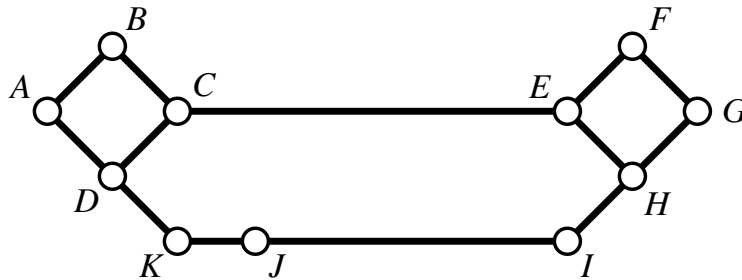
Figure 4.3: Example of failed clustering with a single overlength edge. In this case, cutting the graph based on the edge CE will not find two clusters of nodes.

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 E, the node at the opposite end of the edge CE, 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 our aim of 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.

4.1.3   Multiple Overlength Edges

There are a number of cases when finding single overlength edges with which to partition the graph is not 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).

Table 4.1: Breadth-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	$\{F\}$	$\{A, B, C, D, E, F, G,$ $H, I, J, K\}$	{}
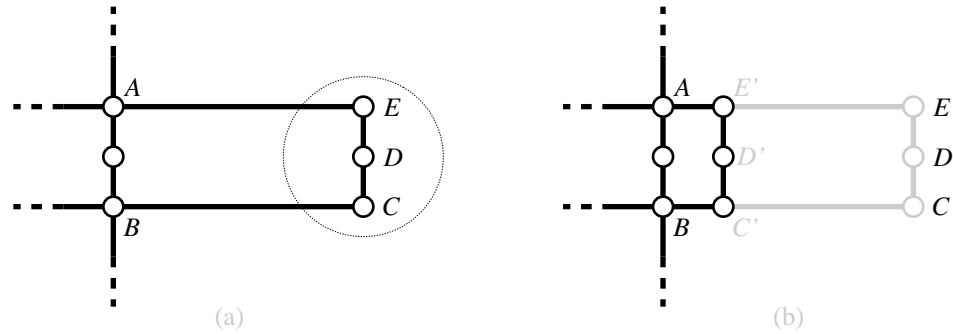
Figure 4.4: Clustering multiple overlength edges. The edges AE and BC are too long (a) and it is only possible to reduce the length of these edges by moving nodes C, D and E at the same time (b).

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.			

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 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 the one we started from. If it was not for this termination condition, 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 along the 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 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 edge 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
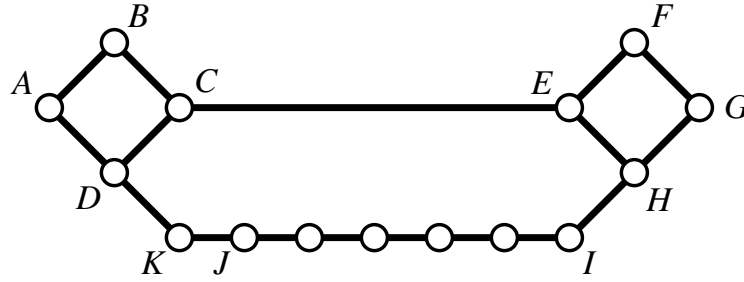
Figure 4.5: Example of failed clustering with overlength edges.

would not be possible to make an improvement such that the edge CE could be reduced in length. This is therefore an example of a local minimum that can not 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 that of lines which 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 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 A and F are discounted from forming part of a cluster
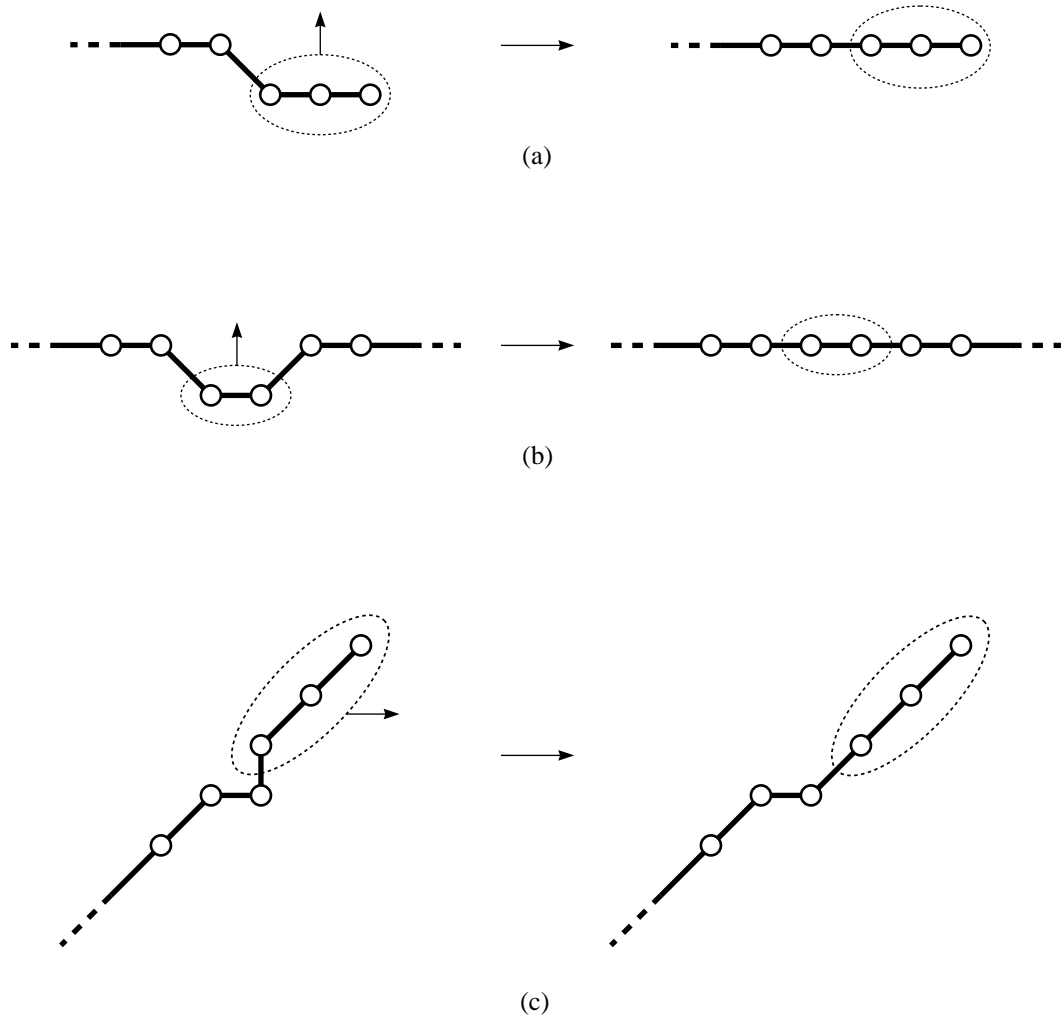
Figure 4.6: Finding non-straight lines. Potential clusters are highlighted by dashed areas. The right-hand side of (a), (b) and (c) show the result of moving the indicated clusters of nodes to a more optimal location. In the case of (c), the resulting movement of the cluster to the right introduces the possibility of reclustering in a similar manner to (a).
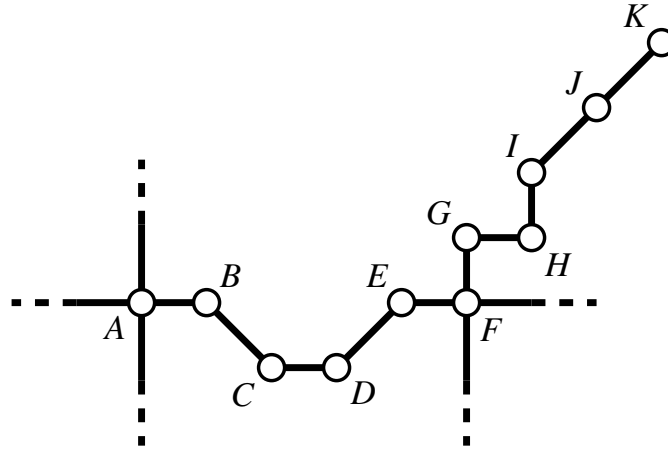
Figure 4.7: Clustering nodes to find non-straight lines. Ultimately, six clusters will be identified in this graph: $\{BC\}$, $\{CD\}$, $\{DE\}$, $\{GH\}$, $\{HI\}$ and $\{IJK\}$.

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 connected nodes which are collinear. 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, more complex situations can arise where a visual inspection of the graph would highlight cases where improvements can be made. These improvements to the visual aesthetics of the graph can be made by partitioning it into two parts and moving one of these partitions. These partitions usually cannot be discovered using the method for finding clusters by multiple overlength edges (Section 4.1.3) as 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:
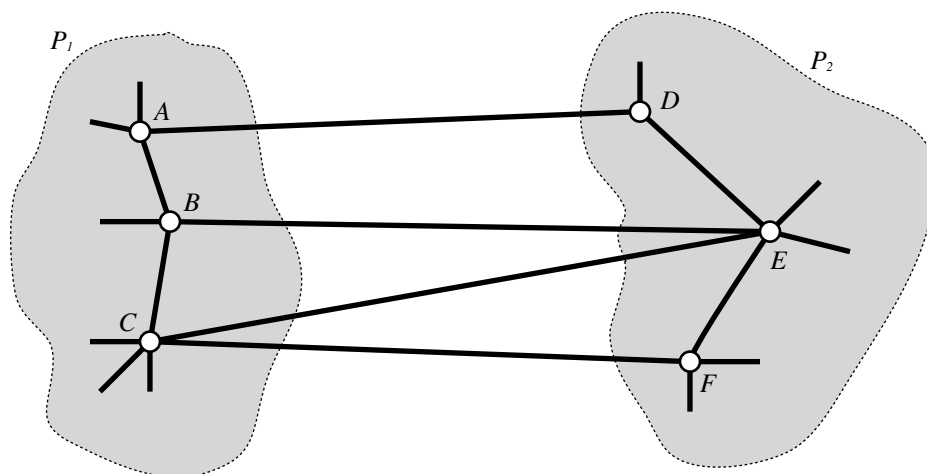
Figure 4.8: Partitioning a graph into two. The edges AD, BE, CE and CF are all too long but the only way of shortening them is to move either the left-hand partition P_1 or the right-hand partition P_2. Both P_1 and P_2 contain other overlength edges that would stop the method for clustering based on overlength edges from finding these partitions.

1. Preprocess the graph to remove edge crossings, unwanted multiple edges and trailing lines. 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 lines do not 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.

4.3.1   Finding Faces

When finding faces in a planar graph, we are essentially looking for minimal cycles that enclose a region of the graph. To do this, we must first ignore any multiple edges between any two nodes (i.e. where more than one line passes between two adjacent
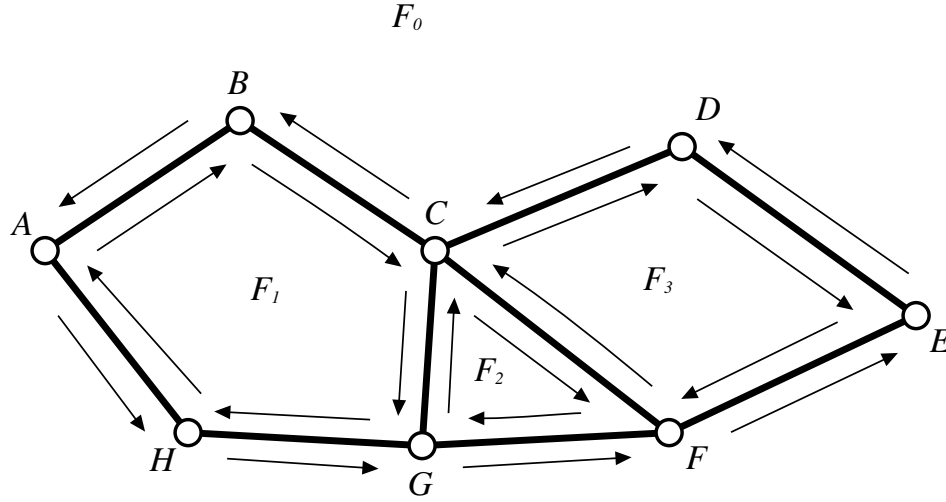
Figure 4.9: Finding faces. Arrows indicate the direction of traversal of each edge, for example, an arrow from A to B represents the from-to traversal $\overrightarrow{AB}$.

stations) by treating them as a single edge. We describe the method that we use to find the set of faces in the graph as we will reuse the idea of traversing edges in Section 4.3.4 when finding cuts in the graph.

We define a *traversal* of an edge as being either *to-from* or *from-to*. For example, an edge between nodes A and B can be traversed to A from B (to-from: denoted as $\overleftarrow{AB}$) or from A to B (from-to: $\overrightarrow{AB}$). When an edge has been traversed in both directions, we describe that edge as being *fully traversed*. The ultimate aim of our face-finding algorithm is to ensure that every edge has been fully traversed.

To find a face F, take any edge, e_0, that has not yet been fully traversed. We then follow that edge either to-from or from-to by consistently taking the next edge in an anticlockwise direction, e_n, at each node adding each edge to F. When $e_n = e_0$ we have found the minimal cycle that encloses F.

To illustrate this with an example, consider the graph shown in Figure 4.9. This example shows the direction of traversal of each edge in order to find each of the faces. For example, face F_1 can be found by starting at edge AB and following it through edges BC, CG, GH and AH. We can define each face in the graph by also giving the direction of traversal of each edge:

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

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

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

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

Notice that F_0 is the outside face. Every edge in F_0 is therefore also an *outside edge*.

In the previous section, we mentioned that trailing edges can be removed implicitly during the face finding algorithm. A trailing line is a connected subset of edges forming

a line containing only nodes with degree one or two. To remove these trailing edges from the graph during the face finding algorithm, we should remove edges e_n and e_{n+1} if $e_n = e_{n+1}$. In other words, if the node at the to-end of e_n has degree one.

4.3.2 Deriving the Dual Graph

The complete dual graph, G^*, of a planar graph, G, is the graph with nodes representing faces and with nodes being adjacent iff the corresponding faces in G are adjacent (i.e. the faces 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.

We can derive the dual graph from the sets of faces. An example of the derivation of a dual graph is shown in Figure 4.10.

4.3.3 Diminishing the Dual Graph

When finding appropriate cuts in the graph, it is not desirable to include in the dual graph those edges which are already at the preferred length, lg. The reason for this is that in finding cuts we are trying to reduce the length of many edges at the same time—if any edge is already at the preferred length then it will not be possible to move either partition without shortening the edge with preferred length. In diminishing the dual graph, G^*, we remove any edges that represent adjacent faces in G by way of an edge with length less than or equal to lg.

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, $G = \{A, B, C, D, E, H\}$, 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 with regard 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 one another. 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
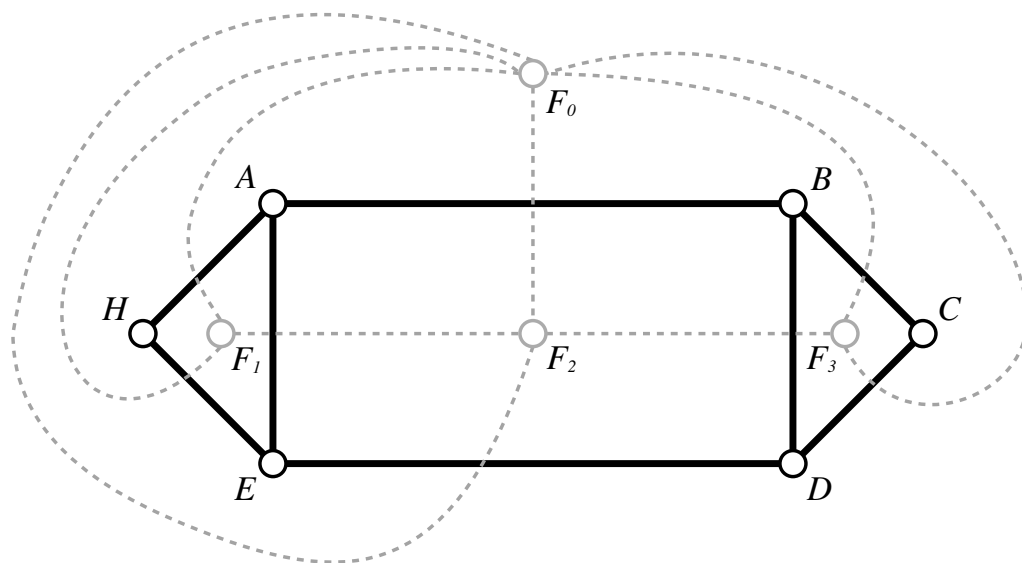
Figure 4.10: Example of a planar graph, G, and the complete dual graph, G^*. The planar graph $G = \{A, B, C, D, E, H\}$ is shown in black and the dual graph, $G^* = \{F_0, F_1, F_2, F_3\}$, is shown in grey with dotted edges.
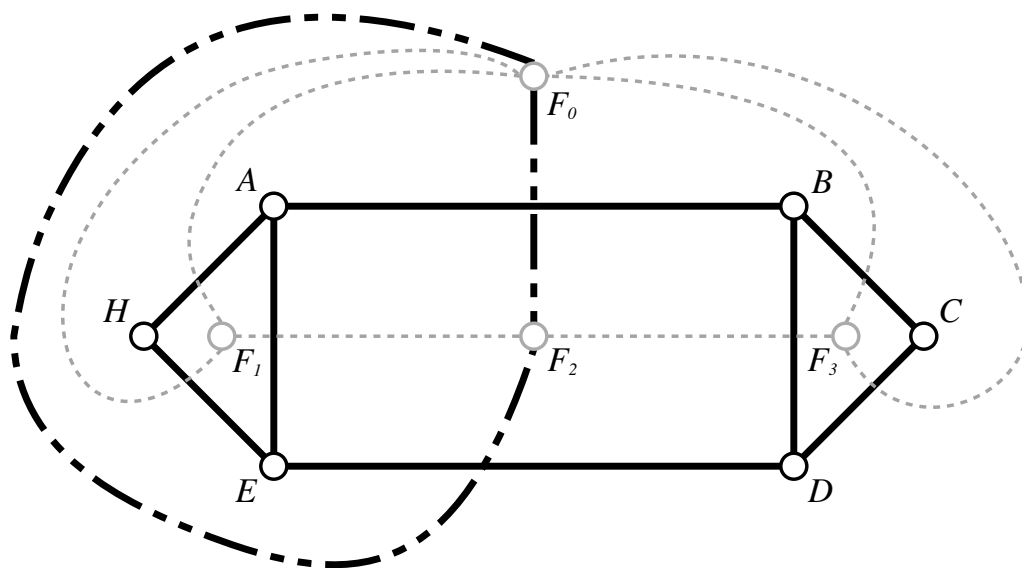
Figure 4.11: Example of a cycle in the dual graph equating to a cut in the planar graph represented by the dual graph. The cycle $\{F_0, F_2, F_0\}$ in the dual graph G^* (shown in bold) equates to the cut $P = \{AB, DE\}$ in the planar graph G.
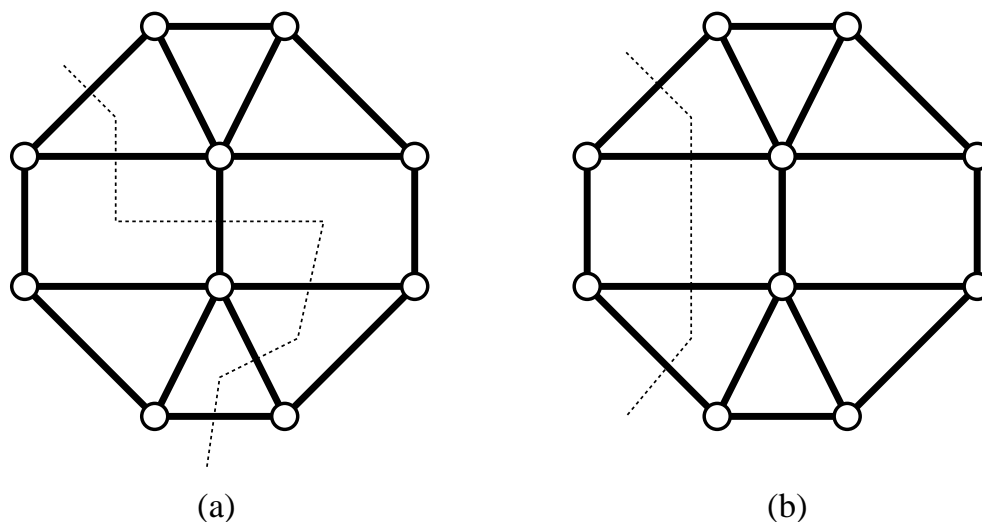
Figure 4.12: Example of cuts which are likely to lead to poor and good partitions. The example in (a) is likely to lead to poor partition selection while the example in (b) is likely to lead to a better partition selection.

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. For example, a cut should include only parallel edges but where this is not possible, the cut should contain edges as close as parallel as possible. In this case, a cut may contain, say, some number of horizontal edges and one diagonal edge. Figure 4.12 illustrates this: the cut in (a) is not likely to lead to partitions that can be improved using the optimiser while the cut in (b) consists of edges which are roughly parallel (although not exactly as it is not possible to cut the graph using just parallel edges other than horizontally through the middle) which means that the partitions are more likely to be improved by the optimiser.

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 number 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.
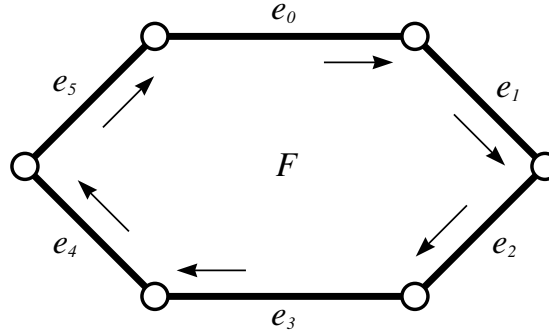
Figure 4.13: Finding the exit edge.

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 little likelihood of success.

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 start with the entry edge, e_0, and use the set of edges that describe the cycle forming the edge of the face, F.

We have to consider each edge in F as being directed such that the direction 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 4.13 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:

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

$$e_2 : \boldsymbol{E}_2 = \begin{bmatrix} -1 \\ -1 \end{bmatrix} \qquad e_3 : \boldsymbol{E}_3 = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$$

$$e_4 : \boldsymbol{E}_4 = \begin{bmatrix} -1 \\ 1 \end{bmatrix} \qquad e_5 : \boldsymbol{E}_5 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

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
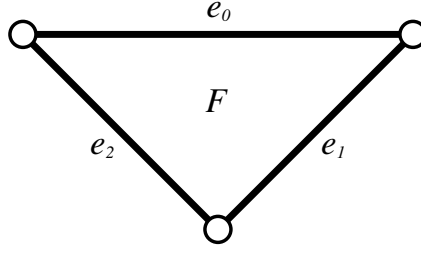
Figure 4.14: Finding the exit edge in a triangular cycle. Both edges e_1 and e_2 are candidates for being the exit edge but only one can be selected.

"other side" of F and also parallel to e_0 is considered as an exit edge.

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

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

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

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

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

As can be seen from these calculations, if the entry edge in Figure 4.13 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 4.13 as e_3 is both parallel and has opposite direction to the entry edge e_0.

In the case where there is 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 4.14 poses such a dilemma.

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

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

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:

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

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

We have three potential strategies for resolving this dilemma:

1. *Make an arbitrary decision.* It does not 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. Hence 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.
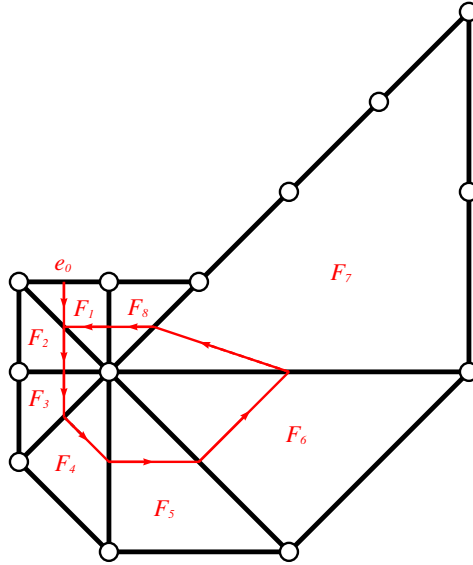
Figure 4.15: Non-terminating cutting algorithm. If the initial entry edge is e_0, the cut as indicated by the red lines with arrows will result in a perpetual loop in the cutting algorithm.

(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, this algorithm may never terminate. 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 4.15. 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, 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 $e_{\text{exit}} \in \Phi$.

As well as a non-terminating algorithm, Figure 4.15 also illustrates a case where the cut may not include edges which are parallel or mostly parallel. In fact, the cut illustrated includes almost no two edges with the same orientation. This comes about because 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 4.16. Also notice that the cycle that previously stopped the algorithm from terminating is no longer apparent.
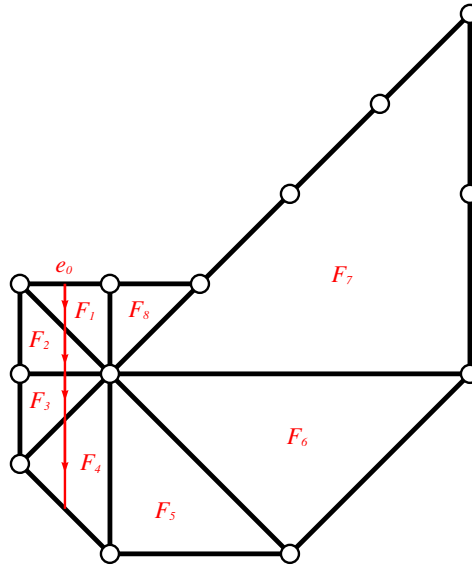
Figure 4.16: Modified cutting algorithm to remove the risk of an infinite loop in the cutting algorithm. The cut is indicated by the red lines with arrows.

4.4 Conclusion

This chapter introduced three methods to identify clusters of nodes in a graph based on overlength edges, bends in lines and partitions of the graph. These clusters were identified with the intention of achieving more optimal solutions when laying out metro maps using multicriteria optimisation by responding to the causes of some of the more common local minima in the search space.

The clustering methods were incorporated as an integral part of the multicriteria optimisation method for laying out metro maps as illustrated in Algorithm 3.2 (lines 13 to 13 form the part of the method that deals with identifying and moving clusters of nodes).

Chapter 5

Implementation

This chapter explains how we implemented a system to automatically lay out metro maps according to the methods introduced in the previous two chapters. The software that was written to support our method is described, including the format used for data to represent metro maps. We then provide a discussion regarding how appropriate criteria weightings were selected for each diagram as well as an analysis on particular local minima.

5.1 Software Implementation

We implemented our method using software written by us using the Java programming language [Jav08]. Java is an object-oriented programming language which makes it easy and logical to encapsulate functionality regarding each of the "objects" in a metro map in individual classes. We modelled a graph as a `Graph` object which contained a collection of `Nodes` and a collection of `Edges`. `Nodes` contained references to the incident `Edges` and `Edges` contained references to the two endpoint `Nodes`. A separate `HillClimber` class encapsulated the bulk of the functionality for each part of our method, including methods to move nodes and to determine clusters and partitions. Other classes represented station labels (`Label`) and faces (`Face`) used during the clustering stage (Section 4.3).

A Java Swing [JS008] graphical user interface allowed us to visualise the current embedding of a graph interactively, as shown in Figure 5.1. The user interface also allowed us to interact with the graph, allowing us to move nodes and create or delete nodes and edges. We were also able to specify settings for running our layout method including configuration of all of the criteria weightings and setting other parameters such as the grid spacing or number of iterations. We were able to watch the progress interactively while laying out a metro map.
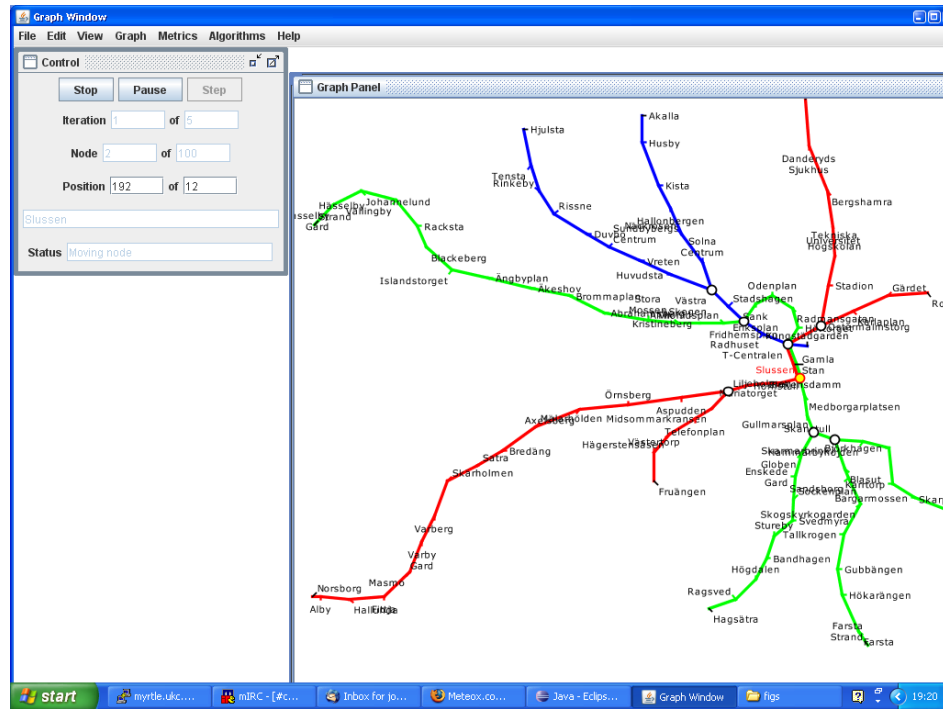
Figure 5.1: Screenshot of our metro map layout software showing the general map editing screen.

Additional features were implemented to allow us to investigate the values of the criteria for the immediate area around a particular node. This allowed us to direct our selection of criteria and the exploration of local minima.

Finally, we were able to import or export the maps using our own data format (described in Section 5.2) and to capture the current map using encapsulated PostScript (EPS) using EPS Graphics2D [Mut08] or as a PNG image file.

Figures 5.1 and 5.2 show screenshots of our software in action.

5.1.1 Computational Efficiency

We were not overly concerned about the efficiency of our software and there remain many areas that we could reduce the time taken to produce larger maps. For smaller maps of around thirty nodes we were able to produce drawings where each iteration took just a two or three minutes on a 1.8GHz single-core machine. Larger maps take an exponentially longer time to draw and maps in excess of 100 nodes took an hour or more per iteration, i.e., the London Underground with a few hundred nodes could take a three or four hours per iteration.

However, we did make some efficiency improvements where they resulted in a very significant reduction in running time. One such improvement was to not calculate some
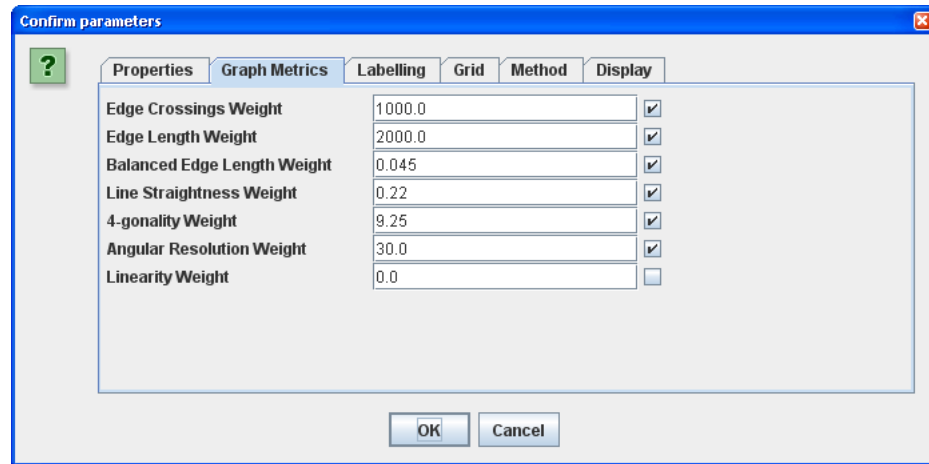
Figure 5.2: A second screenshot of our metro map layout software showing options for configuring criteria weightings.

of the attributes of edges such as their length or angle unless either of the endpoints of the edge changed position.

Improvements to the computational efficiency of our software remain for future research (see Section 9.3.4).

5.1.2 Calculation of Criteria in Practice

Each `Node`, `Edge` and `Label` object in our graph model maintained state relating to its position and dimensions in the drawing area. From these attributes we were able to compute the values for each criterion as and when required by providing methods in the `Graph` class. These methods returned `double`-precision floating point values.

5.2 Metro Map Data Format

We created a custom abstract data format to represent metro maps and to allow us to load and save maps from our software. The format consists of a list of `NODE` definitions and a list of `EDGE` definitions. Appendix B provides the data in this format as necessary for each of the example maps illustrated in Chapter 6.

A `NODE` definition looks like this:

```
NODE:label=Five Points:x=-84.3916:y=33.7539
```

where the `label` field contains the station name and the `x` and `y` fields contain the x- and y-coordinates for the node. In this case, the coordinates used are the latitude and longitude of the station.

An `EDGE` definition describes either a single edge or a sequence of edges to form a metro line. An example `EDGE` definition looks like this:

```
EDGE:label=Central:color=255,0,0:adjlist=Ealing Broadway,West Acton,
North Acton
```

where the `label` field contains the name of the metro line, `color` is the colour of the line (red, green and blue values from zero through 255) and `adjlist` contains a comma-separated list of station names. In this example, a metro line starting at Ealing Broadway and ending at North Acton and passing through West Acton is defined. A total of two edges would be created by this definition. The station names given in the `adjlist` must match exactly with the name given in the `label` field for that station.

5.3   Selection of Criteria Weightings

The criteria weightings that were used to produce the maps shown in Chapter 6 were produced mostly using trial and error. First, each criterion was weighted such that the values of the weighted criteria were approximately equal—this allowed us to negate the effect of some criteria being different orders of magnitude (normalisation). Next, the method was run on the map and the criteria modified depending on the results.

The weightings for some criteria are more sensitive to modification than others. This would be the case where two or more criteria have an effect on the same aspect of the graph. For example, both the edge length criterion (c_{N2}) and the balanced edge length criterion (c_{N3}) clearly have an effect on the lengths of edges; the line straightness criterion (c_{N5}) and the octilinearity criterion (c_{N6}) can both affect the orientation of edges. In these cases, altering the weighting for one of the criteria can also have an effect on the other criteria. This feedback is often the cause of local minima (see Section 5.4) and is the main reason why there are differences in some but not all of the criteria weightings for the maps in Chapter 6. In these cases, we developed a way within our software to be able to inspect each criterion independently for all the possible node positions in the search space. We were then able to make small modifications to the criteria weightings to fine-tune them to take account of the feedback.

An equivalent process was used for the label position criteria.

5.3.1   Effect of Criteria Weightings

The effect of each of the six node movement criteria on the graph shown in Figure 5.3 (of the Stockholm T-bana) are illustrated in the five figures from Figure 5.4 through Figure 5.8 (the edge crossings criterion is not illustrated here). In each of these five figures, the graph was drawn using the same parameters for everything except the criteria weightings. Ten iterations of our method were applied to each example. All criteria weightings were set to zero except the one in question so that the effect of that criterion in isolation can be studied.
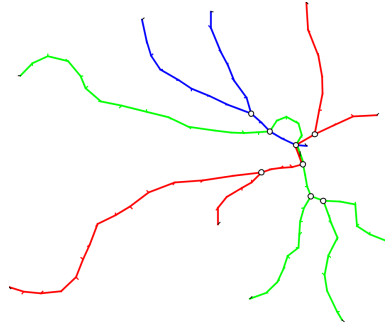
Figure 5.3: Graph used to illustrate the effect of criteria weightings (Stockholm T-bana).

(a)	(b)
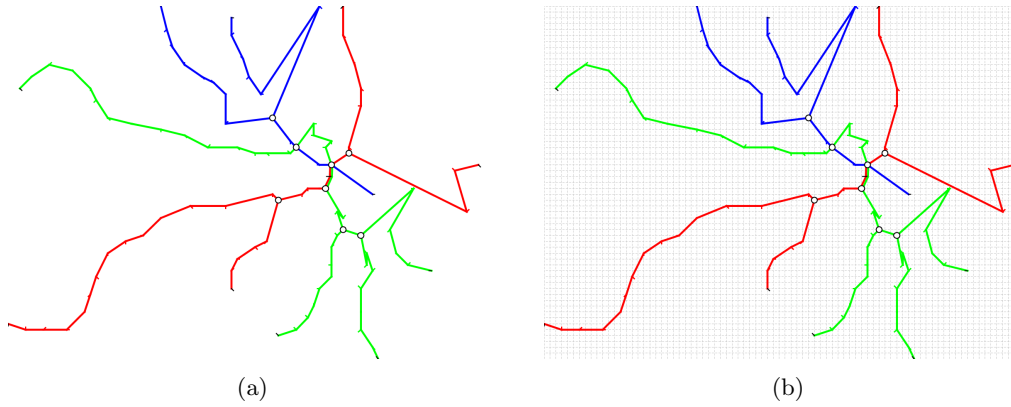
Figure 5.4: Effect of the angular resolution criterion, c_{N1}, shown without (a) and with the grid (b).

(a)	(b)
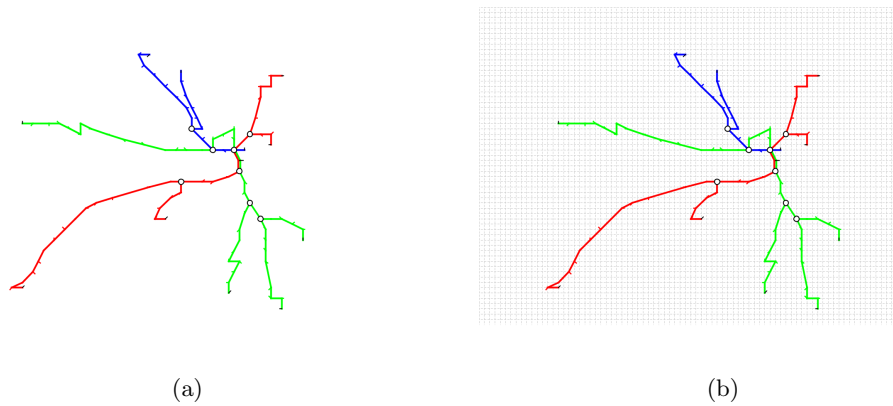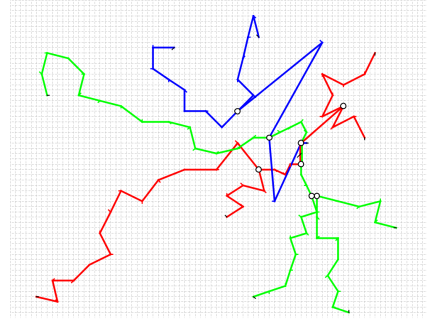
Figure 5.5: Effect of the edge length criterion, c_{N2}, shown without (a) and with the grid (b).
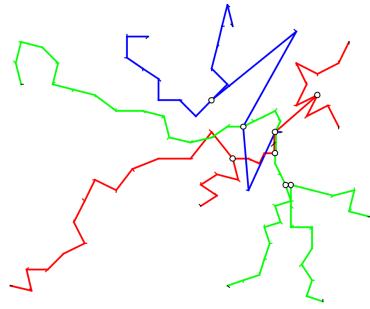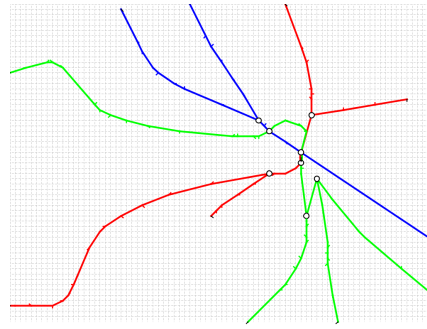
(a) (b)

Figure 5.6: Effect of the balanced edge length criterion, c_{N3}, shown without (a) and with the grid (b).
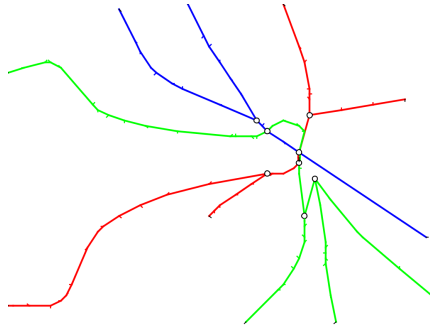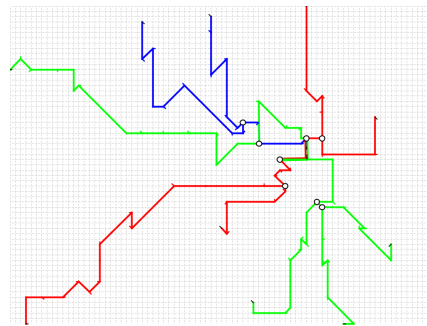
(a) (b)

Figure 5.7: Effect of the line straightness criterion, c_{N5}, shown without (a) and with the grid (b).
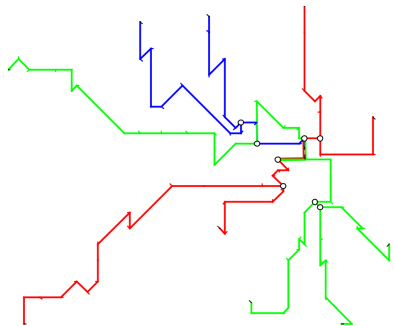
(a) (b)

Figure 5.8: Effect of the octilinearity criterion, c_{N6}, shown without (a) and with the grid (b).

As is to be expected, none of the criteria in isolation are able to produce an acceptable map that conforms to the metro map metaphor. The angular resolution criterion illustrated by Figure 5.4 only has an effect on nodes with degree three or more and as such, none of the other nodes were affected. The edge length criterion (Figure 5.5) and the balanced edge length criterion (Figure 5.6) both clearly have the desired effect on edge lengths, but at the expense of line straightness—especially so for the balanced edge length criterion. The line straightness criterion (Figure 5.7) produces the straightest lines while the octilinearity criterion (Figure 5.8) produces jagged lines but all are drawn vertically, horizontally or at 45° diagonal.

5.4   Local Minima

This section identifies and explains a number of local minima that are evident when applying our method to different maps. All of these local minima can be illustrated by examples in the results presented in Chapter 6. This is not an exhaustive list, but these are the most common examples that we discovered:

- Triangular faces (Section 5.4.1)

- Conflict between angular resolution and line straightness (Section 5.4.2)

- Conflict between octilinearity and line straightness (Section 5.4.3)

- Long labels (Section 5.4.4)

- Labels on the inside of line bends (Section 5.4.5)

5.4.1   Triangular Faces

Triangular faces can cause local minima when there is no 90° internal angle or the two edges adjacent to the 90° angle are not the same length. No other triangular face will have three octilinear edges. As such, examples like that illustrated in Figure 5.9 can be fairly common. In this example, node C should be moved to the left to reduce the length of edge AC and to make edge BC octilinear. However, this would be a local minimum if node C cannot move to the left if this causes an increase in other criteria. In metro maps, triangular faces tend to appear in dense areas of the map—areas that are least likely to be improved by moving individual nodes. Examples of local minima involving triangular faces in our results can be seen in the maps of Mexico City (Section 6.1) and San Francisco (Section 6.5).

5.4.2   Conflict Between Angular Resolution and Line Straightness

Figure 5.10 shows an example of a local minimum caused by a conflict between the line straightness and angular resolution criteria. In this case the line straightness criteria
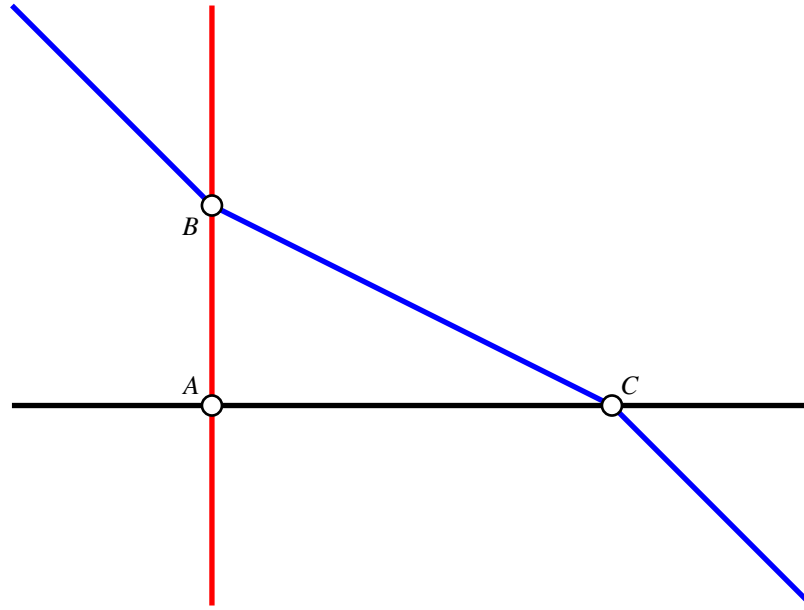
Figure 5.9: An example of a local minimum involving a triangular face.

would attempt to move nodes E and D upwards so that the red and blue lines are straighter. However, this would cause $\angle ABE$ and $\angle CBD$ to become too small when considering the angular resolution criterion around node B. In this example, moving nodes E or D would also cause the octilinearity criterion to worsen. This conflict between criteria is common in dense areas of maps where many lines pass through some nodes. A particular example of this is highlighted with our result for the Mexico City map (Section 6.1).

5.4.3 Conflict Between Octilinearity and Line Straightness

The octilinearity and line straightness criteria can often conflict. This is illustrated by two examples shown in Figure 5.11 and Figure 5.12.

In the first example, three edges are not octilinear but the line is quite straight. If the value of the line straightness criterion is larger than the value of the octilinearity criterion then the line can never be improved without moving two nodes simultaneously in opposite directions (of the two nodes in the centre, the left-hand one would have to move down and the right-hand one would have to move up). If only one node is moved, the octilinearity criterion is improved but the line straightness criterion is worsened.

Similarly, in the second example, a line that zig zags cannot be improved without causing the octilinearity criterion to worsen. Node A would have to move to the new position A' and node B would have to move to the new position B' in order to improve the line straightness criterion. However, the position of the node between A and B would cause the two edges between A' and B' to not be octilinear.
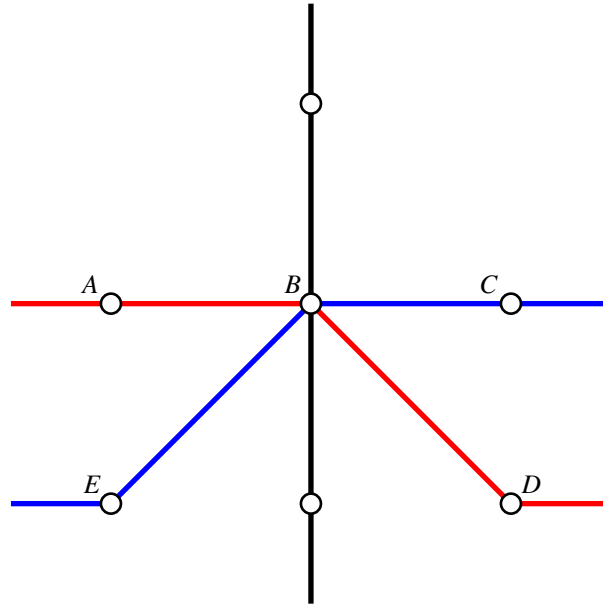
Figure 5.10: An example of a local minimum involving a conflict between the angular resolution and line straightness criteria.
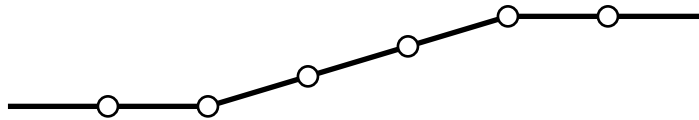
Figure 5.11: An example of a local minimum involving a conflict between the octilinearity and line straightness criteria.

Examples of these two local minima can be seen in the results for the Atlanta (Section 6.2) and Stockholm (Section 6.6) maps.

5.4.4 Long Labels

Long labels are problematic for all algorithms involving map labelling and our method is no exception. While we have attempted to mitigate the problem by splitting long labels over multiple lines, this is not always effective. The example in Figure 5.13 shows an extreme case where the node in the centre of the graph has a very long label. Two possible positions for the label are shown but all of the positions in the labelling space will cause an occlusion with other edges and labels. In this case, the label could fit by increasing the number of lines over which the text is divided. However, our method does not have any way to do this at present.

Long labels do occur in some of our results in Chapter 6, notably those of San Francisco (Section 6.5) and Washington D.C. (Section 6.9).
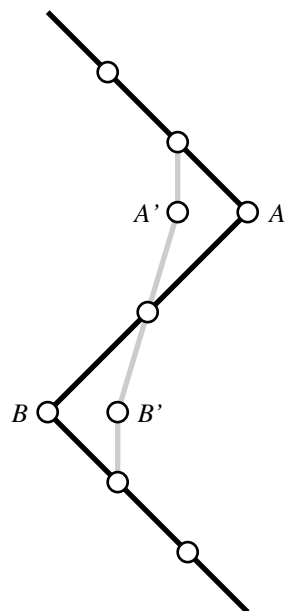
Figure 5.12: A second example of a local minimum involving a conflict between the octilinearity and line straightness criteria.
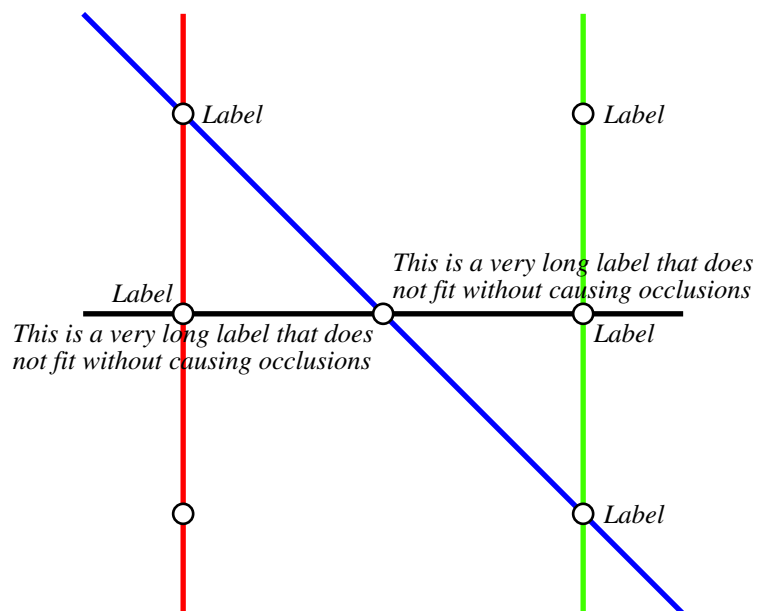
Figure 5.13: An example of a local minimum involving long labels.
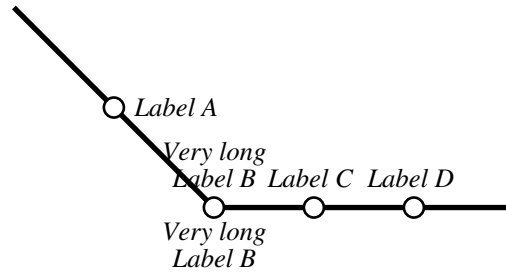
Figure 5.14: An example of a local minimum involving labels positioned on the inside of line bends.

5.4.5   Labels on the Inside of Line Bends

One further example of a local minimum involving labels is that of labels on the inside of line bends. This is illustrated in Figure 5.14 where label B is long. The label position consistency criterion would prefer the label for node B to be positioned above or to the right of the node, but neither of these positions are possible without causing edge occlusions or an occlusion with the label for node C. The ideal position for label B is below the line (shown in grey), but the label position consistency criterion will not recommend this position.

The Auckland map in our results (Section 6.3) illustrates an example of this local minimum.

Chapter 6

Results and Discussion

This chapter presents a number of examples of our method when applied to various real-world metro maps. The following maps are illustrated here:

1. Mexico City (Section 6.1): in-depth discussion of the method.

2. Atlanta MARTA (Section 6.2).

3. Auckland (Section 6.3).

4. Bucharest (Section 6.4).

5. San Francisco BART (Section 6.5).

6. Stockholm T-bana (Section 6.6).

7. Sydney CityRail (Section 6.7).

8. Toronto (Section 6.8).

9. Washington D.C. WMATA (Section 6.9).

In the case of the Mexico City map, an in-depth discussion is presented which aims to illustrate how the method performs on a substantially-sized map with 175 nodes. For the Sydney CityRail map, comparisons are given between the results produced by our method and those by Hong *et al.* [HMdN04, HMdN06], Nöllenburg [Nöl05] and Merrick and Gudmundsson [MG07].

Section 5.3 discussed how the weightings were determined. Each example shown in this Chapter has the weightings and other parameters detailed. To help the reader, small excerpts of the final map are shown when they are referenced.

6.1   Mexico City: Discussion

The Mexico City *Sistema de Transporte Colectivo Red del Metro* map [Mex08] is a complex, decentralised map. It has a relatively high number of lines and faces and a total of 175 stations (Table 6.1). The officially published map is shown in Figure 6.7. Unusually for a metro system in a large city, there is no obvious centre of the map and most lines intersect many other lines (the purple line 1 intersects all but one of the other lines). Also unusual is that the officially published map lacks any clear use of the characteristics of metro maps described in Section 2.2. The geographic sketch map of the network is shown in Figure 6.2. The high number of faces in this map should provide an interesting illustration of our partitioning algorithm.

The map produced using our method is shown in Figure 6.8. We used the criteria weightings shown in Table 6.2 to produce this map. For this map, we have produced intermediate layouts showing the map at the end of the first iteration (Figure 6.3), the second iteration (Figure 6.4) and the third iteration (Figure 6.5). For the fourth iteration, the changes were negligible, so the intermediate layout is omitted.

To illustrate how the method optimises the layout, the charts in Figure 6.6 show how the total weighted node movement criteria, m_N, decreased during each iteration. These charts show that the majority of the optimisation occurred during the first iteration and by the fourth iteration there was only a small improvement. Table 6.3 shows the values of m_N after the node movement and cluster movement phases. The number of nodes identified in each cluster is also shown. As is to be expected, the number of clusters identified at each iteration falls. By the fifth iteration, no improvement to m_N was made and the value of m_N was 0.156 times that of the initial value.

Our finished map shows a significant enhancement over both the geographic and official maps. The spacing between stations is much more regular and nearly all the edges have been drawn octilinearly. The labelling is also of high quality, particularly along long lines. The line to the top-right of the map (see Figure 6.1(a)) shows an interesting meander where the line has been compressed horizontally in order to fit within the bounds of the drawing area. Due to the large number of faces in this map, the clustering by partitioning algorithm was very effective in straightening a number of lines and compressing some overlength edges (that would not have compressed using the overlength edges clustering algorithm). However, the clustering and partitioning algorithms were not able to address all non-octilinear edges and some examples remain.

Examples of local minima are notable in our map, particularly where several lines pass through a station (as at *Chabacano*, Figure 6.1(b)) or where a triangular face exists (such as the one bounded by *Morelos*, *San Lazaro* and *Candelara*, Figure 6.1(c)). In the first of these examples, two lines are passing through a single station from east to west. The line straightness criterion tends to force these lines to become horizontal thereby reducing the angle between them. This could be avoided by increasing the weighting

for the angular resolution criterion, but in practice this tends to result in less optimal conditions elsewhere in the map. The second local minima example involves triangular faces. These faces can only be drawn with each edge being octilinear when two of the edges are of the same length and one of the internal angles is 90° (the face then forms an isoceles triangle).

(a)   (b)   (c)
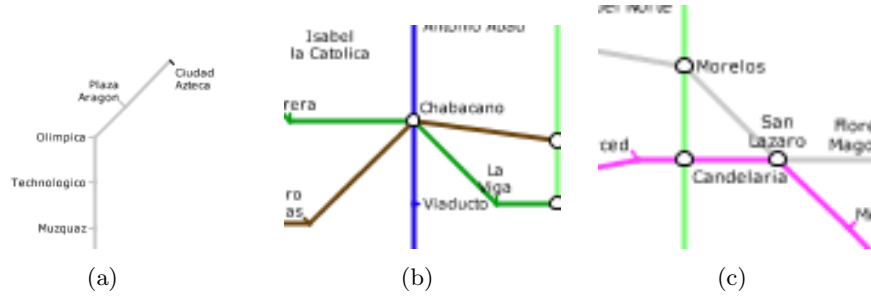
Figure 6.1: Specific referenced examples in the automatically-drawn Mexico City map.

Table 6.1: Characteristics of the Mexico City map.

Stations	175
Interchange Stations	24
Edges	165
Lines	11
Faces	19

Table 6.2: Criteria weightings and parameters used for the Mexico City map.

Node movement		Labelling		Other Parameters	
w_{N1}	30000.0	w_{L1}	300.0	Iterations	5
w_{N2}	50.0	w_{L2}	80.0	Pref. grid spacing l	4
w_{N3}	45.0	w_{L3}	19.0	Grid Spacing g	40
w_{N4}	1000000.0	w_{L4}	15.0	Min. Cluster Distance	3
w_{N5}	220.0	w_{L5}	3.0	Max. Node Movement	8
w_{N6}	9250.0	w_{L6}	1.0		
		w_{L7}	30.0		
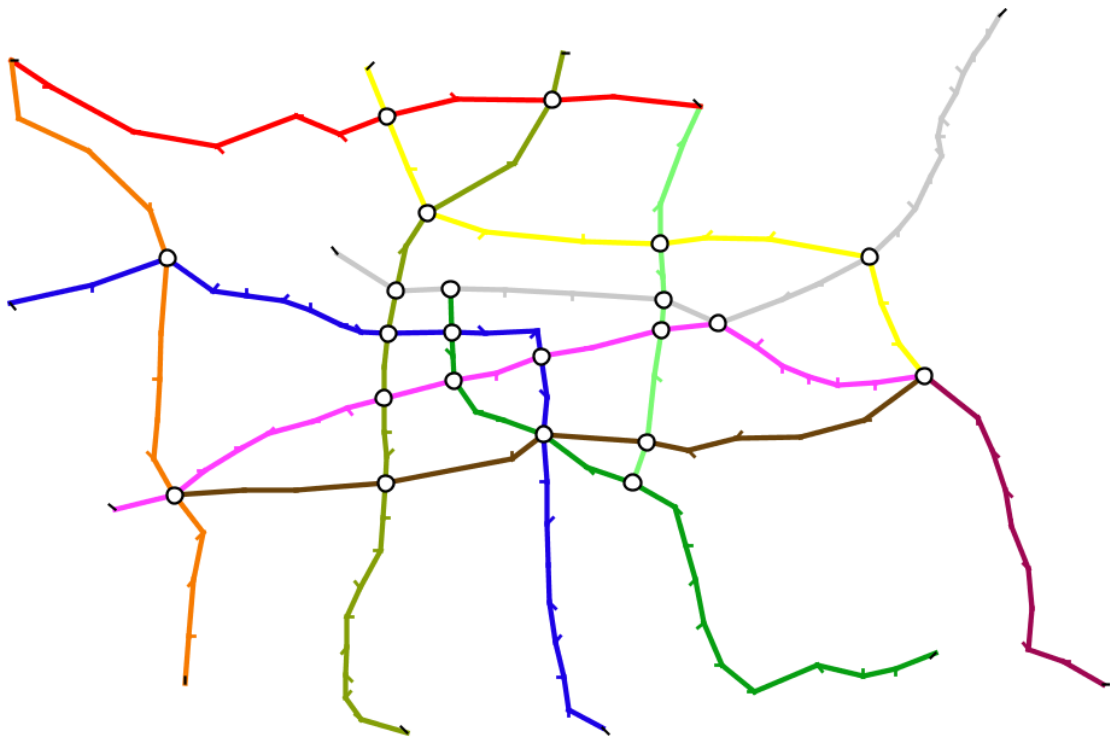
Figure 6.2: Mexico City map: geographic layout.
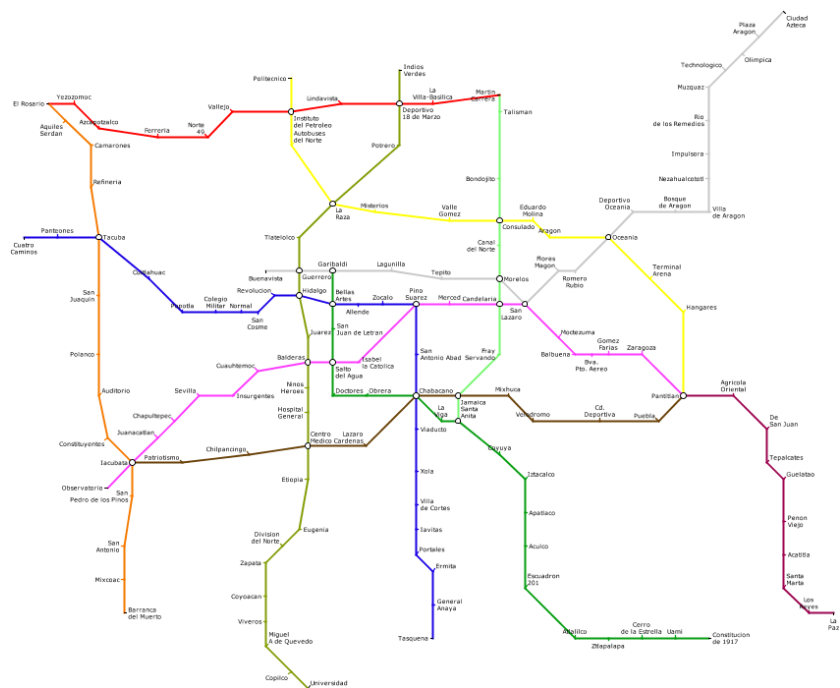
Figure 6.3: Mexico City map: after iteration 1.
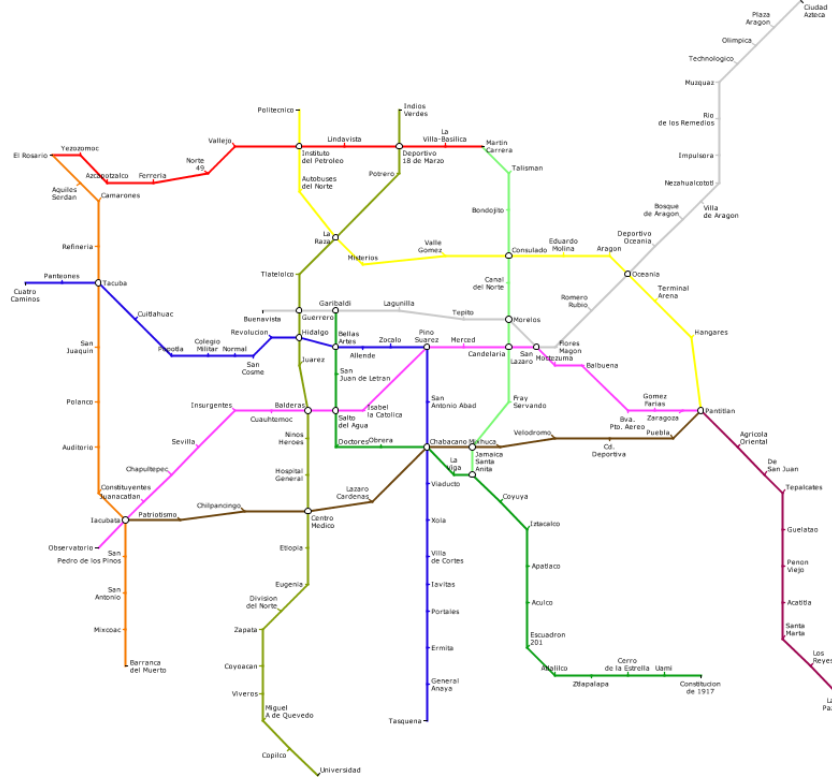
Figure 6.4: Mexico City map: after iteration 2.

Table 6.3: Total weighted node criteria, m_N, over five iterations of drawing the Mexico City map.

Iteration	m_N after moving nodes	m_N after moving clusters	Clusters
Start	4.408×10^8		
1	1.764×10^8	1.130×10^8	172
2	0.941×10^8	0.788×10^7	125
3	0.741×10^7	0.724×10^7	108
4	0.691×10^7	0.689×10^7	104
5	0.689×10^7	0.689×10^7	90
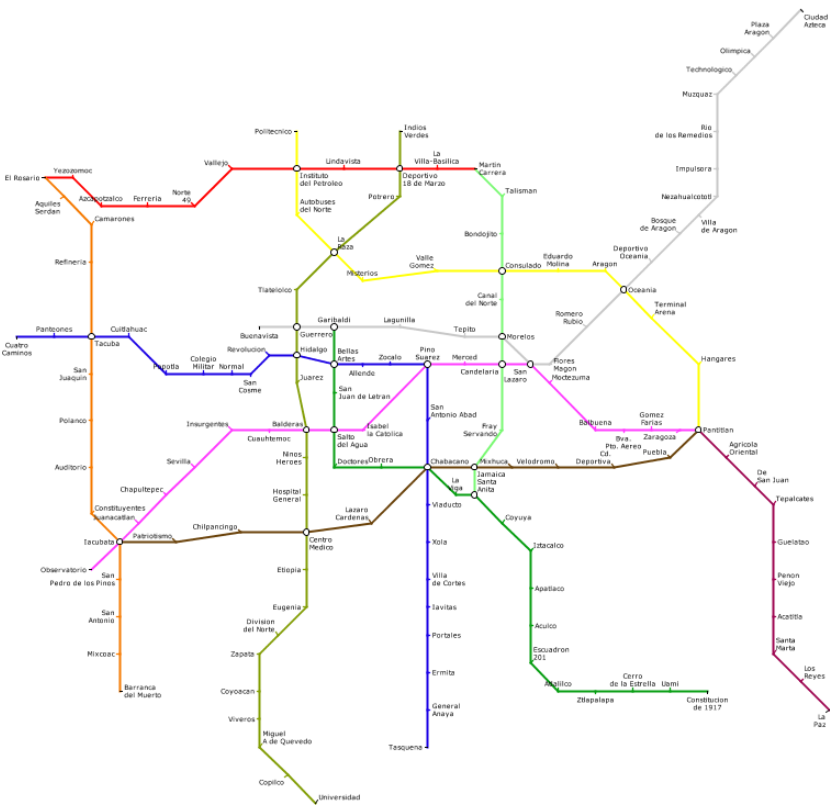
Figure 6.5: Mexico City map: after iteration 3.
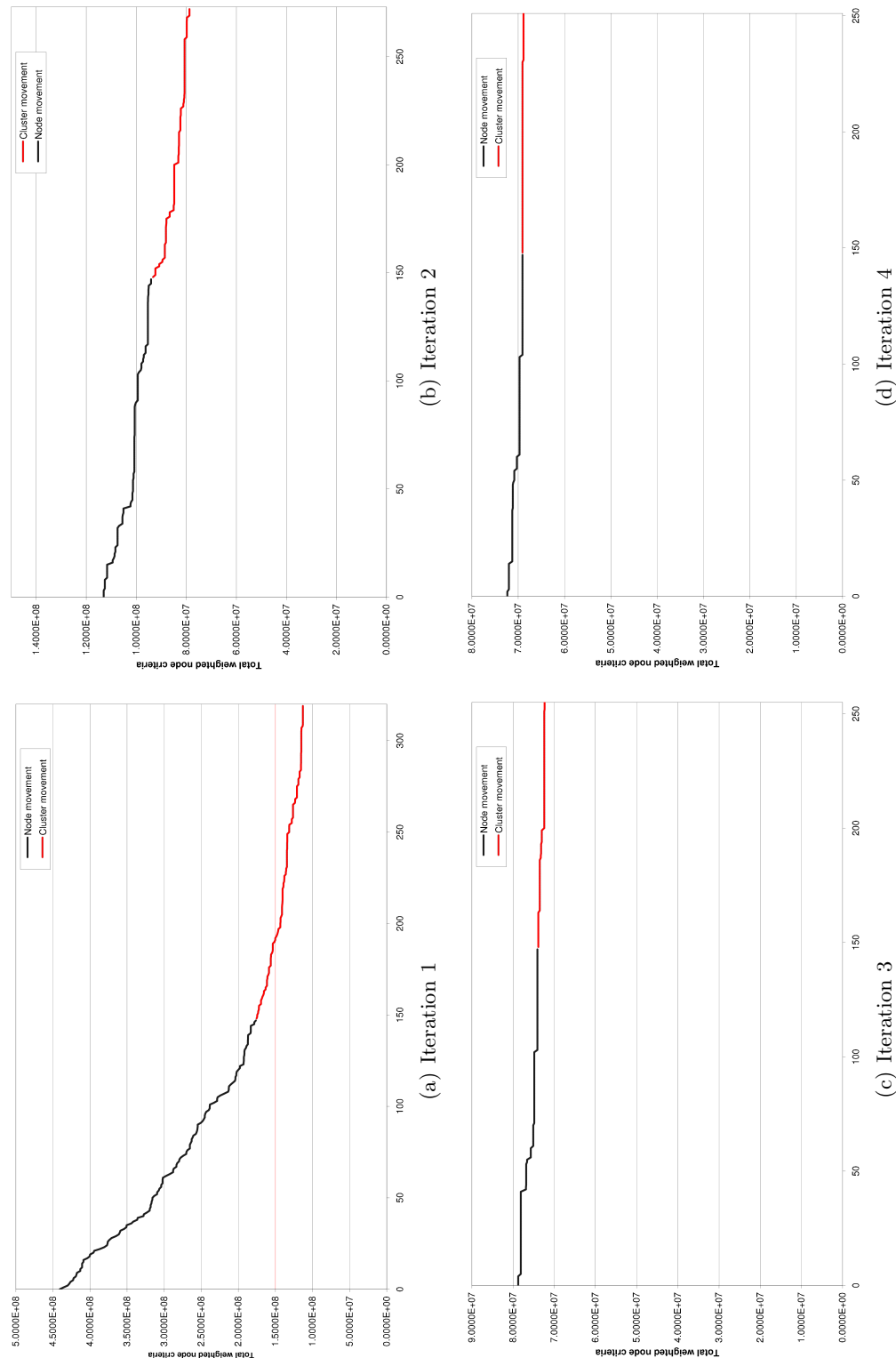
Figure 6.6: Charts showing the decrease in the value of the total weighted node and labelling criteria over five iterations of drawing the Mexico City map. The x-axis on each chart represents the progression of each iteration as each node or cluster is moved.
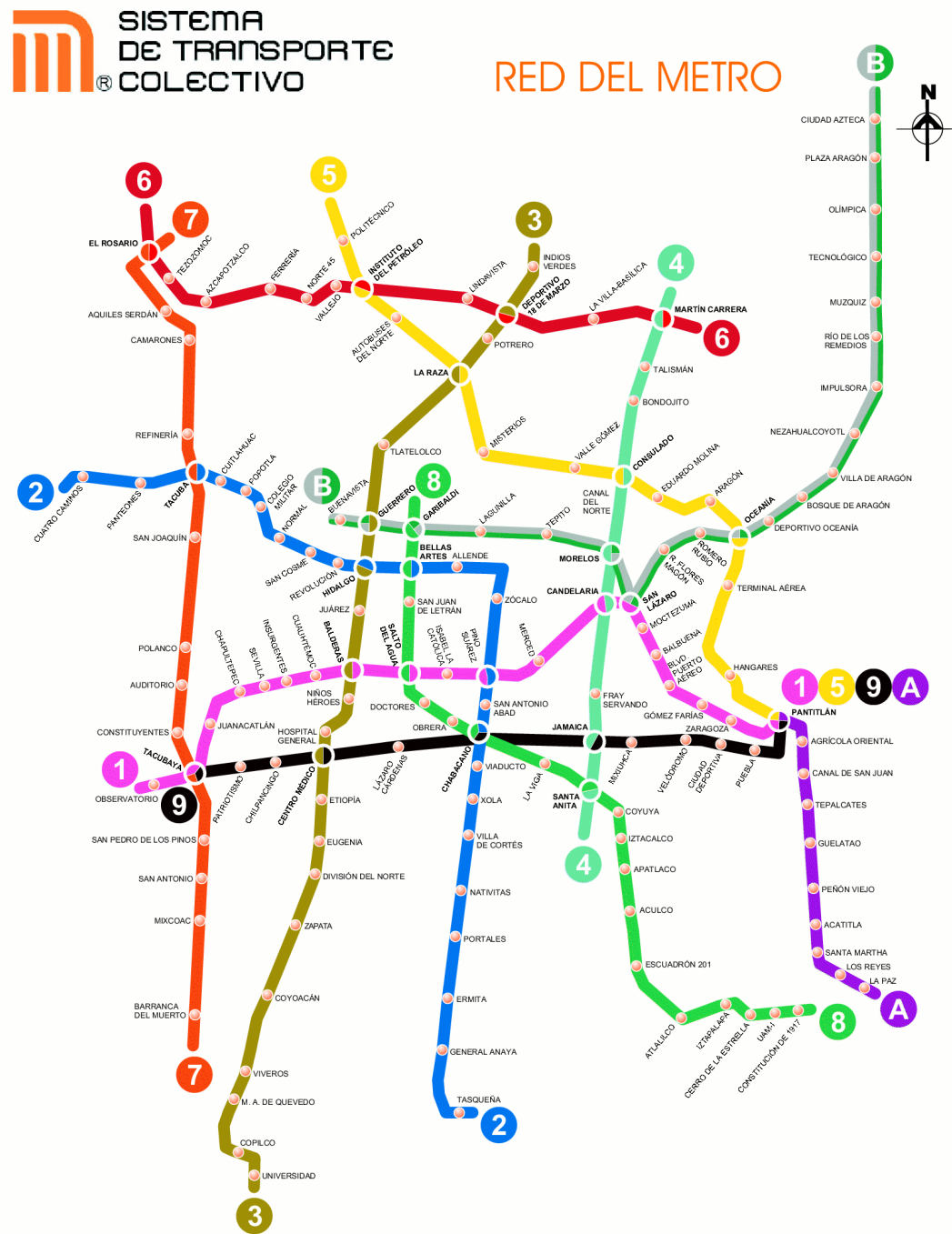
Figure 6.7: Mexico City map: official layout.

Figure 6.8: Mexico City map: our layout.

6.2 Atlanta

The Atlanta MARTA metro map [MAR08] is a simple radial map with just two lines and a single face (the outside face). As such this map should show the effect of clustering overlength edges and non-straight lines. Other characteristics of the Atlanta map are shown in Table 6.4. The official published map is shown in Figure 6.11 and the geographic sketch map is shown in Figure 6.10. The published map shows a small amount of metadata in the form of some highways. Note that our maps were produced based on an older map than that shown in Figure 6.11—the only difference is that the blue east-west line and the orange north-south line are split into two lines (blue and green for the east-west line and red and orange for the north-south line).

The map produced with our method is shown in Figure 6.12 which was produced using the criteria weightings shown in Table 6.5.

Our layout method clearly demonstrates the efficacy of our clustering algorithms for overlength edges and non-straight lines. The spacing of stations along each line is much more regular than both the geographic map and the official published map. The three edges between *Inman Park/Reynoldstown* and *Decatur* (shown in Figure 6.9(a)) were not picked up by any of the clustering algorithms, but they might have been drawn more octilinearly if the octilinearity criterion was increased. Labelling is generally of a high standard, except the area just to the left of *Five Points*, shown in Figure 6.9(b). In this case, a very long label (*Dome/Philips Arena/GWCC/CNN*) would require much more space when placing it horizontally and therefore the label obscures one other station just below *Five Points*. The published map handles the labelling along the horizontal east-west lines using diagonal labels.

(a)   (b)
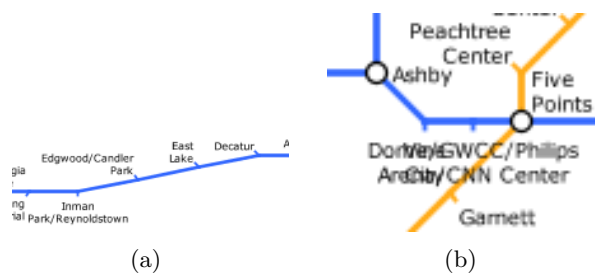
Figure 6.9: Specific referenced examples the automatically-drawn Atlanta MARTA map.
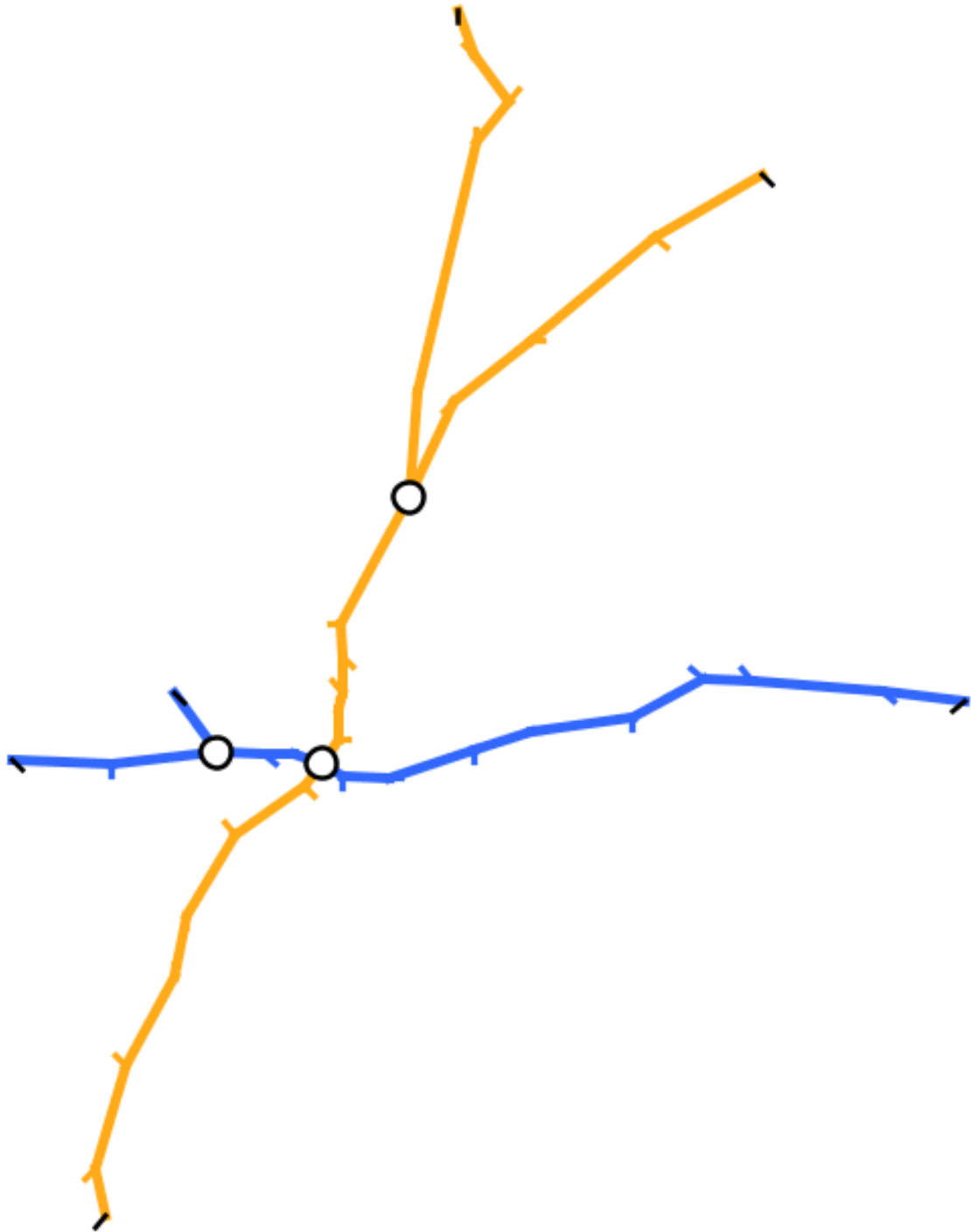
Figure 6.10: Atlanta MARTA map: geographic layout.

Figure 6.11: Atlanta MARTA map: official layout.
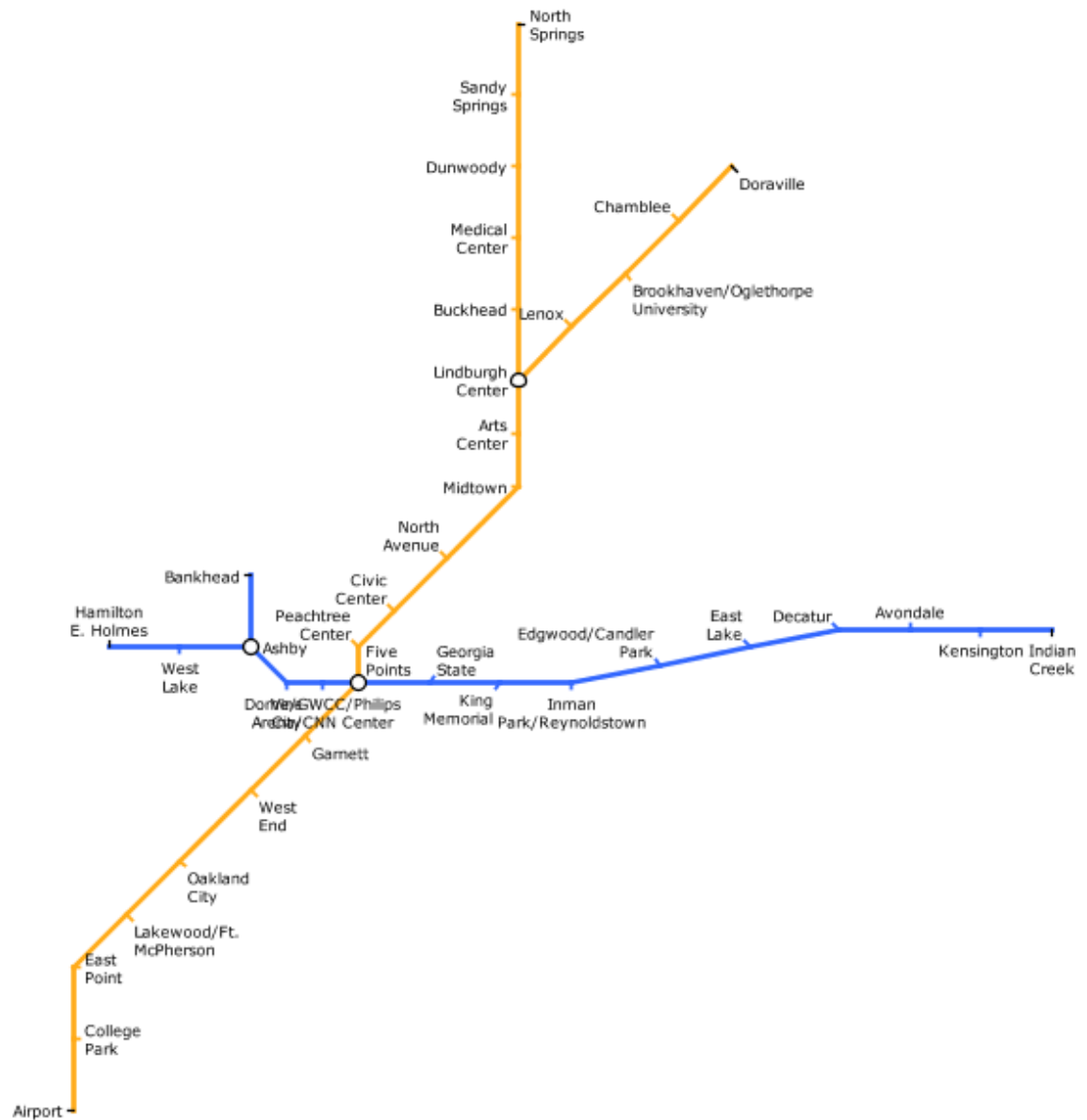
Figure 6.12: Atlanta MARTA map: our layout.

Table 6.4: Characteristics of the Atlanta MARTA map.

Stations	39
Interchange Stations	3
Edges	36
Lines	2
Faces	1

Table 6.5: Criteria weightings and parameters used for the Atlanta MARTA map.

Node movement		Labelling		Other Parameters	
w_{N1}	30000.0	w_{L1}	300.0	Iterations	5
w_{N2}	50.0	w_{L2}	80.0	Pref. grid spacing l	4
w_{N3}	45.0	w_{L3}	19.0	Grid Spacing g	50
w_{N4}	1000000.0	w_{L4}	15.0	Min. Cluster Distance	3
w_{N5}	2200.0	w_{L5}	3.0	Max. Node Movement	8
w_{N6}	9250.0	w_{L6}	1.0		
		w_{L7}	30.0		

6.3   Auckland

The Auckland map [Auc07] is a simple radial map with a small loop in the city centre. The official published map (Figure 6.15) has a strong horizontal bias, probably so that the map can fit easily in train carriages. The geographic map (Figure 6.14) shows that the map really features an east-west line and two north-south lines which run in parallel for much of the way. The characteristics of the Auckland map are shown in Table 6.6.

The version of the Auckland map drawn using our method is shown in Figure 6.16 and uses the criteria weightings given in Table 6.7.

Our finished map clearly follows the topography of the network more than the published map. As such, the relative positions of stations are more obvious. The loop in the centre of the map is quite effective—the two lines heading south of the city centre pass through *Westfield* in as straight a line as possible without octilinearity of angular resolution being sacrificed (see Figure 6.13(a)). The spacing of stations along the lines is also very regular, especially considering that the spacing of stations on the geographic map is quite irregular. Labelling is generally good, but the bends in the long lines have forced some unattractive position inconsistencies. One label (*Glen Eden*) occludes an edge (see Figure 6.13(b)), but in this case the position consistency and position criteria preferred the occlusion than have the label on the other side of the line. *Avondale* is a similar situation to *Glen Eden* but this node was moved as otherwise the label would have occluded the label for *New Lynn*.

Figure 6.13: Specific referenced examples in the automatically-drawn Auckland map.

Table 6.6: Characteristics of the Auckland map.

Stations	39
Interchange Stations	1
Edges	52
Lines	3
Faces	2

Table 6.7: Criteria weightings and parameters used for the Auckland map.

Node movement		Labelling		Other Parameters	
w_{N1}	30000.0	w_{L1}	300.0	Iterations	4
w_{N2}	50.0	w_{L2}	80.0	Pref. grid spacing l	4
w_{N3}	45.0	w_{L3}	19.0	Grid Spacing g	50
w_{N4}	1000000.0	w_{L4}	15.0	Min. Cluster Distance	3
w_{N5}	220.0	w_{L5}	3.0	Max. Node Movement	8
w_{N6}	9250.0	w_{L6}	1.0		
		w_{L7}	30.0		
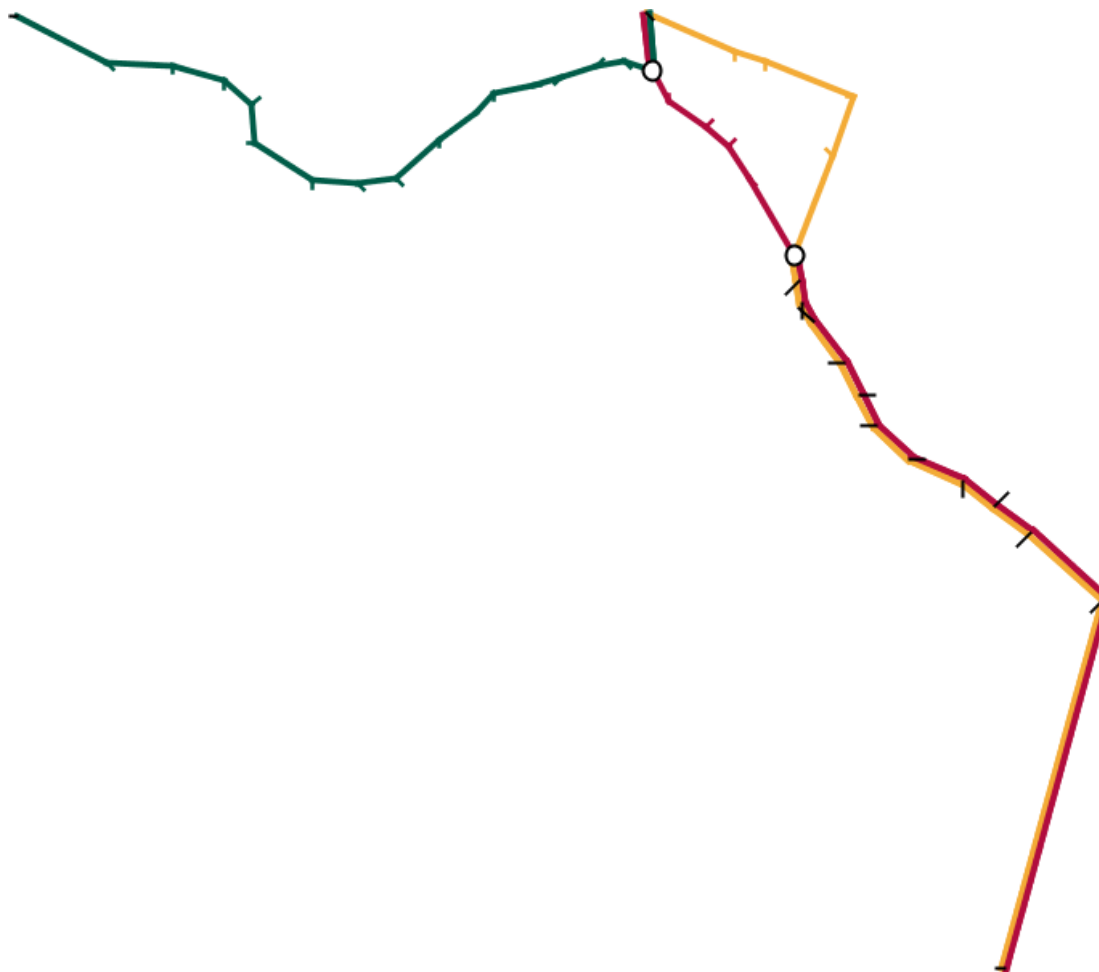
Figure 6.14: Auckland map: geographic layout.

Figure 6.15: Auckland map: official layout.

Figure 6.16: Auckland map: our layout.

(a)
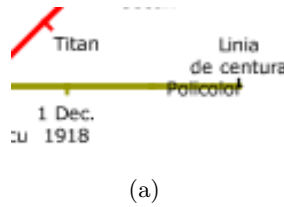
Figure 6.17: Specific referenced examples in the automatically-drawn Bucharest map.

6.4 Bucharest

The Bucharest metro map [Met08] is an example of a centralised, radial map but with an interesting circular line in the middle (the red line). The official published map is shown in Figure 6.19 and the geographic sketch map is shown in Figure 6.18. Characteristics of the map are given in Table 6.8. The official map features two notable curiosities: an unorthodox kink in the red line on the right-hand side of the map and a similar kink in the same line on the left-hand side of the map.

The map drawn using our method is shown in Figure 6.20 and uses the criteria weightings given in Table 6.9.

Our map clearly demonstrates the application of many of the characteristics of metro maps. Most of the edges are drawn octilinearly and the circular loop remains prominent. The map labelling is reasonably good. There are a handful of examples where labels occlude lines (for example, that shown in Figure 6.17(a))—in these cases, a local minimum was reached where the weighting on the occlusion criteria did not quite outweigh a combination of the other labelling criteria.

Table 6.8: Characteristics of the Bucharest map.

Stations	45
Interchange Stations	6
Edges	45
Lines	3
Faces	3

Table 6.9: Criteria weightings and parameters used for the Bucharest map.

Node movement		Labelling		Other Parameters	
w_{N1}	30000.0	w_{L1}	300.0	Iterations	5
w_{N2}	50.0	w_{L2}	80.0	Pref. grid spacing l	4
w_{N3}	45.0	w_{L3}	19.0	Grid Spacing g	50
w_{N4}	1000000.0	w_{L4}	15.0	Min. Cluster Distance	3
w_{N5}	220.0	w_{L5}	3.0	Max. Node Movement	8
w_{N6}	9250.0	w_{L6}	1.0		
		w_{L7}	30.0		

Figure 6.18: Bucharest map: geographic layout.
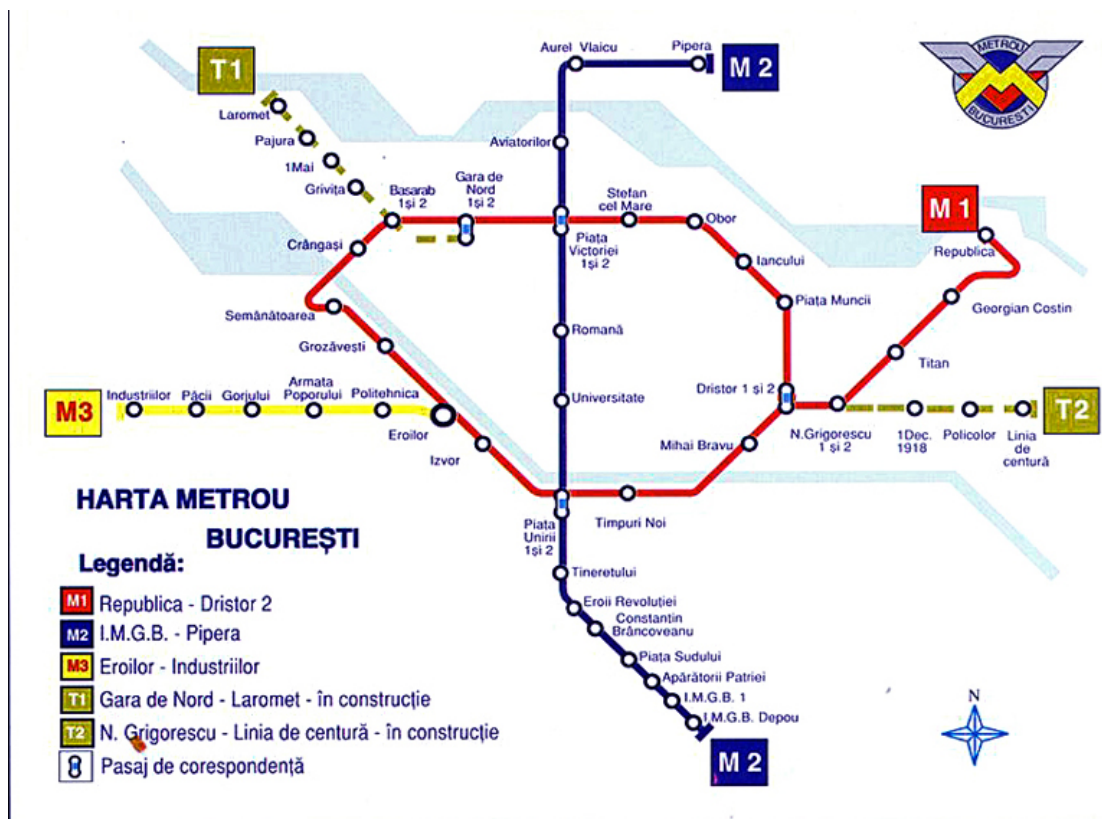
Figure 6.19: Bucharest map: official layout.

Figure 6.20: Bucharest map: our layout.

6.5 San Francisco BART

The San Francisco Bay Area Rapid Transit (BART) map [BAR07] features five lines radiating from the San Francisco Bay area of California. The network covers a sizeable area: around 60km from north to south and 70km from east to west. It also features some rather long lines between neighbouring stations, for example, *Castro Valley* to *Dublin/Pleasanton* is 16km. Not all the stations are as far apart though. The distance from *Oakland City Center/12th St* to *19th St/Oakland* is only 550m. However, the official published map (shown in Figure 6.23) still uses a geographic layout style, as can be seen when compared with the geographic map (shown in Figure 6.22). The use of a geographic map for a network that covers such a large area means that some stations are drawn very far apart while others are drawn very close together—the lines between some pairs of neighbouring stations are barely discernable. The density of stations in some areas of the map means that some labels cannot be positioned immediately next to the nodes. This is particularly prevalent in the Oakland/Berkeley area where three labels are positioned some way from their nodes and linked with a line. Characteristics of the San Francisco BART map are given in Table 6.10.

The version of the San Francisco BART map drawn using our method is shown in Figure 6.24 and uses the criteria weightings given in Table 6.11.

Our final map shows regular node spacing and all except one edge has been drawn octilinearly (Figure 6.21(a)). Labelling is also good, with only one label occluding edges (Figure 6.21(a)). The meandering of the lines from *West Oakland* to *Daly City* (Figure 6.21(b)) is curious and may have been caused when the nodes were aligned with grid intersections and there was not enough space on the grid for the nodes. The area around *Oakland City Center/12th St* also suffers from a sharp angle in one of the lines that pass through that node (Figure 6.21(a)).

(a)   (b)
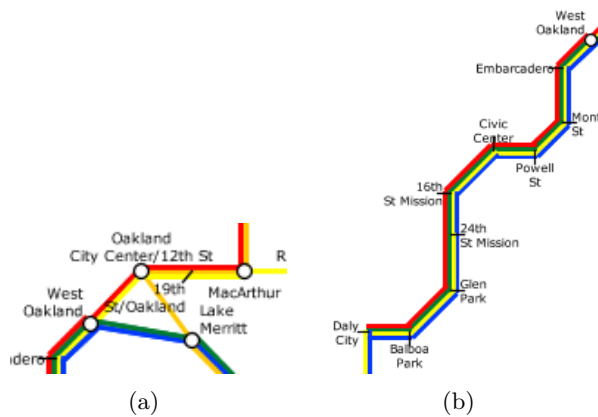
Figure 6.21: Specific referenced examples in the automatically-drawn San Francisco map.

Table 6.10: Characteristics of the San Francisco map.

Stations	43
Interchange Stations	6
Edges	94
Lines	5
Faces	1

Table 6.11: Criteria weightings and parameters used for the San Francisco map.

Node movement		Labelling		Other Parameters	
w_{N1}	30000.0	w_{L1}	300.0	Iterations	7
w_{N2}	50.0	w_{L2}	80.0	Pref. grid spacing l	4
w_{N3}	45.0	w_{L3}	19.0	Grid Spacing g	50
w_{N4}	1000000.0	w_{L4}	15.0	Min. Cluster Distance	3
w_{N5}	220.0	w_{L5}	3.0	Max. Node Movement	8
w_{N6}	9250.0	w_{L6}	1.0		
		w_{L7}	30.0		
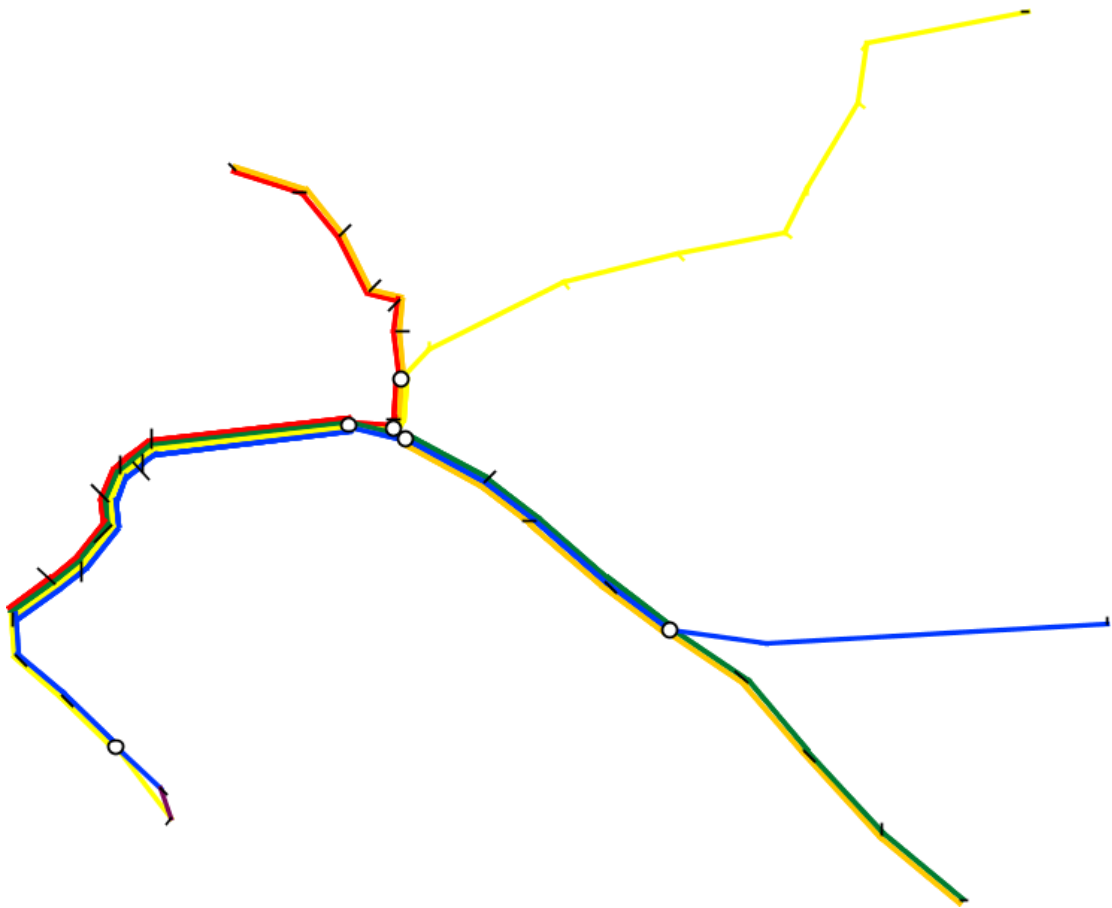
Figure 6.22: San Francisco BART map: geographic layout.
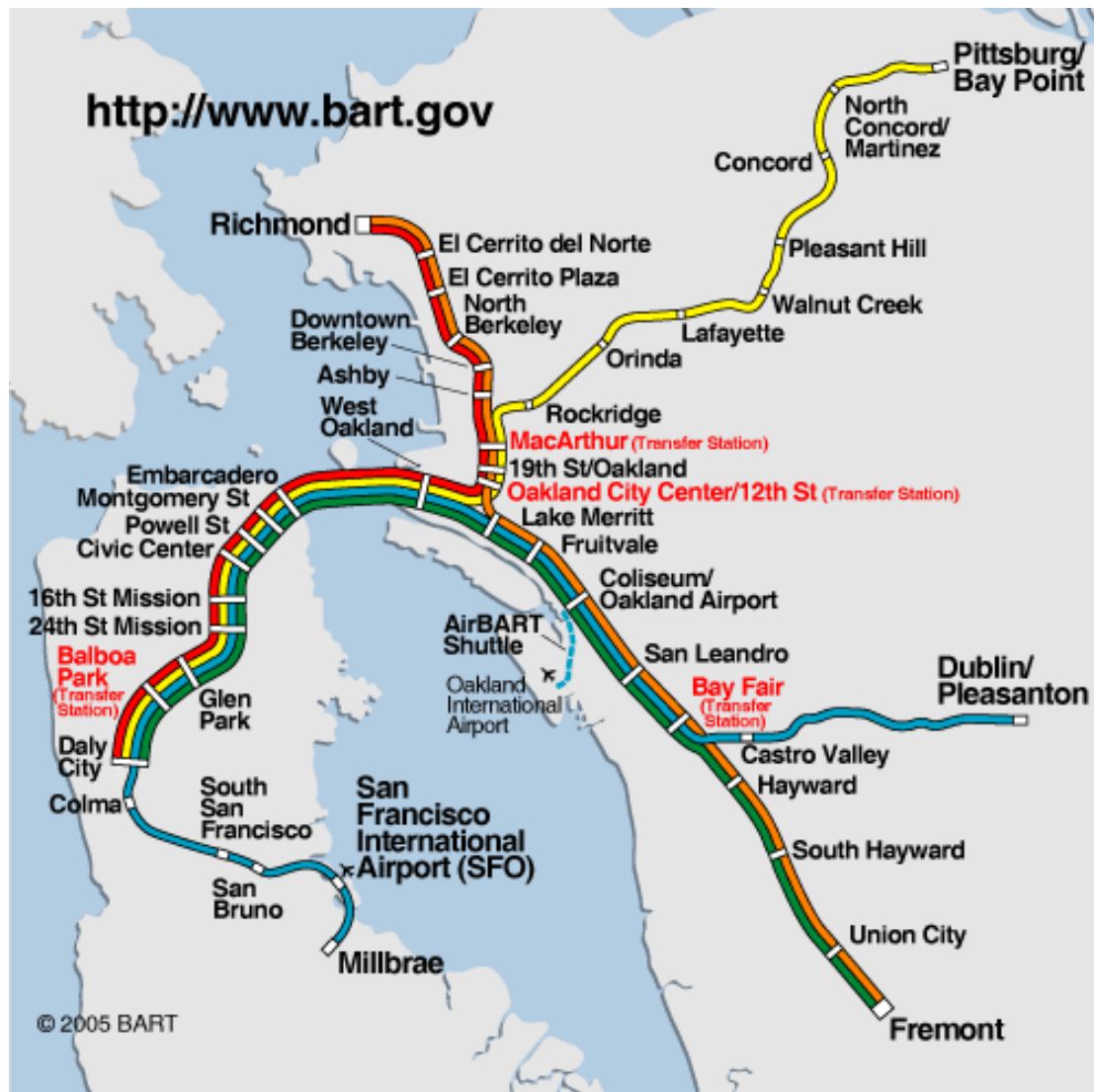
Figure 6.23: San Francisco BART map: official layout.
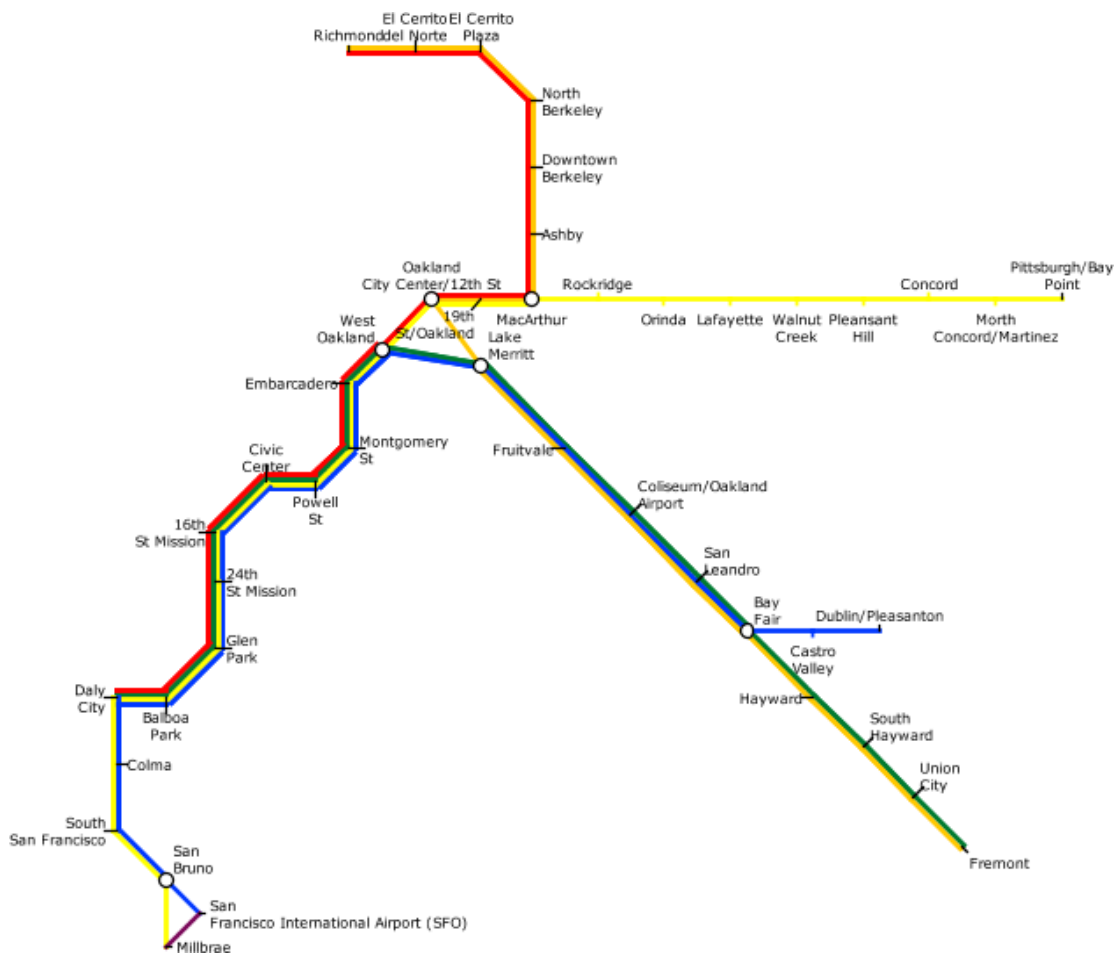
Figure 6.24: San Francisco BART map: our layout.
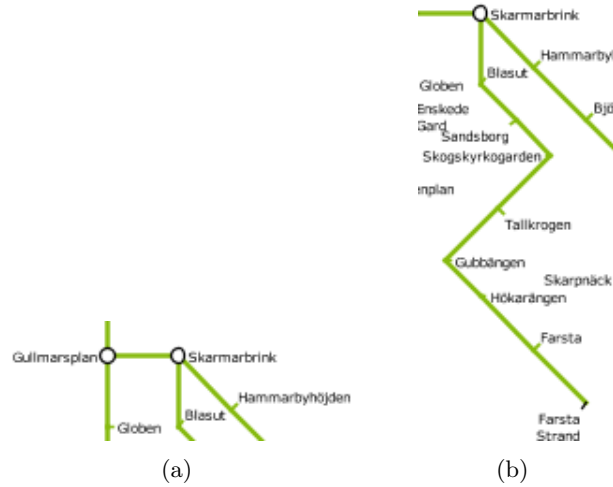
Figure 6.25: Specific referenced examples in the automatically-drawn Stockholm map.

6.6   Stockholm

The Stockholm map [SL008] is a larger version of a centralised radial map. The official map (shown in Figure 6.27) is characterised by the starburst effect with all lines radiating from the *T-Centralen* station. Characteristics of the Stockholm map are given in Table 6.12. As with other radial maps, there are very few faces—only two in this case. The geographic sketch of the Stockholm map is shown in Figure 6.26.

The version of the Stockholm map drawn using our method is shown in Figure 6.28 and uses the criteria weightings given in Table 6.13.

Our map retains the characteristic radial features of the official map but also retains some of the geographic topography (particularly the way the two lines on the left-hand side of the map meander). Two areas of our map are more compressed than would be ideal where two lines run almost parallel to each other. The junction of the lines at the bottom-right area of the map suffers from an over-dependence on the angular resolution criterion, particularly around *Gullmansplan* and *Skarmarbrink* (Figure 6.25(a))—this area also highlights how the line straightness criterion breaks down for branches in lines. The line from *Skarmarbrink* to *Farsta Strand* in the south-east of the map has a curious zig-zag form (Figure 6.25(b)), mainly due to having to fit in around the labels that had already been positioned along the two adjacent lines. The labelling is generally of good quality but occlusions are present mainly where two lines have been drawn too close together.
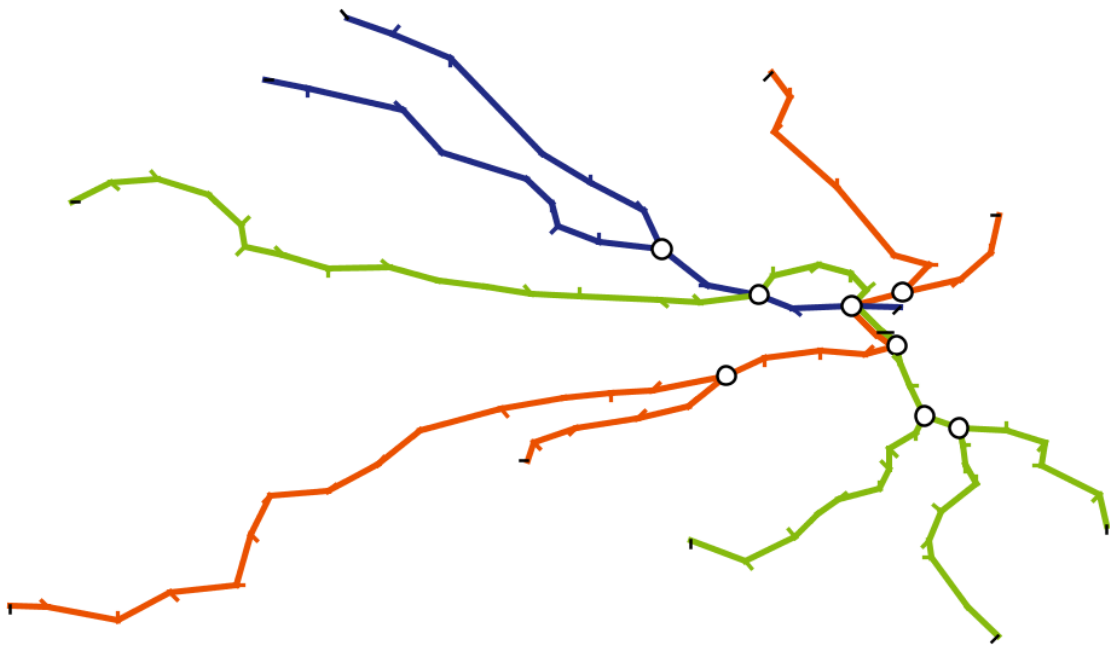
Figure 6.26: Stockholm map: geographic layout.

Figure 6.27: Stockholm map: official layout.

Table 6.12: Characteristics of the Stockholm map.

Stations	100
Interchange Stations	9
Edges	101
Lines	3
Faces	2

Table 6.13: Criteria weightings and parameters used for the Stockholm map.

Node movement		Labelling		Other Parameters	
w_{N1}	30000.0	w_{L1}	300.0	Iterations	6
w_{N2}	50.0	w_{L2}	80.0	Pref. grid spacing l	4
w_{N3}	45.0	w_{L3}	19.0	Grid Spacing g	30
w_{N4}	1000000.0	w_{L4}	15.0	Min. Cluster Distance	3
w_{N5}	220.0	w_{L5}	3.0	Max. Node Movement	8
w_{N6}	9250.0	w_{L6}	1.0		
		w_{L7}	30.0		

Figure 6.28: Stockholm map: our layout.

6.7   Sydney

The Sydney CityRail [SCR07] is a very large network covering a area of approximately 3600km^2 of metropolitan Sydney. The map makes use of most of the usual metro map design characteristics, including horizontal, vertical and 45°-diagonal lines, scale generalisation and line generalisation. The use of scale generalisation is very prominent in the central Sydney area where most of the lines converge in a tight loop around the city centre. Long horizontal lines have forced the use of diagonal labels, but all diagonal labels are of the same orientation. The official Sydney CityRail map is shown in Figure 6.31 and the geographic map with stations positioned according to their latitude and longitude is shown in Figure 6.30. Characteristics of the Sydney map are shown in Table 6.14. We have constrained that area of interest in the Sydney map to the main metropolitan area of Sydney, bounded by *Berowra* in the north, *Emu Plains* in the west and *Macarthur* and *Waterfall* in the south.

The version of the Sydney CityRail map drawn using our method is shown in Figure 6.32 and uses the criteria weightings given in Table 6.15.

The finished version of our map has succeeded in evening out node spacing and nearly all the edges are drawn octilinearly. Labelling is also of a good quality. One particular area posing a problem for our method is the central area at the right-hand side of the map (Figure 6.29(a)). This section has up the seven lines passing through each station and features a very tight loop. The published map handles this area by significantly increasing the scale (possibly one of the most dramatic uses of scale generalisation in any of the maps presented in this chapter), but our method does not explicitly handle scale generalisation for such a small area of the map. A few edges are not drawn octilinearly, most notably between *Glenfield* and *Wolf Creek* (the bottom-most horizontal line in the map, Figure 6.29(b)). In this case, a local minimum has been reached where none of the clustering algorithms will find the right cluster of nodes as the length of some of the edges is greater than the minimum cluster distance.

We can compare our result with those of Hong *et al.* (Figure 6.33), Nöllenburg (Figure 6.34) and Merrick and Gudmundsson (Figure 6.35). Our result is most similar to Nöllenburg's, but the lines in our map are not as straight due to the use of the 3-link heuristic in Nöllenburg's method. However, our automatically-drawn map incorporated labelling as part of the method meaning that enough space was left for station labels. While it would be possible to achieve a reasonable labelling of Nöllenburg's map, there are areas where this may not be possible without causing label occlusions. Hong's map includes labelling and the use of diagonal labels helps significantly, particularly with long horizontal lines. The dense area around the centre of Sydney (where most of the lines converge) is handled well by Hong and Nöllenburg but less well by our method and Merrick and Gudmundsson's method.

(a) (b)
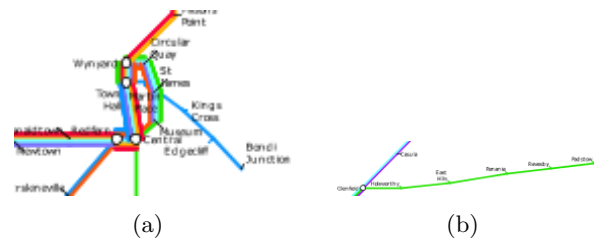
Figure 6.29: Specific referenced examples in the automatically-drawn Sydney map.
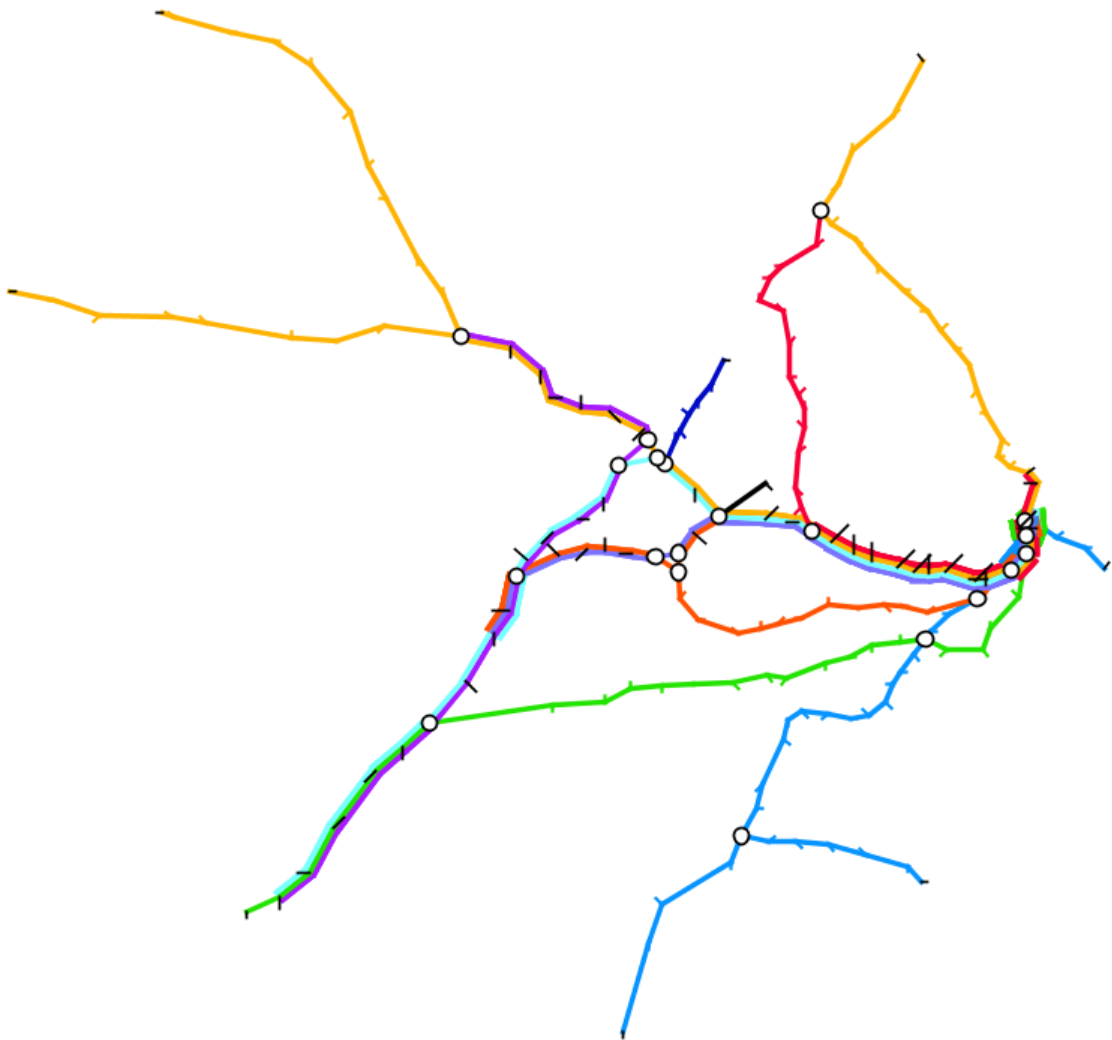
Figure 6.30: Sydney CityRail map: geographic layout.
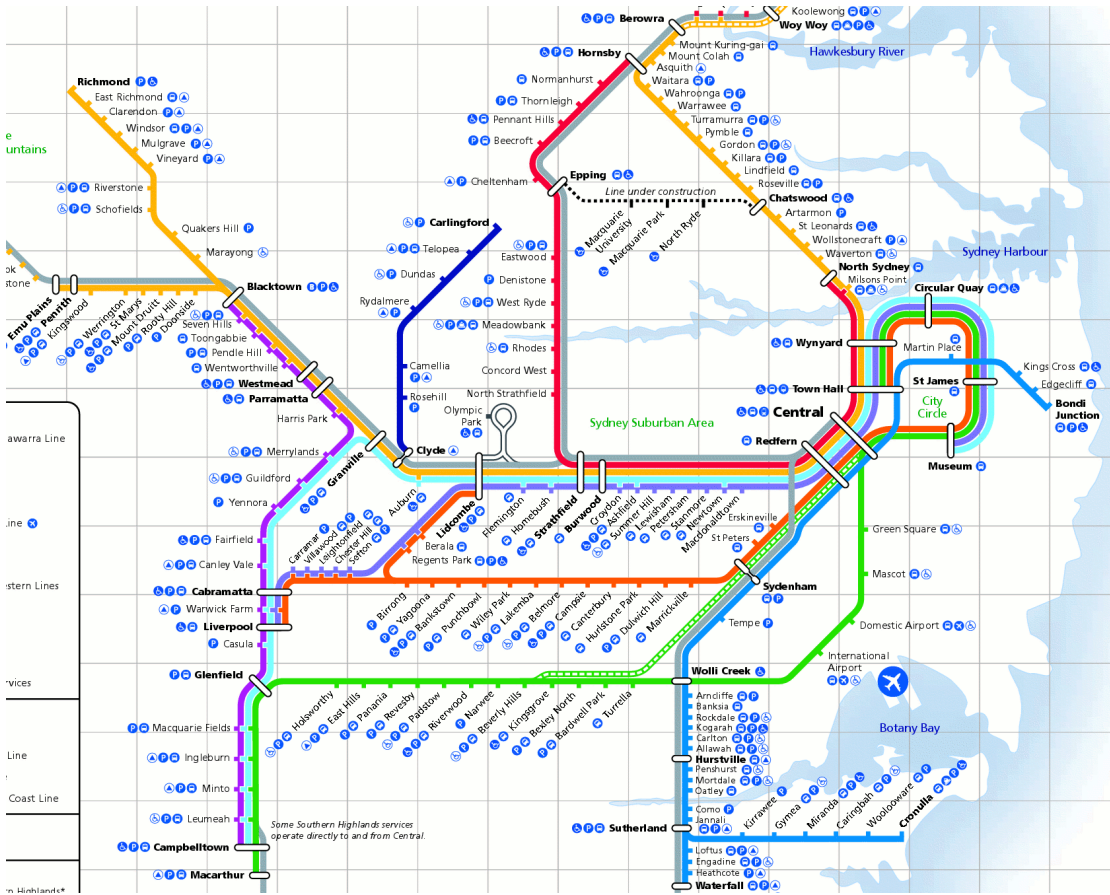
Figure 6.31: Sydney CityRail map: official layout.
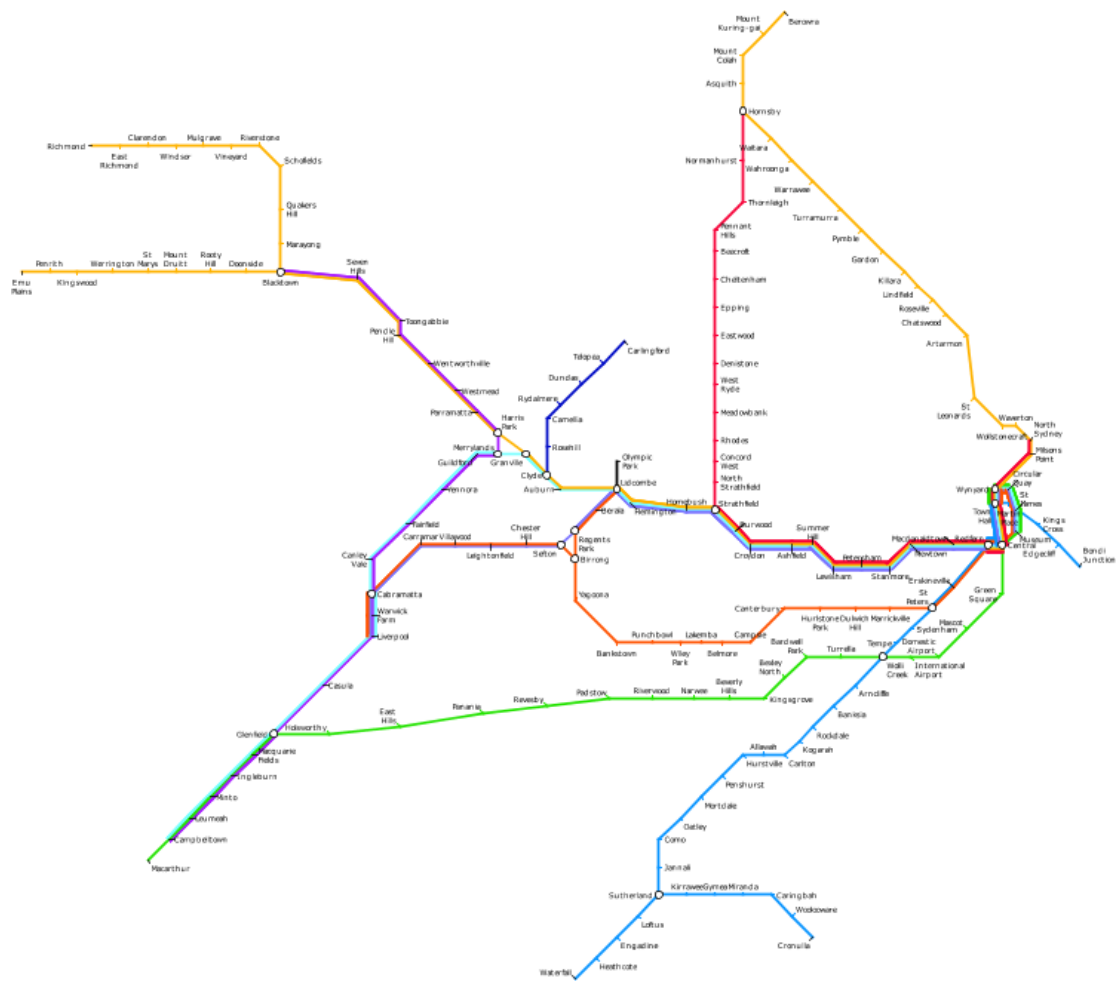
Figure 6.32: Sydney CityRail map: our layout.
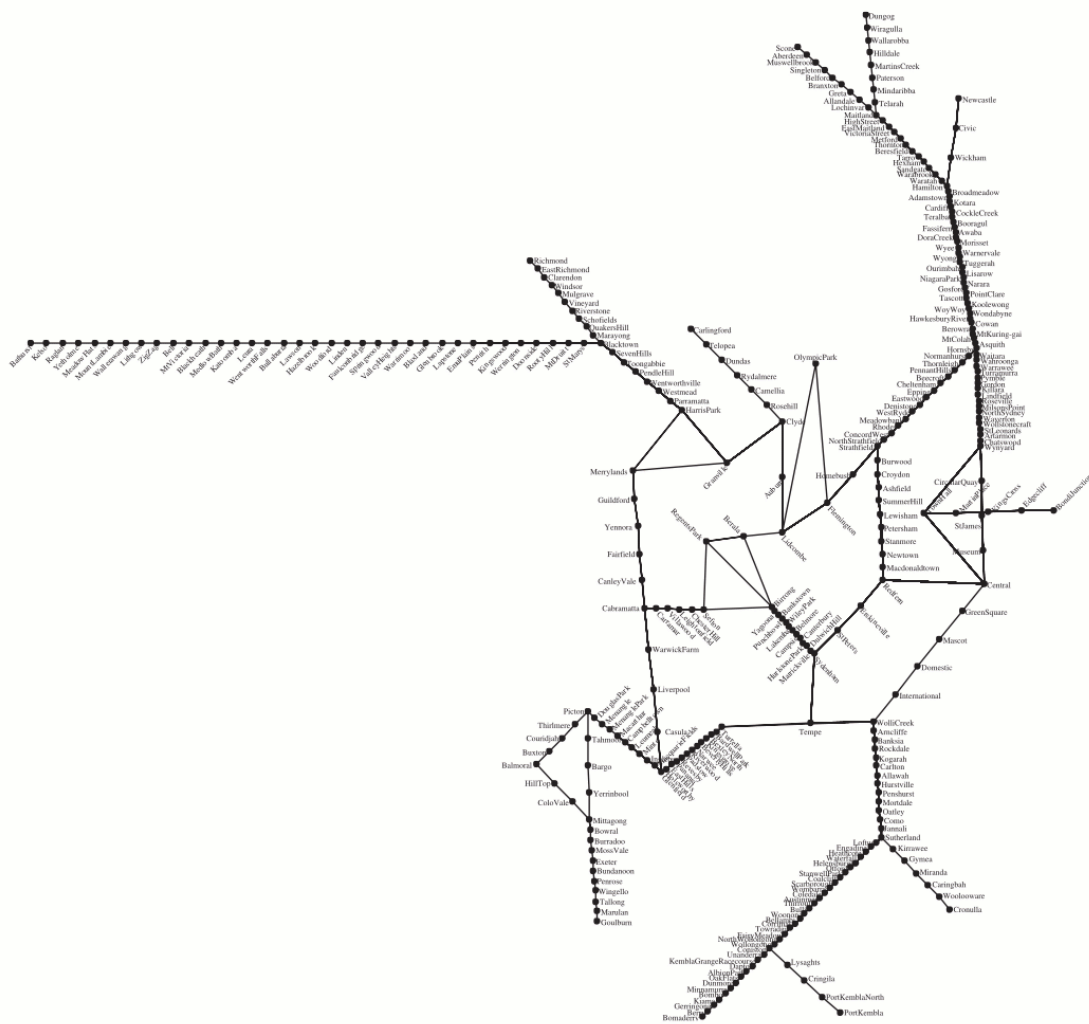
Figure 6.33: Sydney CityRail map: as drawn by Hong *et al.* [HMdN06].
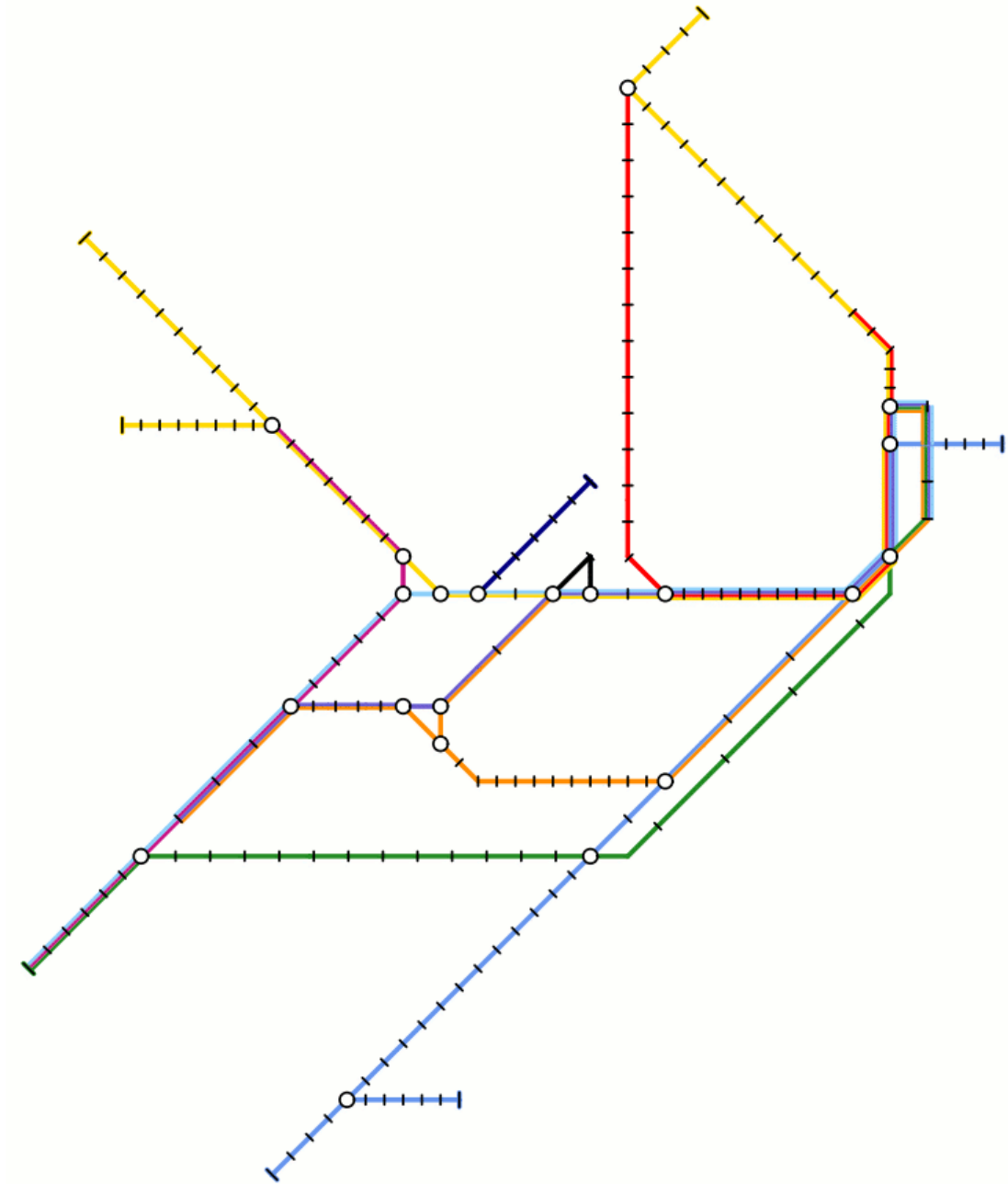
Figure 6.34: Sydney CityRail map: as drawn by Nöllenburg [Nöl05].
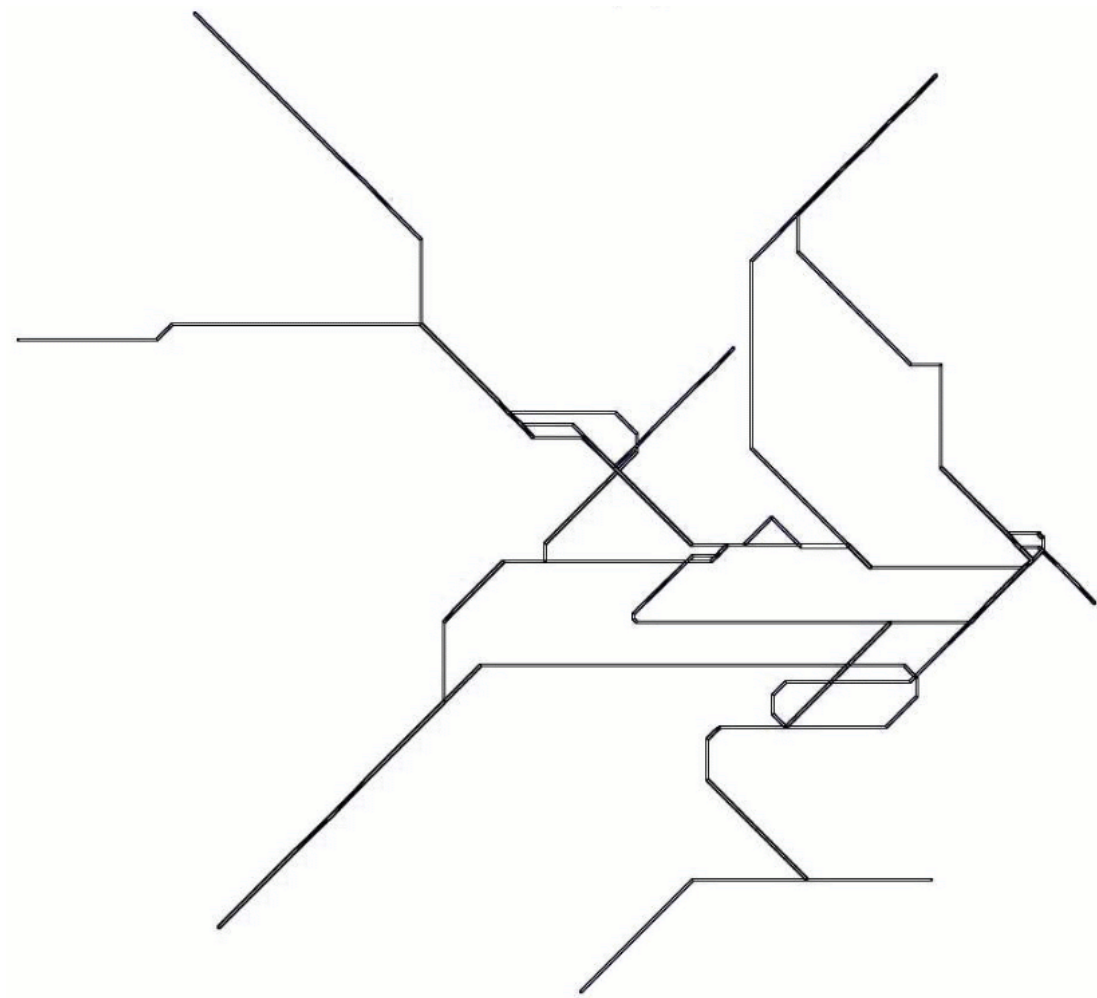
Figure 6.35: Sydney CityRail: as drawn by Merrick and Gudmundsson [MG07].

Table 6.14: Characteristics of the Sydney map.

Stations	173
Interchange Stations	19
Edges	284
Lines	10
Faces	8

Table 6.15: Criteria weightings and parameters used for the Sydney CityRail map.

Node movement		Labelling		Other Parameters	
w_{N1}	30000.0	w_{L1}	300.0	Iterations	10
w_{N2}	50.0	w_{L2}	80.0	Pref. grid spacing l	4
w_{N3}	45.0	w_{L3}	19.0	Grid Spacing g	15
w_{N4}	1000000.0	w_{L4}	15.0	Min. Cluster Distance	3
w_{N5}	220.0	w_{L5}	3.0	Max. Node Movement	8
w_{N6}	9250.0	w_{L6}	1.0		
		w_{L7}	30.0		

(a)   (b)
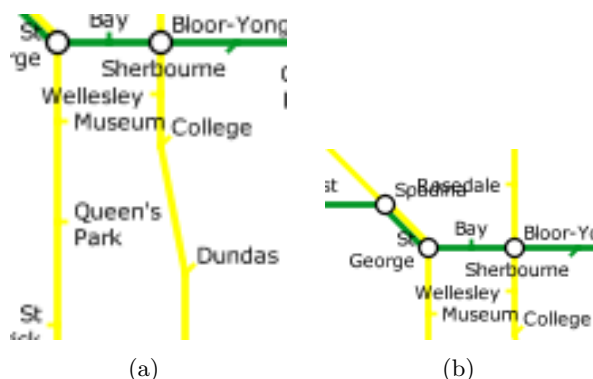
Figure 6.36: Specific referenced examples in the automatically-drawn Toronto map.

6.8   Toronto

The Toronto Transit Commission Subway map [TTC08] is a simple radial map. The characteristics of the Toronto map are given in Table 6.16. The most prominent feature of this map is the U-shaped line that passes through the centre and loops back again. The official published map is shown in Figure 6.38—the horizontal nature of the map comes about from being used in the train carriages. This causes the spacing between stations on vertical lines to be much less when compared with the spacing between stations on horizontal lines. The use of horizontal lines and horizontal labels means that the labels have to alternate from one side of the line to the other for each station. The geographic sketch of the Toronto subway map is shown in Figure 6.37.

The version of the Toronto map drawn using our method is shown in Figure 6.39 and uses the criteria weightings given in Table 6.17.

The map drawn with our automatic layout method emphasises the strong horizontal line from *Kipling* to *Woodbine* with just one change of direction. All except one edge is drawn octilinearly (Figure 6.36(a)). Labelling is generally good but there a number of occlusions in the central area of the map (Figure 6.36(a)). Some of the labels along the horizontal line alternate from one side to the other where there is not enough space, but this is not consistent. The only real problem with the map is not leaving enough space between both arms of the yellow looping line—this is particularly noticeable where the line loops at the bottom of the map (see Figure 6.36(a)).

Table 6.16: Characteristics of the Toronto map.

Stations	70
Interchange Stations	5
Edges	70
Lines	4
Faces	2

Table 6.17: Criteria weightings and parameters used for the Toronto map.

Node movement		Labelling		Other Parameters	
w_{N1}	30000.0	w_{L1}	300.0	Iterations	5
w_{N2}	50.0	w_{L2}	80.0	Pref. grid spacing l	4
w_{N3}	45.0	w_{L3}	19.0	Grid Spacing g	50
w_{N4}	1000000.0	w_{L4}	15.0	Min. Cluster Distance	3
w_{N5}	220.0	w_{L5}	3.0	Max. Node Movement	8
w_{N6}	9250.0	w_{L6}	1.0		
		w_{L7}	30.0		

Figure 6.37: Toronto map: geographic layout.

Figure 6.38: Toronto map: official layout.
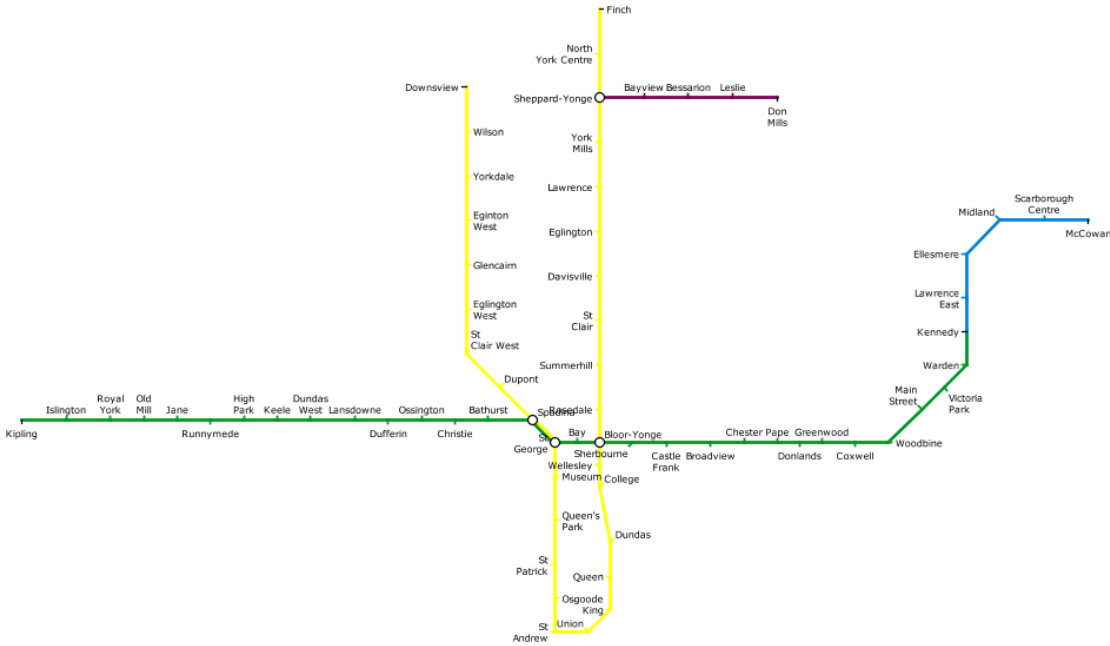
Figure 6.39: Toronto map: our layout.

6.9   Washington D.C.

The Washington Metropolitan Area Transit Authority Metro System Map [WMA08] is from one of the newer metro systems in the world. The official published map is shown in Figure 6.42. The map is characterised by five lines radiating from a relatively compact central area. Characteristics for the Washington map are given in Table 6.18. The official map has been influenced by modern design concepts with very broad lines overlaid on topographic metadata showing main roads, the river and parkland. The map uses a combination of horizontal and two orientations of diagonal labels. There are many examples of cases where labels are allowed to overlap the edges. The text of the labels is also relatively small when compared with the symbol representing the station and the thickness of the line. In some cases, the size of the device used to represent stations causes there to be barely any space between adjacent stations. The geographic map of the Washington Metro with stations positioned according to their latitudes and longitudes is shown in Figure 6.41.

The version of the Washington map drawn using our method is shown in Figure 6.43 and uses the criteria weightings given in Table 6.19.

Our finished layout shows a map with very regular node spacing along lines and all edges have been drawn octilinearly. The line straightness criterion has clearly had an effect, particularly on the green, yellow and red lines. The central area of the map is still slightly too compressed and the edge crossing between *Metro Center* and *Farragut North* has caused an example of poor angular resolution between the red line and the blue/orange lines (Figure 6.40(a)). Labelling is generally good, but in this case a significant number of labels suffer from occlusions, particularly in the centre of the map (Figure 6.40(b)). This is partially due to the labels being very long and partially due to there not being enough space to position the labels ideally. It might be that a modification of the criteria or other parameters might have found a more optimal solution that allows for a better labelling. The blue and orange lines are not drawn as straight as they might be—in one case (around *Metro Center*) this is due to the aforementioned edge crossing, and in another case (around *Federal Center SW* and *Capitol South*, Figure 6.40(b)) this is due to a local minimum caused by the position of the label for *L'Enfant Plaza* and the line to the right not being on quite the same horizontal level as *L'Enfant Plaza*.
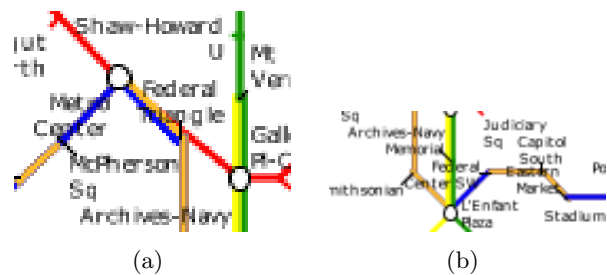
Figure 6.40: Specific referenced examples in the automatically-drawn Washington D.C. map.

Table 6.18: Characteristics of the Washington D.C. map.

Stations	86
Interchange Stations	9
Edges	108
Lines	5
Faces	5

Table 6.19: Criteria weightings and parameters used for the Washington D.C. map.

Node movement		Labelling		Other Parameters	
w_{N1}	30000.0	w_{L1}	300.0	Iterations	20
w_{N2}	50.0	w_{L2}	80.0	Pref. grid spacing l	4
w_{N3}	45.0	w_{L3}	19.0	Grid Spacing g	20
w_{N4}	1000000.0	w_{L4}	15.0	Min. Cluster Distance	3
w_{N5}	220.0	w_{L5}	3.0	Max. Node Movement	8
w_{N6}	15250.0	w_{L6}	1.0		
		w_{L7}	30.0		
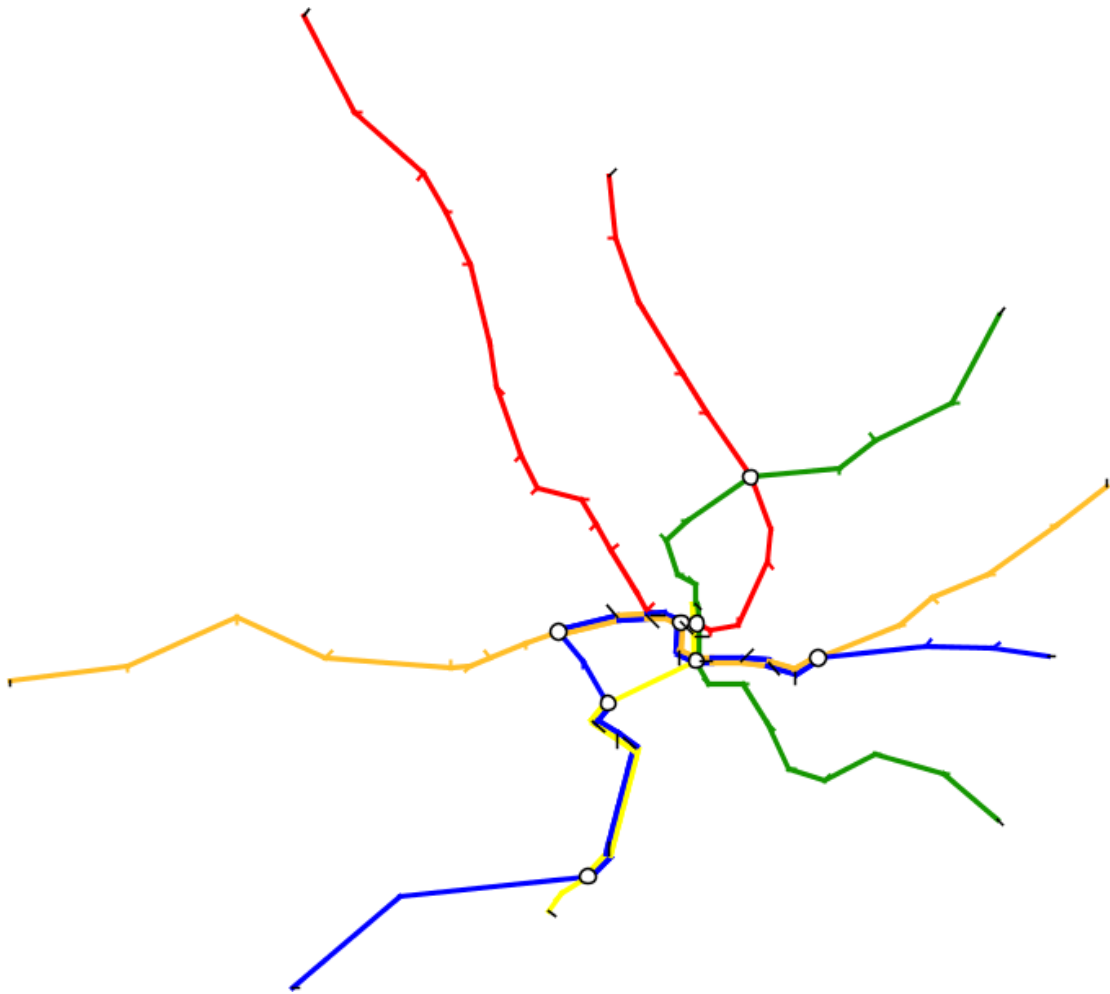
Figure 6.41: Washington D.C. map: geographic layout.

Figure 6.42: Washington D.C. map: official layout.

Figure 6.43: Washington D.C. map: our layout.

Chapter 7

Empirical Evaluation

7.1 Introduction

In the previous chapters, a number of aesthetic criteria for automatically drawing metro maps were detailed, together with a method that combines these criteria in order to produce some automatically drawn metro maps. To show that the maps produced using this method are suitable, we devised an empirical experiment involving a number of human subjects performing route-planning tasks with a variety of different metro maps. In this experiment we measured the time and accuracy of finding optimal routes as well as the preference for each metro map.

7.2 Hypotheses

As introduced in Section 2.2, 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 does not 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 instead we intended to measure the quality indirectly by asking human participants to perform common route-planning tasks and measuring their speed and accuracy.

We have four hypotheses:

Hypothesis 1. A map of a metro system drawn with our automated software is better for finding an optimal route than a geographic map of the system.

Hypothesis 2. A map of a metro system drawn with our automated software is better for finding an optimal route than the official published map of the system.

Hypothesis 3. A map of a metro system drawn with our automated software is preferred over a geographical map of the system.

Hypothesis 4. A map of a metro system drawn with our automated software is preferred over the official published map of the system.

7.3   Design

In order to be able to test these hypotheses, we devised methods that would 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 elicit 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 you go through to get from *Station X* to *Station Y*?

These questions were chosen on the basis that they are likely to be common route-planning tasks when people use metro maps and because they generate easily quantifiable answers. This is in contrast to questions which are not easily quantifiable, such as "what is the shortest path from *Station X* to *Station Y*" or "which stations do you need to change at to get from *Station X* to *Station Y*". A selection of five potential answers was selected for each question. These answers were a sequence of five consecutive integers that included the actual answer. The precise range of the potential answers was arbitrary but some care was taken to ensure that the actual answers were evenly distributed.

Pairs of stations to use for individual questions were chosen to ensure that questions were not trivial. Care was also taken to ensure that there was no ambiguity in the question (to avoid situations such as there being two or more obvious potential routes between the stations in question). During a trial session (see Section 7.4) we were able to refine any questions that we had selected.

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

1. Atlanta Metropolitan Area Rapid Authority (MARTA) Rail Map [MAR08].

2. Bucharest Metro Map [Met08].

3. Mexico City *Sistema de Transporte Colectivo* Map [Mex08].

4. Stockholm *Tunnelbana* Map [SL008].

5. Toronto Transit Commission (TTC) Subway/RT Route Map [TTC08].

6. Washington Metropolitan Area Transit Authority (WMATA) Metro System Map [WMA08].

Table 7.1: Characteristics of metro maps used in the empirical evaluation. The Recife map was used for training questions.

Map	Stations	Lines	Interchange Stations	Edges	Faces
Atlanta	39	2	3	36	1
Bucharest	45	3	6	45	3
Mexico City	175	11	24	165	19
Stockholm	100	3	9	101	2
Toronto	70	4	5	70	2
Washington D.C.	86	5	9	108	5
Recife	28	2	2	28	1

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 7.1 gives details of the characteristics in terms of number of stations (nodes), lines, interchange stations, edges and faces 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 automatic layout method described in Chapters 3 and 4. We also used the Recife metro map [Rec08] 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. The fonts, line styles and icons used to represent stations and labelling all vary from one map to another. Also, some maps incorporate topographic metadata while others do not. These differences can be seen by looking at the examples in Section 2.2. We decided that all three variants of each map would be drawn using the same graphic design to remove any criticism that features such as stations and lines are drawn could make a difference to the subjects' route planning tasks. To illustrate the three map variants, the geographic map (Figure 7.1), normalised published map (Figure 7.2) and automatically drawn map (Figure 7.3) for the Mexico City metro are shown here. The official *Mapa de la Red* metro map for Mexico City is shown in Figure 7.4. All 18 maps as used in the evaluation are shown in Appendix A: Section A.1.1 (Atlanta), Section A.1.2 (Bucharest), Section A.1.3 (Mexico City), Section A.1.4 (Stockholm), Section A.1.5 (Toronto) and Section A.1.6 (Washington D.C.).
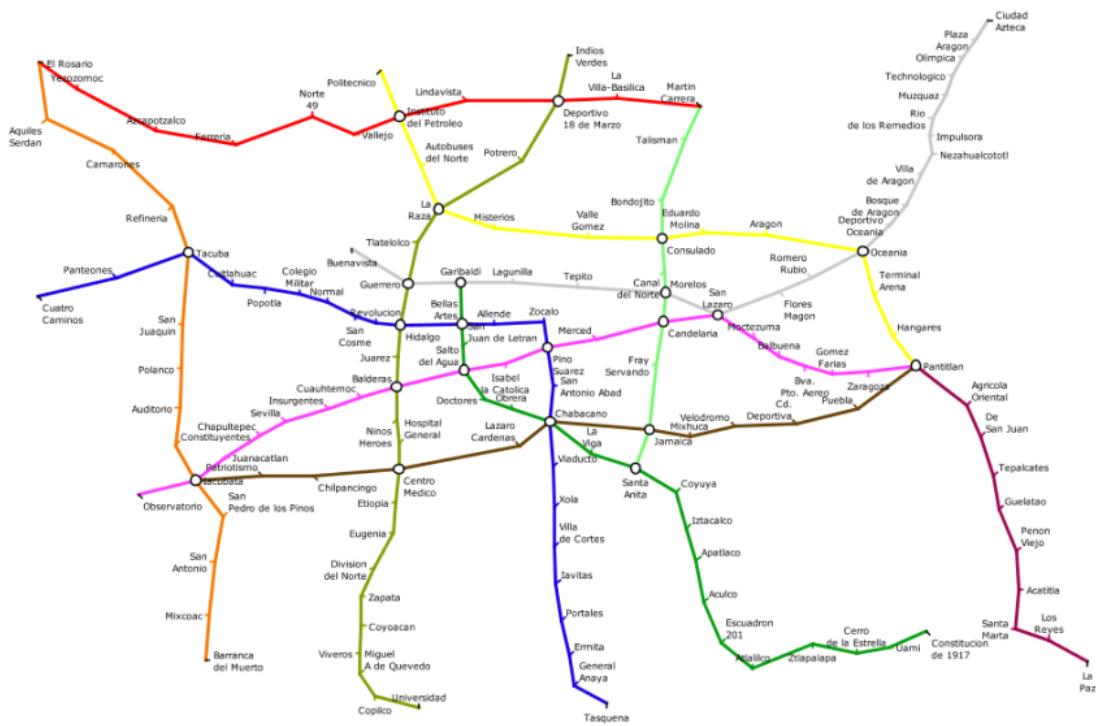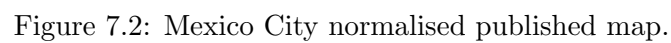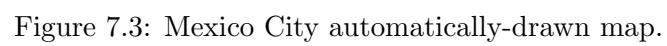
Figure 7.1: Mexico City geographic map.

Figure 7.2: Mexico City normalised published map.

Figure 7.3: Mexico City automatically-drawn map.
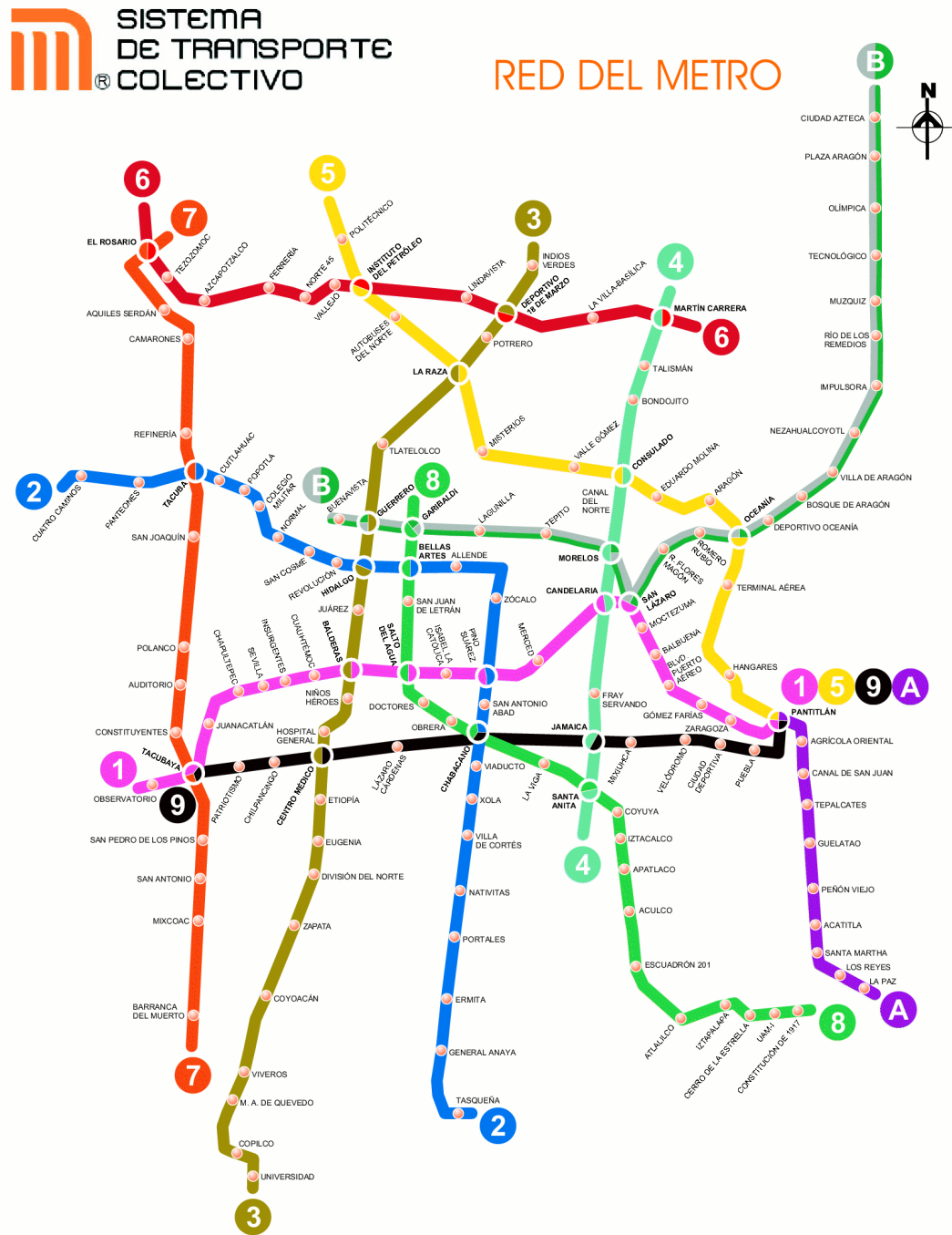
Figure 7.4: Mexico City *Mapa de la Red* official metro map.

Table 7.2: Number of subjects in each experiment group.

Group	Number of subjects
A	13
B	15
C	15

7.4   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 from the University of Kent. As a number of experiment sessions 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 7.2.

Each group received exactly the same questions in exactly the same order but they only saw one variant of each map per group for each question. For example, for a question 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 possibility of learning the map between questions—while not completely eliminated—could 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 us both to time each question in the experiment accurately, and to see automatically whether they answered correctly or incorrectly and to collate the results. It also allowed the environment to be more closely controlled and ensured uniformity in the way that the test was conducted for all subjects. A screenshot of the software application is shown in Figure 7.5.

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 group's questions.

2. A preliminary script was read out. This script introduced the subjects to the information that they needed in order to understand the questions asked of them in
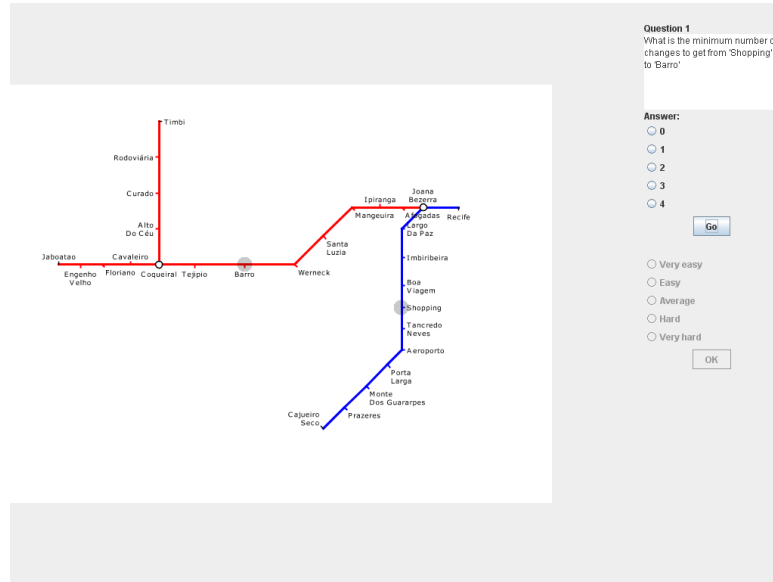
Figure 7.5: Screenshot of the software application used to conduct the empirical evaluation. The map is shown in the left-hand part of the screen with the two stations in the question highlighted. The question and answer options are shown in the right-hand part of the screen.

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 or 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 through by demonstrating the application to the subjects using an overhead projector at the front of the room. 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 explained which parts of each question were to be timed. The full script is shown in Appendix A, Section A.2.1.

3. The subjects were told when they could commence using the software application. Everyone started at the same time and were presented with 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 or disturbance 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 dummy questions until the 20 minutes was

up. These dummy questions were discarded prior to analysis of the results. 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 potential 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 can continue.

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

(d) The subject then selects their perception of the difficulty level of the task from a list of five options from "very hard" to "very easy". The subjects are 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 that showed them how many of their answers were correct and 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 part 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 part 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 completing the tests?

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

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

Prior to the real experiment sessions, a trial session was used to determine any problems in our methodology. During this trial session we were able both to find how much time would be appropriate for the number of questions we were asking and to uncover any ambiguous or impossible questions. The scripts were also refined as a result of this trial session. The results from the trial session were discarded.

7.5   Results and Analysis

This section describes how we analysed the data that we collected from our empirical experiment. The raw data is included in Appendix A. We were interested in measuring the efficiency of the maps with regards to the time that it took to find a solution to a number of questions. Sections 7.5.1 through 7.5.4 detail how we used frailty models to analyse the route-finding tasks. Section 7.5.5 explains our analysis of the preference tasks and Section 7.5.6 looks at any interesting comments made in the questionnaire.

7.5.1   Statistical Models for Analysing Route-Finding Tasks

When designing the format of the empirical experiment, we had to consider the method by which we were to analyse the data afterwards. We considered three options for collecting and analysing the data:

1. Group the subjects into three groups and only present questions involving one variant of each map to each group. For example, group A would only see the geographical maps, group B would see the official published maps and group C would see the automatically-drawn maps. This has the advantage that a simple between-subjects analysis using analysis of variance (ANOVA) could be used.

2. Do not group any subjects but present all three map variants for each question to each subject. This would have allowed a within-subjects analysis of the data using either ANOVA or a *t*-test. However, this method would have required asking the same question three times to each subject (once for each map variant). The possible effect of subjects remembering the maps, questions and answers might have significantly skewed the results and could not be ignored.

3. Group the subjects into three groups and, for each question, present a different variant of each map to each group. This is effectively the same as a combined between- and within-subjects design and was the method that we used for our experiment. This method has the advantage that any learning of the maps and questions is minimised and should produce more meaningful results. However, the analysis of the data is more complex and requires the use of frailty models to handle the effects of multivariate data between groups.

We decided that the most appropriate analysis was the third one and we were able to analyse the results using time-to-event models (also called *named duration* or *survival models*). The variable of relevance was the time for a subject to complete a task correctly. In particular, we study the duration time to complete each task using proportional hazards models to distinguish stratification effects for using one of the map types while considering the effects of map locations. We also used frailty models to measure

random effects across subjects, since our data consists of more than one sample for each subject. Hence, the effect on the duration time to answer a task correctly using different map types is measured through its corresponding factor effect. As we are considering the time to complete a task correctly, we can also consider incorrect answers as right-censored data. These models are explained in more detail in the following section.

7.5.2 Survival and Frailty Models

For our analysis, we used frailty models (introduced by Clayton and Cuzick [CC85, The99, TGP03]) which are a generalisation of Cox's proportional hazards model [Cox72, CO84, EJJ80]. Survival models [Hou00] are more often used for analysing the survival of biological organisms or the failure of mechanical systems, but their use can also be extended to the study of duration data. The idea of survival analysis is to study time-to-event data (for example the survival time or the time taken to complete a task). For our experimental data we are modelling the duration time, T, taken by an individual to correctly answer a question.

Typically, survival models are used to analyse the results of subjects undergoing some form of time-to-event test. Each subject in the study contributes towards learning about the survival function for an underlying population that characterises the time-to-event to complete a certain task with a specific result. As mentioned before, proportional hazards models are used to differentiate the multiplicative effect of different stratification factors in the underlying survival function. The frailty component extends the proportional hazards model with the inclusion of an additional multiplicative random effect allowing the study of clustering patterns across observations. In our study, the multiplicative factors are defined by dummy variables and observations are clustered by individuals.

During the study there were some individuals who did not respond with correct answers to some of the questions. We made the assumption that all these cases would eventually be answered correctly if the individual had enough time. Therefore, these cases were considered as truncated (right-censored) duration times. In this way, the information contained in these cases can also be included in the analysis.

The mathematical description of survival models with proportional hazards and frailties is as follows. Let T be the time to complete a task correctly. Then assume that T is a random variable having a cumulative distribution function

$$P(t) = \Pr(T \leq t) \tag{7.1}$$

and probability density function

$$p(t) = \frac{d}{dt}P(T) \tag{7.2}$$

where t is time. The survival function is then defined as the complement of P, i.e.,

$$
\begin{aligned}
S(t) &= 1 - P(t) \\
&= \Pr(T > t). \tag{7.3}
\end{aligned}
$$

In other words, the survival function $S(t)$ gives the probability that an individual would have to spend more time than t to complete the given task and that at time t, he/she had not completed the task. The time that a question is answered can also be represented by the hazard function, $h(t)$, which gives the instantaneous rate of failure at time t on condition that something survives until at least t. The hazard function is defined by

$$
\begin{aligned}
h(t) &= \lim_{\Delta t \to 0} \frac{\Pr\left(T \in [t, t + \Delta t) \mid T \geq t\right)}{\Delta t} \\
&= \frac{p(t)}{S(t)}. \tag{7.4}
\end{aligned}
$$

Equivalently, the density function is completely defined in terms of its corresponding hazard and survival function via

$$
p(t) = h(t)S(t). \tag{7.5}
$$

As we mentioned earlier, proportional hazards models with frailties incorporate multiplicative effects given for stratified and clustered observations. Its analytic expression is defined by

$$
h\left(t_{ij} \mid \boldsymbol{x}_{ij}, v_i\right) = h_0\left(t_{ij}\right) \exp\left(\boldsymbol{x}_{ij}' \boldsymbol{\beta}\right) v_i \tag{7.6}
$$

where i indexes the group of individuals, j indexes individuals, $\boldsymbol{x}_{ij}$ is a p-dimensional covariate vector for stratification, $\boldsymbol{\beta}$ is a p-dimensional unknown parameter vector, h_0 is the baseline hazard function, common to all individuals in the study, and v_i is a multiplicative random effect for each cluster (i.e., each v is an independent random variable but with common distribution functions). In our model, $\boldsymbol{x}_{ij}$ is defined as a vector containing seven dichotomous variables. Two of these variables are used to evaluate the effect of the published and automatically-drawn map types. The other five variables measure the effect across the six different maps. The multiplicative variable is usually reparameterised as

$$
\theta_i = \log v_i. \tag{7.7}
$$

Whence θ_i can then be artificially incorporated in the linear proportional hazard, i.e.,

$$
h\left(t_{ij} \mid \boldsymbol{x}_{ij}, v_i\right) = h_0\left(t_{ij}\right) \exp\left(\boldsymbol{x}_{ij}' \boldsymbol{\beta} + \theta_i\right), \tag{7.8}
$$

with

$$\boldsymbol{x}_{ij}^{'}\boldsymbol{\beta} = \beta_{\text{pub}}\mathbf{1}_{\text{pub}} + \beta_{\text{auto}}\mathbf{1}_{\text{auto}} + \sum_{k=2}^{6}\beta_k\mathbf{1}_{\text{map }k}, \tag{7.9}$$

where $\mathbf{1}_Q$ denotes an indicator variable for the characteristic Q. A key feature of the model used is that h_0 has no analytical form, i.e., it is defined nonparametrically. Hence, it is determined only by the observed data.

Considering a set of data, one can learn from the model (Equation 7.8) via its corresponding likelihood functions. It is in these functions that the data censoring process comes into the analysis. A latent variable, δ_{ij}, is defined to be equal to one if the ijth duration time corresponds to a correct answer and zero otherwise. The likelihood function, L, is then defined by

$$
\begin{aligned}
L &= \prod_{\delta_{ij}=1} h(t_{ij}) \prod_{\delta_{ij}=0} S(t_{ij}) \\
&= \prod h(t_{ij})^{\delta_{ij}} S(t_{ij}).
\end{aligned}
\tag{7.10}
$$

Here, the parameters to be estimated are h_0, β and v. We fitted the model using the Bayesian framework [GCSR03], considering that h_0 is approximated by a B-spline and that the v_i have a Gamma distribution function. In particular, we used the package `survBayes` [HHM07] provided in `R` [Tea06], working with the standard prior specification considered therein.

7.5.3   Contrast of Map Type Effects

The idea of including frailties and other stratification variables in the model aims to differentiate effects attributable to the map type only without being perturbed by those factors. Working with the map type effect, we have considered keeping the geographic map as a reference and only including two dichotomous variables for the published and automatically-drawn map types. It is important to notice that these effects are invariant to any similar specification.

The map type effect is measured in terms of the ratio of hazard functions. For instance, the effect of using the published map versus the geographic map, given all the components, is given by

$$\frac{h(t \mid \text{pub}, v, ...)}{h(t \mid \text{geo}, v, ...)} = \exp(\beta_{\text{pub}}). \tag{7.11}$$

Similarly, the effect of using the automatically-drawn map versus the geographic map is given by

$$\frac{h(t \mid \text{auto}, v, ...)}{h(t \mid \text{geo}, v, ...)} = \exp(\beta_{\text{auto}}) \tag{7.12}$$

and the effect of using the automatically-drawn map versus the published map is given by

$$\frac{h(t \mid \text{auto}, v, ...)}{h(t \mid \text{pub}, v, ...)} = \exp(\beta_{\text{auto}} - \beta_{\text{pub}}). \tag{7.13}$$

Here, $h(t \mid \text{Q}, v, ...)$ defines the hazard function for a given map type Q.

Working in terms of Hypothesis 2, we can deduce that the automatically-drawn maps are better than the published maps if it is more probable that less time is required to find an optimal route using an automatically-drawn map than using a published map. In probabilistic terms, this hypothesis requires

$$\Pr(T \leq t \mid \text{auto}) \geq \Pr(T \leq t \mid \text{pub}), \tag{7.14}$$

which is equivalent to

$$\beta_{\text{pub}} \leq \beta_{\text{auto}}, \tag{7.15}$$

since $S_0(t)$ is bounded at $(0, 1)$ for all t.

To conclude, in a Bayesian framework, we can compute the probability of our hypothesis given the observed data. We can say that there is enough evidence in favour of our hypothesis if that probability is greater than or equal to 0.5. Recall that the parameters β_{pub} and β_{auto} are assumed random. Therefore, an equivalent rule to validate our hypothesis would be given by

$$\Pr(\beta_{\text{pub}} \leq \beta_{\text{auto}} \mid \text{data}) \geq 0.5. \tag{7.16}$$

Similarly, for Hypothesis 1, we can compare the automatically-drawn map against the baseline survival function, S_0, i.e.,

$$\Pr(\beta_{\text{auto}} \geq 0 \mid \text{data}) \geq 0.5, \tag{7.17}$$

which is equivalent to (7.16) when $\beta_{\text{pub}} = 0$.

7.5.4 Analysis of Route-Finding Task Data

Tables 7.3 and 7.4 give a summary of the empirical mean and standard deviation and quantiles for estimates of the β vector. Tables 7.5 and 7.6 show the same parameters but where the automatically-drawn maps are being compared with the published maps. Tables 7.7, 7.8 and 7.9 show simple statistics relating to the times taken by subjects to answer tasks for each map type, grouped by all answers, just correct answers and just incorrect answers respectively. Table 7.10 shows the contrast between the automatically-drawn and published map types. Figure 7.6 shows the baseline survival function, S_0.

Table 7.3: Empirical mean and standard deviation for each variable. β_{pub} and β_{auto} are relative to β_{geo} and the other five values are relative to β_{atlanta}.

β vector	Mean	Standard deviation	Naïve standard error	Time-series standard standard error
β_{pub}	-0.182131	0.09013	0.0005204	0.001279
β_{auto}	0.000967	0.08849	0.0005109	0.001285
$\beta_{\mathrm{bucharest}}$	-0.474466	0.11156	0.0006441	0.001819
β_{mexico}	-0.874868	0.11967	0.0006909	0.001942
$\beta_{\mathrm{stockholm}}$	-0.835034	0.12459	0.0007193	0.001896
β_{toronto}	-1.149131	0.13769	0.0007932	0.001975
$\beta_{\mathrm{washington}}$	-0.587355	0.11600	0.0006697	0.001772

The values in Table 7.10 can be used to assess Hypotheses 1 and 2. For Hypothesis 1,

$$\Pr(\beta_{\mathrm{auto}} \geq 0 \mid \mathrm{data}) \approx 0.5, \tag{7.18}$$

as the mean value when comparing the automatically drawn map with the baseline survival function is 0.000967. For Hypothesis 2,

$$\Pr(\beta_{\mathrm{pub}} \leq \beta_{\mathrm{auto}} \mid \mathrm{data}) \approx 0.9768. \tag{7.19}$$

This is illustrated in Figure 7.7 where the value of 0.9768 equates to the area to the right of the dotted line.

Therefore, we can be confident that Hypothesis 2 is valid, i.e. that the maps drawn using our automatic layout method were statistically significantly better for finding an optimal route than the published maps. With respect to Hypothesis 1, we can say that both the automatically-drawn maps and the geographic maps were statistically similar. These results are not surprising when considering the simple statistics such as the mean time to answer questions shown in Tables 7.7, 7.8 and 7.9—the mean times for the automatic and geographic maps are consistently similar and less than the mean times for the published map.

Table 7.4: Quantiles for each variable. β_{pub} and β_{auto} are relative to β_{geo} and the other five values are relative to β_{atlanta}.

β vector	2.5%	25%	50%	75%	97.5%
β_{pub}	-0.3578	-0.24270	-0.182584	-0.12221	-0.005765
β_{auto}	-0.1692	-0.05833	0.001393	0.05949	0.177048
$\beta_{\text{bucharest}}$	-0.6946	-0.54897	-0.475534	-0.39954	-0.253063
β_{mexico}	-1.1061	-0.95616	-0.876046	-0.79452	-0.640189
$\beta_{\text{stockholm}}$	-1.0820	-0.91960	-0.833905	-0.75063	-0.593195
β_{toronto}	-1.4190	-1.24033	-1.147951	-1.05543	-0.884289
$\beta_{\text{washington}}$	-0.8150	-0.66575	-0.586841	-0.50902	-0.357619

Table 7.5: Empirical mean and standard deviation for each variable. β_{pub} and β_{auto} are relative to β_{geo}.

β vector	Mean	Standard deviation	Naïve standard error	Time-series standard standard error
β_{pub}	0.8369	0.07562	0.0004366	0.0010655
β_{auto}	1.0049	0.08917	0.0005148	0.0013004

Table 7.6: Quantiles for each variable. β_{pub} and β_{auto} are relative to β_{geo}.

β vector	2.5%	25%	50%	75%	97.5%
β_{pub}	0.6992	0.7845	0.8331	0.8850	0.9943
β_{auto}	0.8443	0.9433	1.0014	1.0613	1.1937

Table 7.7: Statistics for all answers grouped by map type. All times in seconds.

Map type	Mean	σ	Min.	Max.	Range	Skew	Kurtosis	Standard error
auto	20.88	13.97	4.44	170.64	166.20	4.66	40.23	0.75
geo	21.12	11.94	6.59	86.96	80.37	2.08	6.26	0.64
pub	24.06	17.24	5.73	129.64	123.91	2.54	8.23	0.93

Table 7.8: Statistics for correct answers grouped by map type. All times in seconds.

Map type	Mean	σ	Min.	Max.	Range	Skew	Kurtosis	Standard error
auto	19.78	13.84	4.44	170.64	166.20	5.43	51.81	0.85
geo	19.74	10.52	6.59	67.62	61.03	1.84	4.16	0.71
pub	21.46	14.42	5.73	129.64	123.91	3.05	14.84	0.92
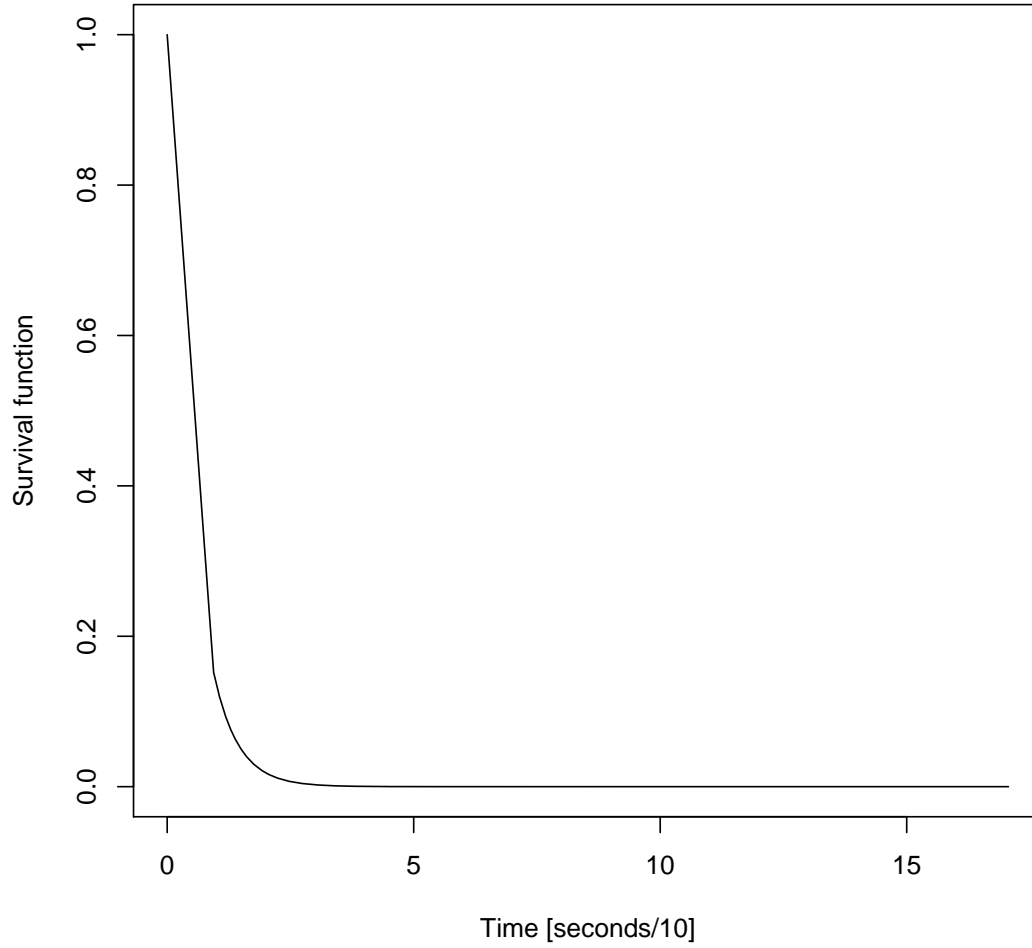
Figure 7.6: Baseline survival function, S_0.

Table 7.9: Statistics for incorrect answers grouped by map type. All times in seconds.

Map type	Mean	σ	Min.	Max.	Range	Skew	Kurtosis	Standard error
auto	24.55	13.85	8.82	101.92	93.1	2.62	10.72	1.56
geo	23.49	13.77	7.05	86.96	79.91	2.06	5.93	1.22
pub	30.46	21.52	10.37	103.14	92.77	1.76	2.52	2.16

Table 7.10: Contrast of map types: automatically-drawn versus published.

	Published	Automatic	Automatic versus Published	Contrast of Automatic versus Published
Min.	-0.5772	-0.362938	-0.1966	0.0000
1st Qu.	-0.2427	-0.058329	0.1226	1.0000
Median	-0.1826	0.001393	0.1836	1.0000
Mean	-0.1821	0.000967	0.1831	0.9768
3rd Qu.	-0.1222	0.059494	0.2450	1.0000
Max.	0.1501	0.378059	0.5176	1.0000
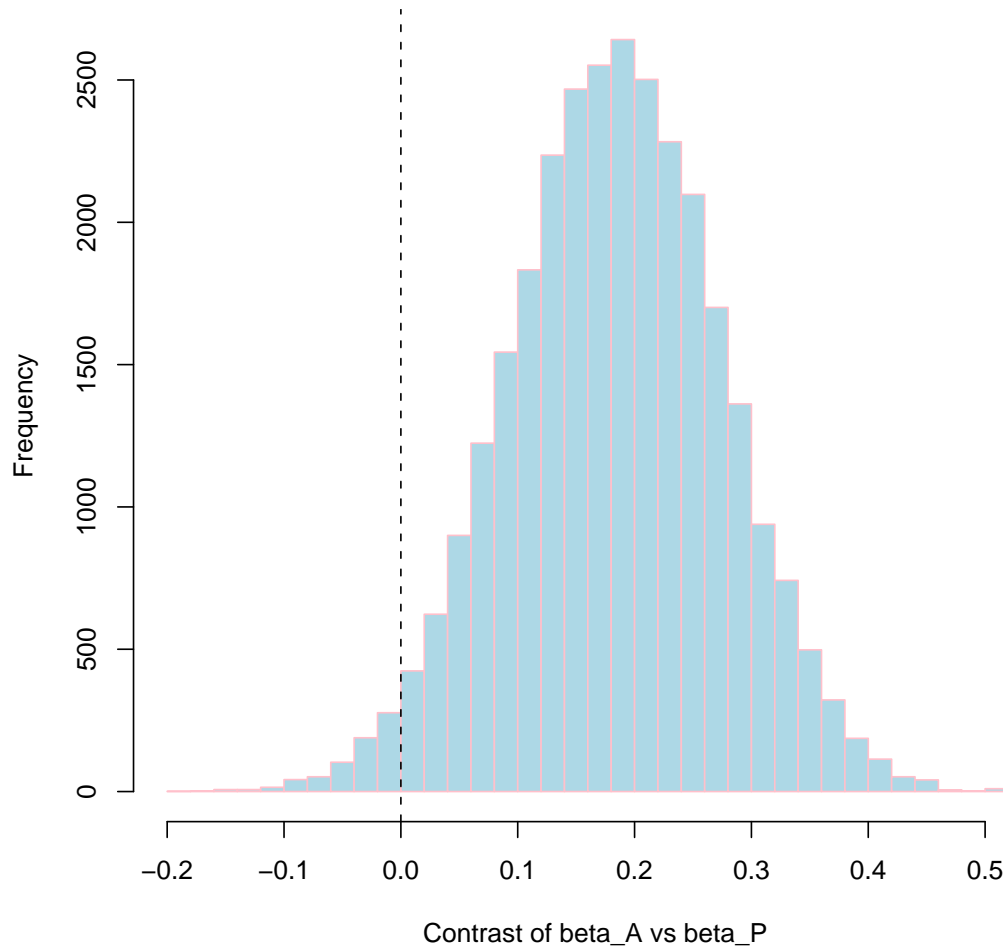
Figure 7.7: Contrast between the automatically drawn maps and published maps using the Bayesian framework.

7.5.5   Preference Tasks

We have analysed the preference data by considering the number and proportion of subjects that most preferred and least preferred each map. The numbers and proportions for the most preferable maps are shown in Table 7.11 and for the least preferable maps in Table 7.12.

Table 7.11: Number and proportion of subjects selecting a given map as their most preferable map.

Map	Number			Percent		
	Auto.-drawn	Geo.	Pub.	Auto.-drawn	Geo.	Pub.
Atlanta	21	10	12	48.84%	23.26%	27.91%
Bucharest	29	9	5	67.44%	20.93%	11.63%
Mexico	41	2	0	95.35%	4.65%	0.00%
Stockholm	5	2	36	11.63%	4.65%	83.72%
Toronto	24	5	14	55.81%	11.63%	32.56%
Washington	33	0	10	76.74%	0.00%	23.26%
ALL	153	28	77	59.30%	10.85%	29.84%

Table 7.12: Number and proportion of subjects selecting a given map as their least preferable map.

Map	Number			Percent		
	Auto.-drawn	Geo.	Pub.	Auto.-drawn	Geo.	Pub.
Atlanta	10	23	10	23.26%	53.49%	23.26%
Bucharest	8	12	23	18.60%	27.91%	53.49%
Mexico	1	5	37	2.33%	11.63%	86.05%
Stockholm	10	32	1	23.26%	74.42%	2.33%
Toronto	2	18	23	4.65%	41.86%	53.49%
Washington	1	40	2	2.33%	93.02%	4.65%
ALL	32	130	96	12.40%	50.39%	37.21%

The proportions of individuals in the survey that prefer the automatically-drawn maps (shown in Table 7.11) were generally higher than the proportions of individuals that preferred either of the other two maps. The only map where this was not the case was the Stockholm map. The results for Mexico City are particularly notable as none of the 43 individuals preferred the official published map. The maps which were least preferable (Table 7.12) also show a preference for the geographic maps to be the least preferable type, followed by the official published maps.

The proportions were analysed using a normal approximation to a binomial model. A result was considered statistically significant if the significance level was less than 0.05 (95%). Tables 7.13 and 7.14 show the p-values and 95% confidence intervals for the most preferable and least preferable maps respectively. Positive, statistically significant results (i.e. those which support the most or least preferable map) are highlighted on bold.

For five of the six maps, the automatically-drawn map was found most preferable over the other two types. Only the Stockholm map was different, with most people preferring the published map. This is possibly due to the highly symmetrical radial layout of the published map. Three maps were least preferable with the geographic map and three with the published map. When comparing each map type across the six city maps, the automatically-drawn maps were significantly more preferable than the other two types and the geographic maps were significantly less preferable than the other two types.

These observations support both Hypothesis 3 and Hypothesis 4 that the automatically-drawn maps are generally more preferable amongst the group of individuals surveyed than either the geographic maps or the official published maps.

Table 7.13: *p*-values and 95% confidence intervals for subjects selecting a given map as their most preferable map. Positive, statistically significant results with a significance level $p < 0.05$ are highlighted in bold.

| Map | *p*-value | | | 95% confidence interval | | |
	Auto.-drawn	Geo.	Pub.	Auto.-drawn	Geo.	Pub.
Atlanta	**0.0310**	0.1610	0.4504	$14.94 - 27.06$	$3.94 - 16.06)$	$5.94 - 18.06$
Bucharest	**< 0.0001**	0.0845	0.0025	$22.94 - 35.06$	$2.94 - 15.06)$	$-1.06 - 11.06$
Mexico	**< 0.0001**	0.0001	< 0.0001	$34.94 - 47.06$	$-4.06 - 8.06)$	$-6.06 - 6.06$
Stockholm	0.0025	0.0001	**< 0.0001**	$-1.06 - 11.06$	$-4.06 - 8.06)$	$29.94 - 42.06$
Toronto	**0.0018**	0.0018	0.9141	$17.94 - 30.06$	$-1.06 - 11.06)$	$7.94 - 20.06$
Washington	**< 0.0001**	< 0.0001	0.1610	$26.94 - 39.06$	$-6.06 - 6.06)$	$3.94 - 16.06$
ALL	**< 0.0001**	< 0.0001	0.2346	$138.16 - 167.84$	$13.16 - 42.84$	$62.16 - 91.84$

Table 7.14: p-values and 95% confidence intervals for subjects selecting a given map as their least preferable map. Positive, statistically significant results with a significance level $p < 0.05$ are highlighted in bold.

Map	p-value			95% confidence interval		
	Auto.-drawn	Geo.	Pub.	Auto.-drawn	Geo.	Pub.
Atlanta	0.1610	**0.0051**	0.1610	$3.94 - 16.06$	$16.94 - 29.06$	$3.94 - 16.06$
Bucharest	0.0405	0.4504	**0.0051**	$1.94 - 14.06$	$5.94 - 18.06$	$16.94 - 29.06$
Mexico	< 0.0001	0.0025	**< 0.0001**	$-5.06 - 7.06$	$-1.06 - 11.06$	$30.94 - 43.06$
Stockholm	0.1610	**< 0.0001**	< 0.0001	$3.94 - 16.06$	$25.94 - 38.06$	$-5.06 - 7.06$
Toronto	0.0001	0.2356	**0.0051**	$-4.06 - 8.06$	$11.94 - 24.06$	$16.94 - 29.06$
Washington	< 0.0001	**< 0.0001**	0.0001	$-5.06 - 7.06$	$33.94 - 46.06$	$-4.06 - 8.06$
ALL	< 0.0001	**< 0.0001**	0.1866	$17.16 - 46.84$	$115.16 - 144.84$	$81.16 - 110.84$

7.5.6 Questionnaire

The feedback from all subjects for the four questionnaire questions that they were asked about is shown in Appendix A.3.3. The questions in this questionnaire were mainly for our own curiosity—it was interesting to note some of the responses about the features of the metro maps layouts from people who were using the maps.

These are some of the note-worthy observations from the four questionnaire questions:

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

   - Seven people mentioned that they had seen metro maps that were not used in our experiment (mainly the London Underground map).
   - Four people said that they had seen maps that were used in the experiment.

2. Which features of the metro map layout did you find most helpful when completing the tests?

   - Many of the responses for the most helpful features noted the colour of lines as being most useful.
   - Other responses highlighted straight lines, good spacing between stations and the labels and the angle between lines.

3. Which features of the metro map layout did you find least helpful when completing the tests?

   - The least helpful features included the use of curved lines, squashed up text and lines and overlapping labels.
   - Other comments mentioned that some text was too small to read (although it wasn't actually necessary to read the text of the labels for our tests) and that yellow lines were difficult to see on a white background. These last two comments come about from the restrictions on the way that the test was conducted using a computer screen.

4. Did you find any of the questions ambiguous?

   - Some responses mentioned that there were several potential routes to choose from. While this was possibly the case, we tried to be careful in selecting questions and the set of possible answers based on what appeared to be the best route and trying to make the answers as unambiguous as possible.
   - Other responses highlighted the difficulty in reading some of the yellow parts of maps and others appeared to be unsure as to what constituted a change.

7.6 Conclusion

Our experiment set out to confirm four hypotheses related to comparing automatically-drawn, geographic and the official published maps of various real-world metro systems in terms of their use for finding optimal routes and the preferences of these maps.

We were able to confirm Hypothesis 2 that compared our automatically-drawn maps with published maps in finding the optimal route. Hypothesis 1 remained inconclusive, but performances using the automatically-drawn maps and the geographic maps were statistically similar. This is promising as it shows that maps produced using our method stand up well compared to the published maps.

The observations made of the preferences of the group of individuals surveyed support both Hypothesis 3 and Hypothesis 4 that the automatically drawn maps are more preferable than the geographic maps or the official published maps.

As with any empirical experiment of this nature, there are a number of limitations with our methodology. More subjects taking part in the experiment would have allowed us to be more confident of our results and would have ensured that our experiment was more representative. Nearly all of our subjects were drawn from the Computer Science Department at the University of Kent: this is unlikely to be an accurate sample of the population at large, although we believe it to be adequate for the experiment in question. We made a significant effort to ensure that each experiment session was conducted in as similar a way as possible to each other, but we cannot rule out any differences between sessions. Nor can we rule out subjects from earlier sessions talking to subjects due to take part in a later session (although they were discouraged from doing so). Other limitations include the way that the graphic designs of the published maps had to be normalised to make them visually comparable to the other map types and the way that geographic and published maps were subsequently labelled.

Chapter 8

Applications of the Metro Map Metaphor

8.1 Flexibility and Extensibility

One of the key advantages of multicriteria optimisation is that it is relatively easy to change the criteria that are used when laying out a map. This allows the flexibility to draw maps with different characteristics. These different characteristics allow us to consider using our method of drawing metro maps for different application areas beyond that of drawing public transport maps. Previous work on applying the metaphor to other application areas was described in Section 2.7.

This chapter describes how our metro map layout method was adapted to two different application areas: the visualisation of project plans (Section 8.3) and website navigation visualisation (Section 8.4). Both these application areas lend themselves particularly well to being modelled using the metro map metaphor. First, both applications are modelling interconnections between entities. For project plans, these interconnections are the dependencies between the various tasks; for website navigation, the interconnections are the links between different pages of a website. Second, there is a need to explore these interconnections and to determine "routes" through the visualisations. With project plans, the need is to be aware of the relative ordering of work tasks and the duration of work tasks; for website navigation, it is important to know how to find a path from one page from another.

8.2 Modifying the Metro Map Layout Method for Different Applications

As stated earlier, it is relatively easy to modify the set of criteria and the weightings for criteria to suit new application areas. Chapter 5 showed how the criteria and weightings

were adapted to fit the traditional metro map metaphor for its original purpose of visualising public transport networks. In a similar way, the modifications for each application area can be broken down to the selection of which criteria to use and which criteria weightings are most appropriate. Essentially, any changes to the criteria or weightings will be driven by any deviation from the original metro map metaphor definition described in Section 2.3. As well as the selection of different criteria and weightings, the initial embedding can also have an effect on the way that a particular application area modifies the metro map metaphor.

The selection of criteria is entirely dependent on the application area. Criteria can be both introduced and removed. An illustration of this is given by the project plan application (shown in the following section) where a new timescale criterion is introduced and the edge length criterion is removed. This is to suit a particular deviation from the metro map metaphor (notably, the introduction of a linear time axis). Also, the application areas presented here are completely abstract in nature and have no basis in the geographic setting of public transport networks. The introduced criteria are therefore likely to form some method of determining concrete positions for those abstract entities.

Weightings can also have an effect, but this is significantly less marked than the introduction or removal of a particular criterion. In the examples presented in this Chapter, the criteria weightings used were similar to those used for the other metro map examples shown in Chapter 6. The only modifications that were essential were to take account of new or removed criteria. In these cases, the weightings were determined in the same way as described in Section 5.3.

Particular modifications for the two application areas are discussed in more detail in the relevant Sections.

8.3 Visualisation of Project Plans

At present, Gantt charts and PERT (program evaluation and review technique) network charts are predominantly used for the mapping of projects in organisations [Gan73]. Gantt charts show each task in a project as a horizontal bar, aligned against a time axis. The time axis allows the exact temporal relationships between the tasks to be visualized. Each task bar can also be 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. A typical, simple Gantt chart is shown in Figure 8.1.

PERT network charts model the same project planning data but visualise it using a series of interconnecting timelines. As a network, each node represents a particular task and dependencies between tasks are shown as edges between the nodes representing

	Vorgangsna	Start	Finish	Duration	Pred	Resource I
1	Start 1	Wed 17/07/02	Fri 29/11/02	98 dys		1
2	Start 2	Fri 02/08/02	Tue 26/11/02	83 dys		2
3	Start 3	Fri 02/08/02	Fri 04/10/02	46 dys		3
4	Start 4	Fri 02/08/02	Fri 29/11/02	86 dys		4
5	Task 1	Mon 02/12/02	Fri 10/01/03	30 dys	1	1
6	Task 2	Wed 27/11/02	Tue 07/01/03	30 dys	2	2
7	Task 3	Mon 07/10/02	Fri 29/11/02	40 dys	3	3
8	Task 4	Tue 03/12/02	Fri 10/01/03	29 dys	4	4
9	Task 6	Mon 02/12/02	Fri 20/12/02	15 dys	7	3
10	Task 7	Mon 13/01/03	Fri 28/02/03	35 dys	5,8	1,2,3,4
11	Task 15	Mon 03/03/03	Fri 04/04/03	25 dys	10	1
12	Task 16	Mon 07/04/03	Fri 25/04/03	15 dys	11	2
13	Task 8	Mon 28/04/03	Tue 20/05/03	17 dys	12	4,3
14	Task 17	Wed 21/05/03	Fri 20/06/03	23 dys	13	1,2,3,4
15	Task 9	Mon 07/07/03	Wed 09/07/03	3 dys	14	1,2,3,4

Figure 8.1: A typical Gantt chart.
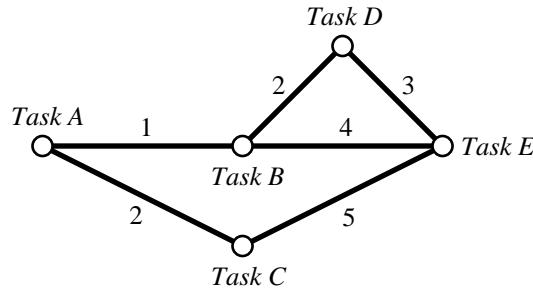
Figure 8.2: A simple PERT network chart. Durations (in days) are shown on the connecting edges. The critical path in this network is through Tasks *A*, *C* and *E* and takes 7 days.

those tasks. The duration of a task can be shown either by annotating the edge between that task and the next with a time value or adding the duration value as a label to the node for that task. An example of a simple PERT network chart is shown in Figure 8.2. While PERT networks are good at showing dependencies between tasks, they are less useful for showing the time taken by tasks as this information is provided in text form.

While Gantt charts and PERT network diagrams 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 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 [BM04] introduced the Tube Map Visualisation for projects in an organisation and evaluated its strengths and limitations in an comparative study [BMR+05]. Their findings are discussed in Section 2.7.

Modifications to our method for laying out metro maps are described in Section 8.3.1. In particular, we introduce a new criterion to take the time axis into account and to change the method for initial positioning of nodes. It is also worth pointing out that there is no geographical constraint as there would be with a normal map of a public

transport network. The concrete positions of nodes are determined based on their dependencies and times as modelled by the timescale criterion.

8.3.1   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 x_n and start time t_n the timescale criterion, c_{Nt} is found by calculating

$$c_{Nt} = (|t_n - x_n|)^2 .$$

(8.1)

This has the effect of severely penalising nodes that have strayed a long way from their start time while having little effect on nodes that are close to their start time. As with other criteria, the timescale criterion is sufficiently weighted so as to be effective when used in combination with the other criteria.

A side effect of the timescale criterion is that the edge length criterion is no longer required for this application, as these two criteria would conflict with each other.

8.3.2   Initial Embedding

When drawing metro maps, the starting position for the multicriteria optimisation 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 on the same y-coordinate. The initial x-coordinate of the node is determined by the start time of the task.

8.3.3   Implementation

Figure 8.3 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 8.1. Each line is drawn with a different colour 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, by studying where the lines from more than one member intersect, which tasks are important to more than one member of the project (such as *Task 7*). 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 octilinear.
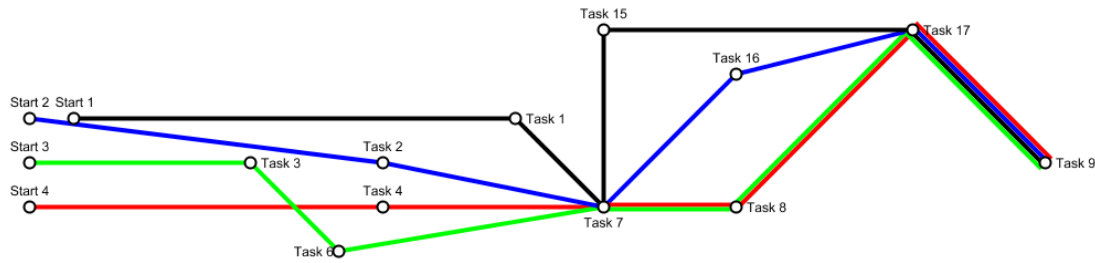
Figure 8.3: The resulting project map drawn using our automated metro map layout method. The chart is aligned from left to right along a time axis. Each line represents a group in the project; each station an individual or collective milestone in the project.

These problems could obviously be solved with manual editing of the project map, but this is not really desirable as it is meant to generate usable project maps automatically.

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 symbols representing the nodes. These problems are usually fixed in metro maps using stretched symbols (see Figure 2.12(a) and Section 2.2.5).

8.4 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 8.4, and site maps (which show all the pages on the website, normally grouped into categories or listed alphabetically), as shown in Figure 8.5. All of these methods are created using simple text—very few actually use a graphical visualisation 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 visualisation of the site. However, nearly all site maps 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
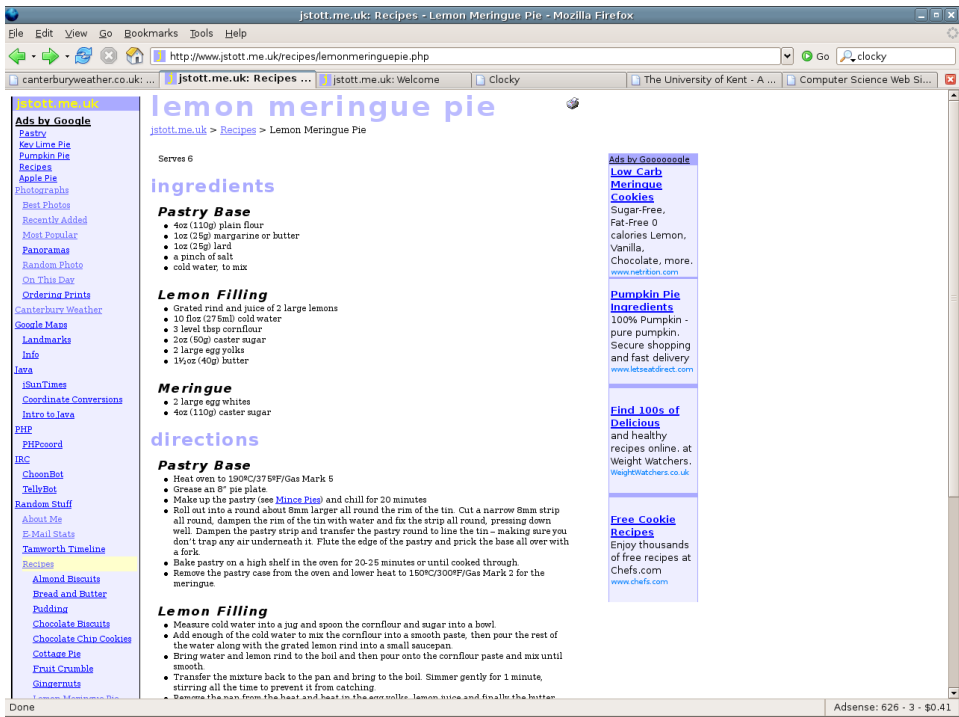
Figure 8.4: Example of a website that uses both breadcrumb trails (underneath the page heading) and a hierarchical list of contents (down the left-hand side). From jstott.me.uk.
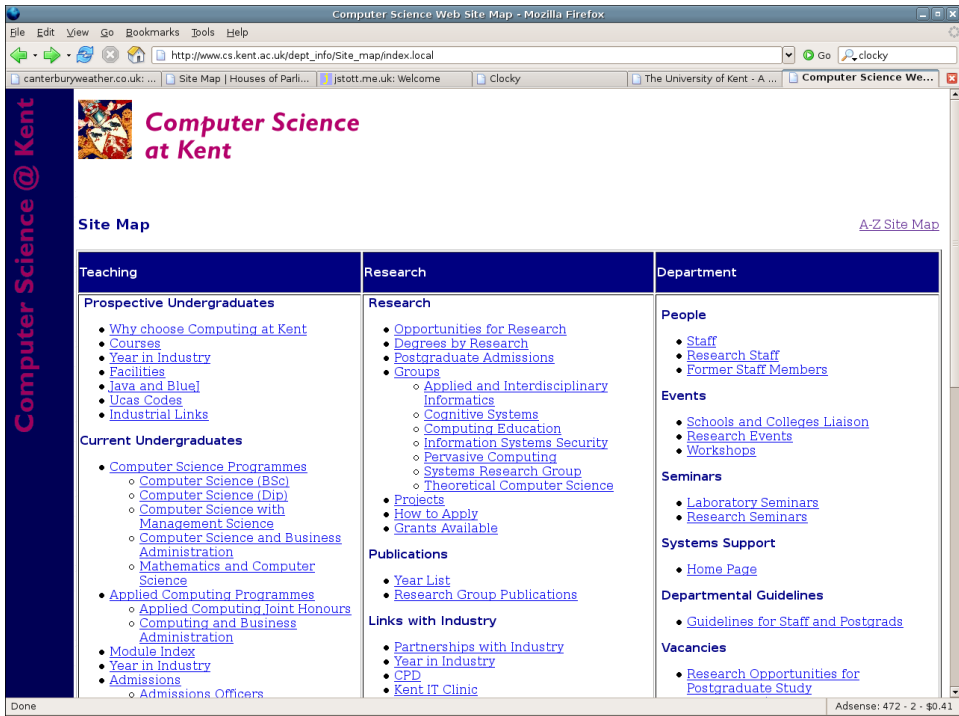
Figure 8.5: Example of a website site map. From the University of Kent Computer Science website.

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 visualisation 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 visualisation 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). The example navigation model used here is based on a first-year Computer Science course at the University of Kent for teaching Java and object-oriented programming. The topics were generated with reference to the material for this course.

One of the key differences in the approach required to draw such a metro map style diagram 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 optimisation.

Another difference is the way in which the user is likely to interact with the diagram. In this case, the user must be able to accept the visualisation and be able to use it effectively in order to justify our claim that use of 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).

In the following two sections, we present two different results using a different initial embedding for each one.

8.4.1 Initial Embedding with a Spring Embedder

The first of our website navigation maps uses an initial embedding created using a spring embedder (Section 2.4.4 and [Ead84]). The spring embedder operates on a randomly

Figure 8.6: A website navigation map using the metro map metaphor using a spring embedder to produce an initial embedding. The starting point for this map is the "Objects and Classes" node in the middle.

produced embedding with the intention of untangling the edges and separating the various edges. The repulsive force between unconnected nodes ensures that nodes that are not immediately adjacent are placed further apart allowing more space for labelling. Enough iterations of the spring embedder are executed so 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 8.6 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 visualisation 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 does not 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 does not 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. Again, these modifications are deviations from the original metro map metaphor but are necessary to support the new application area.

8.4.2 Initial Embedding with a Hierarchical Layout

Our second result uses a hierarchical layout algorithm [STT81] to find an initial embedding. A hierarchical layout algorithm makes sense in this case 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 required to maintain this conceptual ordering. Figure 8.7 shows the result of drawing a website navigation map using the metro map metaphor with a hierarchical layout algorithm producing the initial embedding. To provide an idea of how the visualisation might be incorporated into a website, a mock-up is available online at `http://www.jstott.me.uk/javatube/`.

As with the first result, using a spring embedder for the initial layout, this result clearly shows many of the characteristics of metro maps. One of the advantages of this result over the first is that the 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

Figure 8.7: A website navigation map using the metro map metaphor using a hierarchical layout algorithm to produce an initial embedding.

layout algorithm to enforce the conceptual ordering. A top-to-bottom navigation makes it more intuitive for users to find their way around the map. The direction of navigation could again be reinforced using directed edges as extra visual cues—this would be another deviation away from the metro map metaphor to support the application area.

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.

8.5 Conclusion

This chapter described two applications for the metro map metaphor beyond that of drawing public transport maps. The metaphor was applied to the visualisation of project plans and website navigation models. The required modifications to our method in order to cater for the different visualisations were also described. We hope that the use of automatic methods for laying out maps using the metro map metaphor will encourage its use for these and other application areas.

Chapter 9

Conclusion

This chapter provides a summary of the contributions that this research makes and also describes enhancements that would improve our method further given extra time.

9.1   Summary of Contributions

This thesis introduced a novel method for the layout of metro maps using a hill climbing multicriteria optimiser. This is the first time that a multicriteria optimisation method has been applied to the problem of producing metro maps. The method incorporates a number of node movement and label positioning criteria which, when combined, allow nodes and labels to be moved to more optimal locations in the search space (Chapter 3). Clusters of nodes were identified and moved similarly using a number of clustering and partitioning methods (Chapter 4). This method addresses our motivation for the automatic layout of metro maps and has been demonstrated with a number of real-world examples (Chapter 6). The effectiveness of this method was assessed in an empirical experiment (Chapter 7).

Using empirical experiments, we showed that the automatically-drawn metro maps can be more effective for route planning than the official published maps used, and equally effective as the geographic maps used. In addition, in many cases, the automatically-drawn metro map was preferred over both the geographic and published maps.

The use of an automatic layout method for drawing metro maps allows this metaphor to be used much more readily for the visualisation of concepts away from the traditional use for public transport networks. We demonstrated how our method can be modified to allow the production of maps of project plans and for the visualisation of website navigation hierarchies (Chapter 8).

9.2 Limitations of this Work

The main limitation of this work is the lack of rigour in the determination of what values to use for each criterion weighting. It is difficult to tell whether the results that we produced were optimal or whether they could have been improved by modification of the criteria weightings.

The empirical experiment described in Chapter 7 has scope for improvement. In particular, a larger number and a wider demographic range of subjects would have allowed us to be more confident of our results. The manner in which the design of the geographic maps and published maps had to be normalised may also have had an influence on the results and there may be better alternative ways of handling the problem of different design styles between maps.

9.3 Further Work

This section discusses some potential ideas that could be applied to our method in the future.

9.3.1 Other Metro Map Characteristics

A number of other characteristics of metro maps can be incorporated into our method. These include drawing lines using several segments (polylines), using rounded corners rather than angular corners for lines, and better handling of multiple parallel lines.

In our method we consider each edge between two nodes as a single line. Allowing edges between nodes to be formed of several segments will allow a much greater freedom for the positioning of nodes. This is likely to be of great help in situations where lines cannot quite be drawn perfectly octilinearly, as shown in Figure 9.1. Many real world maps also use rounded corners at the point where lines change direction—these corners have a finite size that would certainly need to be taken into consideration. It is worth noting that some maps—notably that of the London Underground—have barely any situations where a line changes direction under a node. If a line does need to change direction it is usually just before or just after a node, thereby improving the appearance of the continuity of a metro line through the node. Examples of using rounded corners are shown in Figure 9.2.

We did not explicitly handle multiple parallel lines in our method and we also modelled stations as point features. Many maps use finite-sized devices to represent interchange stations between several lines, as shown in Figure 2.12 on page 21. When considering drawing parallel lines like this, the order that these lines pass through stations becomes important so that the continuity of the metro line is maintained, as
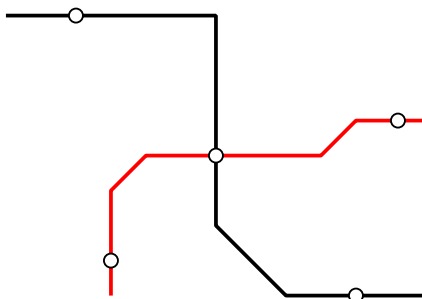
Figure 9.1: Drawing lines with multiple line segments.
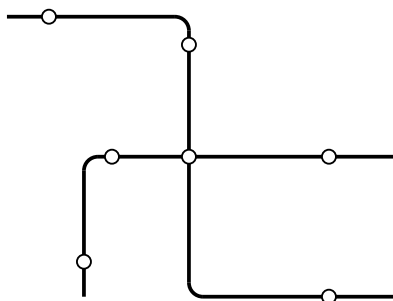
Figure 9.2: Using rounded corners on lines. Note that the positioning of nodes for stations is affected by the rounded corner.

shown in Figure 9.3. Benkert *et al.* discuss how the ordering of parallel edges can be optimised [BNUW07].

Finally, we did not consider incorporating topographic metadata in our method, yet many real-world maps feature some form of topographic metadata. This is a particularly challenging enhancement when one considers the variety of different types of metadata that could be included on a map. Metadata might be in the form of boundary line (such as a coastline), a line feature (such as a road or river) or even an area feature (such as parkland).

9.3.2   Node Movement Criteria and Weightings

Several areas of future work involve the node movement criteria and their weightings. There is the possibility of introducing further criteria to consider aesthetic features other than nodes or edges. For example, it may be desirable to maximise the area of faces thereby increasing the space available for labels. Also, a more rigorous and

Figure 9.3: Ordering of multiple metro lines passing through a node. In the left-hand example, the ordering of the three metro lines is not preserved as they pass through the node, whereas in the right-hand example, the ordering is preserved.

methodical approach to determining the weightings to use for each of the criteria is required. There is scope for this to be done empirically, for example, by producing maps with differing criteria weightings and conducting studies in a similar manner to those that we conducted in Chapter 7. The challenge in this case is that confounding factors may make it difficult to determine the effect of each individual criterion.

The results presented in Chapter 6 and the discussion in Section 5.4 highlighted a number of examples of local minima in the search space. It would be useful to characterise these local minima in order to investigate how the method could be adapted or extended in order to reduce the possibility of the minima occurring.

9.3.3   Labelling

A number of features of labelling on existing metro maps have been omitted from our handling of labelling. The most prominent of these is to include diagonal labels, and extra criteria could also be added to enhance the labelling results.

Diagonal labels are a fairly common feature of metro maps. While our results using only horizontal labels are generally good, certain circumstances can produce occasions where labelling with diagonal labels would be more aesthetically pleasing. Particular cases include long, horizontal lines where there is very little space for length labels above or below the line, or diagonal lines where it is desirable to keep the labels perpendicular to the line. Diagonal labels impose an extra level of complexity when testing for occlusions or proximity as the rectangular bounding box of the label can no longer be used for these purposes.

Further criteria can be added: the proximity of labels to other labels and edges would be useful in some circumstances (the proximity to nodes was necessary to avoid ambiguities and was discussed in Section 3.9.5), and the cases where horizontal labels are allowed to alternate either side of a horizontal line could be explicitly handled with a new criterion.

9.3.4   Computational Efficiency

As described in Section 5.1, we were not concentrating on optimising the computational efficiency of our software. Several enhancements could be made to our method to improve its computational efficiency, for example, only calculating local criteria or making judgements about whether it is worth considering a node or label for movement. In the implementation of our method we incorporated a number of efficiency improvements, but there is room for still more.

Frequently, nodes are positioned once in the first iteration of our method and they never move again. It should be possible to identify at least some nodes that have already found an optimum position and exclude those from the set of nodes to process for each iteration. This could be done by assessing whether anything in the vicinity of a node

has changed such that an improved position for that node may have become available. Clearly, if the node was not moved in previous iterations and nothing has changed in the locality of that node, it is unlikely that any change will be an improvement.

Local criteria can be used to significantly speed up our method. For example, it would be pointless to test whether a label on one side of the map occluded labels on the other side of the map. Equally, when moving a node, the only edges that need be considered when calculating the edge length criterion are those that are adjacent to that node.

9.3.5   Other Enhancements

One enhancement that could make a significant effect of the quality of our maps is to apply some pre-processing steps on the initial layout. For example, the centrality-based scaling methods of Merrick and Gudmundsson [MG06] could help make space for labels in dense or complex maps. Centrality-based scaling allows a graph to be scaled based on some measure of the density or centrality of areas of the graph (see Section 2.8.4). The measures of centrality could be extended to incorporate not only the density of nodes but also the density of labels and lines (or edges). For metro maps, this would allow a greater scale not just for areas where there are lots of stations but also for areas where many lines converge or where there are a large number of labels.

9.4   In Conclusion

With this work, we have been able to demonstrate empirically that our hill-climbing multicriteria optimisation method is capable of producing maps that conform to the metro map metaphor. We have shown how the optimisation method can be built from simple building blocks of individual criteria and combined using weightings to be able to produce quality maps. We have shown how the metro map metaphor can be applied to other application areas and how the availability of a method for automatically producing these maps to a good quality means that they stand a greater chance of being adopted for wider use.

Bibliography

[AF84]   John Ahn and Herbert Freeman. A program for automatic name place-
             ment. *Cartographica: the International Journal for Geographic Informa-
             tion and Geovisualization*, 21(2–3):101–109, 1984.

[AH01]   Silvania Avelar and R. Huber. Modeling a public transport network for
             generation of schematic maps and location queries. In *Proceedings of the
             20th International Cartographic Conference*, pages 1472–1480, 2001.

[AH06]   Silvania Avelar and Lorenz Hurni. On the design of schematic transport
             maps. *Cartographica: the International Journal for Geographic Infor-
             mation and Geovisualization*, 41(3):217–228, 2006.

[AM00]   Silvania Avelar and Matthias Müller. Generating topologically correct
             schematic maps. In *Proceedings of the 9th International Symposium on
             Spatial Data Handling*, pages 4a.28–4a.35, Beijing, China, 2000.

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

[Auc07]   Veolia Transport Auckland website. `http://www.connexauckland.
             com/`, 2 September 2007.

[AV00]   Pankaj K. Agarwal and Kasturi R. Varadarajan. Efficient algorithms for
             approximating polygonal chains. *Discrete and Computational Geometry*,
             23(2):273–291, 2000.

[Ave02]   Silvania Avelar. *Schematic maps on demand: design, modeling and vi-
             sualization*. PhD thesis, Swiss Federal Institute of Technology, Zurich,
             2002.

[BAR07]   San Francisco Bay Area Rapid Transit (BART) website. `http://www.
             bart.gov/`, 2 September 2007.

[BBS97]   Jürgen Branke, Frank Bucher, and Hartmut Schmeck. Using genetic algorithms for drawing undirected graphs. In *Third Nordic Workshop on Genetic Algorithms and their Applications*, pages 193–206, Helsinki, Finland, 1997. Finnish Artificial Intelligence Society (Hrsg.).

[BCC+06]   Prosenjit Bose, Otfried Cheong, Sergio Cabello, Joachim Gudmundsson, Marc J. van Kreveld, and Bettina Speckmann. Area-preserving approximations of polygonal paths. *Journal of Discrete Algorithms*, 4(4):554–566, 2006.

[BCD+02]   Gill Barequet, Danny Z. Chen, Ovidiu Daescu, Michael T. Goodrich, and Jack Snoeyink. Efficiently approximating polygonal paths in three and higher dimensions. *Algorithmica*, 33(2):150–167, 2002.

[BEKW02]   Ulrik Brandes, Markus Eiglsperger, Michael Kaufmann, and Dorothea Wagner. Sketch-driven orthogonal graph drawing. In *Proceedings of the 10th International Symposium on Graph Drawing (GD2002)*, volume 2528 of *Lecture Notes in Computer Science (LNCS)*, pages 1–11. Springer-Verlag, 2002.

[Ber99]   François Bertault. A force-directed algorithm that preserves edge crossing properties. In *Proceedings of Graph Drawing 1999*, volume 1731 of *Lecture Notes in Computer Science (LNCS)*, pages 351–358. Springer-Verlag, 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 *Lecture Notes in Computer Science (LNCS)*, pages 41–53. Springer, 2000.

[BM91]   Barbara P. Buttenfield and Robert B. McMaster, editors. *Map generalization: making rules for knowledge representation*. Longman, London, 1991.

[BM04]   Remo Aslak Burkhard and Michael Meier. Tube map: evaluation of a visual metaphor for interfunctional communication of complex projects. In *Proceedings of the 4th International Conference on Knowledge Management (I-KNOW 2004)*, pages 449–456, Graz, Austria, 2004. Know-Centre Graz.

[BMR$^+$05] Remo Aslak Burkhard, Michael Meier, Peter Rodgers, Matthias Thomas Jelle Smis, and Jonathan Mark Stott. Knowledge visualization: a comparative study between project tube maps and Gantt charts. In Klaus Tochterman and Hermann Maurer, editors, *Proceedings of the 5th International Conference on Knowledge Management (I-KNOW 2005)*, Journal of Universal Computer Science, pages 388–395, Graz, Austria, 2005. Know-Center Graz.

[BNUW07]   Marc Benkert, Martin Nöllenburg, Takeaki Uno, and Alexander Wolff. Minimizing intra-edge crossings in wiring diagrams and public transportation maps. In Michael Kaufmann and Dorothea 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.

[BW88]   Kurt E. Brassel and Robert Weibel. A review and conceptual framework of automated map generalization. *International Journal of Geographical Information Systems*, 2:229–244, 1988.

[Cab04]   Sergio Cabello. *Geometric Problems in Cartographic Networks*. PhD thesis, Utrecht University, 2004.

[Cam01]   John Campbell. *Map Use and Analysis*. McGraw-Hill, Boston, 4th edition, 2001.

[CBKF01]   Hernan Casakin, Thomas Barkowsky, Alexander Klippel, and Christian Freska. Schematic maps as wayfinding aids. In *Spatial Cognition II*, number 1849 in Lecture Notes in Computer Science (LNCS), pages 54–71. Springer-Verlag, 2001.

[CC85]   David G. Clayton and Jack Cuzick. Multivariate generalization of the proportional hazards model (with discussion). *Journal of the Royal Statistical Society, Series A*, 148:82–117, 1985.

[CC96]   Wing Shiu Chan and Francis Y. L. 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]   Danny Z. Chen and Ovidiu Daescu. Space-efficient algorithms for approximating polygonal curves in two-dimensional space. *International Journal of Computational Geometry and Applications*, 13:95–111, 2003.

[CdBvDvK01] Sergio Cabello, Mark de Berg, Steven van Dijk, and Marc J. van Kreveld. Schematization of road networks. In *Proceedings of the 17th Symposium on Computational Geometry*, pages 33–39, Medford, USA, 2001. ACM.

[CdBvK02]   Sergio Cabello, Mark de Berg, and Marc J. van Kreveld. Schematization of networks. Technical Report UU-CS-2002-040, Institute of Information and Computing Sciences, Utrecht University, Utrecht, 2002.

[CDH$^+$05] Danny Z. Chen, Ovidiu Daescu, John Hershberger, Peter M. Kogge, Ningfang Mi, and Jack Snoeyink. Polygonal path simplification with angle constraints. *Computational Geometry: Theory and Applications*, 32(3):173–187, 2005.

[Čer85]   V. Černý. A thermodynamical approach to the travelling salesman problem: an efficient simulation algorithm. *Jounal of Optimization Theory and Applications*, 45:41–51, 1985.

[CJ90]   Anthony C. Cook and Christopher B. Jones. A Prolog rule-based system for cartographic name placement. *Computer Graphics Forum*, 9:109–126, 1990.

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

[CO84]   David R. Cox and David Oakes. *Analysis of Survival Data*. Chapman and Hall, 1984.

[Cox72]   David R. Cox. Regression models and life-tables (with discussion). *Journal of the Royal Statistical Society*, 34:187–220, 1972.

[CR99a]   Vijay Chandru and M. R. Rao. *Algorithms and Theory of Computation Handbook*, chapter 32, pages 32/1–32/45. CRC Press, 1999.

[CR99b]   Vijay Chandru and M. R. Rao. *Algorithms and Theory of Computation Handbook*, chapter 31, pages 31/1–31/37. CRC Press, 1999.

[CSP96]   Michael K. Coleman and Douglas Stott Parker. Aesthetics-based graph layout for human consumption. *Software—Practice and Experience*, 26(12):1415–1438, 1996.

[CvK02]   Sergio Cabello and Marc J. van Kreveld. Schematic networks: an algorithm and its implementation. In Dianne E. Richardson and Peter van Oosterom, editors, *Advances in Spatial Data Handling*, pages 475–486, Ottawa, Canada, 2002. Springer.

[dBETT94a] Giuseppe di Battista, Peter Eades, Roberto Tamassia, and Ioannis G. Tollis. Algorithms for drawing graphs: an annotated bibliography. Available from `ftp://wilma.cs.brown.edu/pub/papers/compgeo/gdbiblio.ps.Z`, 1994.

[dBETT94b] Giuseppe di Battista, Peter Eades, Roberto Tamassia, and Ioannis G. Tollis. Annotated bibliography on graph drawing algorithms. *Computational Geometry: Theory and Applications*, 4:235–282, 1994.

[dBETT99]   Giuseppe di Battista, Peter Eades, Roberto Tamassia, and Ioannis G. Tollis. *Graph Drawing: Algorithms for the Visualisation of Graphs*. Prentice Hall, 1999.

[Den99]   Borden D. Dent. *Cartography: Thematic Map Design*. McGraw-Hill, 5th edition, 1999.

[DF92]   Jeffrey Scott Doerschler and Herbert Freeman. A rule-based system for dense-map name placement. *Communications of the Association of Computing Machinery*, 35(1):68–79, January 1992.

[dFPP90]   Hubert de Fraysseix, János Pach, and Richard Pollack. How to draw a planar graph on a grid. *Combinatorica*, 10:41–51, 1990.

[DH96]   Ron Davidson and David Harel. Drawing graphs nicely using simulated annealing. *ACM Transactions on Graphics*, 15(4):301–331, 1996.

[DM90]   Chen Ding and Prabhaker Mateti. A framework for the automated drawing of data structure diagrams. *ACM Transactions on Software Engineering*, 16(5):543–557, May 1990.

[Dow05]   Andrew Dow. *Telling The Passenger Where To Get Off*. Capital Transport Publishing, 2005.

[DP73]   David Douglas and Thomas 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.

[ECMS86]   Shawn Edmondson, Jon Christensen, Joe Marks, and Stuart Shieber. A general cartographic labeling algorithm. *Cartographica*, 33(4):13–23, 1986.

[EFK00]   Markus Eiglsperger, Ulrich Fößmeier, and Michael Kaufmann. Orthogonal graph drawing with constraints. In *Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2000)*, pages 3–11, San Francisco, California, 2000. ACM/SIAM.

[EFK01]   Markus Eiglsperger, Sándor P. Fekete, and Gunnar Klau. *Orthogonal Graph Drawing*, volume 2025 of *Lecture Notes in Computer Science (LNCS)*, chapter 6, pages 121–171. Springer-Verlag, 2001.

[EJJ80]   Regina C. Elandt-Johnson and Norman L. Johnson. *Survival Models and Data Analysis*. Wiley, 1980.

[Elr88a]   Daniel Elroi. Designing a network line-map schematization software enhancement package. In *Proceedings of the Eighth Annual ESRI User Conference*, 1988.

[Elr88b]   Daniel Elroi. GIS and schematic maps: a new symbiotic relationship. In *Proceedings of GIS/LIS 1988*. American Congress on Surveying and Mapping, 1988.

[ET94]   David Eu and Godfried T. Toussaint. On approximating polygonal curves in two and three dimensions. *CVGIP: Graphical Model and Image Processing*, 56(3):231–246, 1994.

[FA87]   Herbert Freeman and John Ahn. On the problem of placing names in a geographic map. *International Journal of Pattern Recognition and Artificial Intelligence*, 1(1):121–140, 1987.

[FK95]   Ulrich Fößmeier and Michael Kaufmann. Drawing high degree graphs with low bend numbers. In *Proceedings of the Symposium on Graph Drawing*, pages 254–266, 1995.

[FLM95]   Arne Frick, Andreas Ludwig, and Heiko Mehldau. A fast adaptive layout algorithm for undirected graphs. In *Proceedings of Graph Drawing 1994*, volume 894 of *Lecture Notes in Computer Science (LNCS)*, pages 388–403. Springer-Verlag, 1995.

[Flo62]   Robert W. Floyd. Algorithm 97: shortest path. *Communications of the ACM*, 5:354, 1962.

[FR91]   Thomas M. J. Fruchterman and Edward M. Reingold. Graph drawing by force-directed placement. *Software - Practice and Experience*, 21(11):1129–1164, 1991.

[Fre77]   Linton C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40:35–41, 1977.

[FRM03]   Jean Flower, Peter J. Rodgers, and Paul Mutton. Layout metrics for Euler diagrams. In *Seventh International Conference on Information Visualization (IV03)*, pages 273–280. IEEE, 2003.

[FW91]   Michael Formann and Frank Wagner. A packing problem with applications to lettering of maps. In *Proceedings of the Seventh Annual Symposium on Computational Geometry*, pages 281–288, North Conway, New Hampshire, July 1991. ACM.

[Gan73]   Henry Laurence Gantt. *Work, Wages and Profits*. Hive, Easton, Maryland, 2nd edition, 1973.

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

[GCSR03]   Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin. *Bayesian Data Analysis*. Chapman and Hall/CRC, 2nd edition, 2003.

[Ger99]   Sabih H. Gerez. *Algorithms for VLSI Design Automation*. John Wiley and Sons, Chichester, 1999.

[GJ83]   Michael R. Garey and David S. Johnson. Crossing number is NP-complete. *SIAM Journal of Algebraic and Discrete Methods*, 4(3):312–316, 1983.

[GM98]   Carsten Gutwenger and Petra Mutzel. Planar polyline drawings with good angular resolution. In S. H. Whitesides, editor, *Proceedings of Graph Drawing 1998 (GD1998)*, volume 1547 of *Lecture Notes in Computer Science (LNCS)*, pages 167–182. Springer-Verlag, 1998.

[GN98]   E. R. Gansner and S. C. North. Improved force-directed layouts. In S. H. Whitesides, editor, *Proceedings of Graph Drawing 1998 (GD1998)*, volume 1547 of *Lecture Notes in Computer Science (LNCS)*, pages 167–182. Springer-Verlag, 1998.

[GNS07]   Joachim Gudmundsson, Giri Narasimhan, and Michael H. M. Smid. Distance-preserving approximations of polygonal paths. *Computational Geometry Theory and Applications*, 36(3):183–196, April 2007.

[Goo95]   Michael T. Goodrich. Efficient piecewise-linear function approximation using the uniform metric. *Discrete and Computational Geometry*, 14:445–462, 1995.

[Goo07]   Google Maps. `http://www.google.com/maps`, 5 August 2007.

[HCC98]   Ian Heywood, Sarah Cornelius, and Steve Carver. *An Introduction to Geographical Information Systems*. Addison-Wesley, New York, 1998.

[Hen29]   G. Hennet. A map of the county palatine of Lancaster. Available from `http://www.lancashire.gov.uk/environment/oldmap/hennet/index.htm`, 13 February 2008, 1829.

[HHM07]   Volkmar Henschel, Christiane Heiss, and Ulrich Mansmann. *survBayes: Fits a proportional hazards model to time to even data by a Bayesian approach*, 2007. R package version 0.2.1.

[Hir82]   Stephen A. Hirsch. An algorithm for automatic name placement around point data. *The American Cartographer*, 9(1):5–17, 1982.

[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.

[HNR68]   P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

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

[Hou00]   Philip Hougaard. *Analysis of Multivariate Survival Data*. Springer-Verlag, New York, 2000.

[HS98]   John Hershberger and Jack 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.

[HT74]   John Hopcroft and Robert Tarjan. Efficient planarity testing. *Journal of the Association of Computing Machinery*, 21(4):549–568, 1974.

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

[II86a]   Hiroshi Imai and Masao Iri. Computational-geometric methods for polygonal approximations of a curve. *Computational Vision, Graphic and Image Processing*, 36(1):31–41, 1986.

[II86b]   Hiroshi Imai and Masao Iri. An optimal algorithm for approximating a piecewise linear function. *Journal of Information Processing*, 9(3):159–162, 1986.

[II88]   Hiroshi Imai and Masao Iri. Polygonal approximations of a curve-formulation and algorithms. In Godfried T. Toussaint, editor, *Computational Morphology*, Amsterdam, 1988.

[Imh75]   Eduard Imhof. Positioning names on maps. *The American Cartographer*, 2(2):128–144, 1975.

[Jav08]   Java, Sun Microsystems. `http://java.sun.com/`, 23 February 2008.

[Joh82]   David Johnson. The NP-completeness column: an ongoing guide. *Journal of Algorithms*, 3:89–99, 1982.

[Jon89]   C. Jones. Cartographic name placement with Prolog. *IEEE Computer Graphics and Applications*, 9(5):36–47, September 1989.

[JS008]   Java Swing, Sun Microsystems. `http://java.sun.com/javase/6/docs/technotes/guides/swing/index.html`, 23 February 2008.

[KGJV83]   Scott Kirkpatrick, C. D. Gelatt Jr., and Mario P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[KI88]   T. Kato and Hiroshi Imai. The NP-completeness of the character placement problem of 2 or 3 degrees of freedom. In *Record of the Joint Conference of Electrical and Electronic Engineers in Kyushu*, page 1138. In Japanese, 1988.

[KK89]   Tomihisa Kamada and Satoru Kawai. An algorithm for drawing general undirected graphs. *Information Processing Letters*, 31:7–15, 1989.

[Kle99]   Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46:604–632, 1999.

[KM98]   Gunnar Klau and Petra Mutzel. Quasi-orthogonal drawing of planar graphs. Technical Report MPI-I-98-1-013, Max-Planck-Institut für Informatik, Saarbrücken, Germany, May 1998.

[KMS91]   Corey Kosak, Joe Marks, and Stuart Shieber. A parallel genetic algorithm for network-diagram layout. In *Proceedings of the 4th International Conference on Genetic Algorithms*, pages 458–465, 1991.

[Kor88]   Richard E. Korf. *Exploring Artificial Intelligence: Survey Talks from the National Conferences on Artificial Intelligence*, pages 197–237. Morgan Kaufmann, San Mateo, California, 1988.

[KT98]   Konstantinos G. Kakoulis and Ioannis G. Tollis. A unified approach to labelling graphical features. In *14th Annual ACM Symposium in Computational Geometry*. ACM, June 1998.

[KW01]   Michael Kaufmann and Dorothea Wagner. *Drawing Graphs: Methods and Models*, volume 2025 of *Lecture Notes in Computer Science (LNCS)*. Springer-Verlag, 2001.

[LP86]   Thomas K. Langran and Gail E. Poiker. Integration of name selection and name placement. In *Proceedings of the Second International Symposium on Spatial Data Handling*, pages 50–64, Seattle, Washington, 1986. International Geographic Union and International Cartographic Association.

[LS01]   Ulrich Lauther and Andreas Stübinger. Generating schematic cable plans using spring embedder methods. In *Proceedings of Graph Drawing 2001*, volume 2265 of *Lecture Notes in Computer Science (LNCS)*, pages 465–466. Springer-Verlag, 2001.

[MAR08]   Atlanta Metropolitan Area Rapid Transit Authority (MARTA) website. `http://www.itsmarta.com/`, 13 February 2008.

[MB93]   William A. Mackaness and Kate Beard. Use of graph theory to support map generalization. *Cartography and Geographic Information Systems*, 20(4):210–221, 1993.

[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.

[Met08]   Metrorex (Bucharest Metro) website. `http://www.metrorex.ro/`, 13 February 2008.

[Mex08]   Mexico City 'Sistema de Transporte Colectivo' website. `http://www.metro.df.gob.mx/`, 13 February 2008.

[MG06]   Damian Merrick and Joachim Gudmundsson. Increasing the readability of graph drawings with centrality-based scaling. In Kazuo Misue, Kozo 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 Michael Kaufmann and Dorothea Wagner, editors, *Proceedings of the 14th International Symposium on Graph Drawing (GD2006)*, volume 4372 of *Lecture Notes in Computer Science (LNCS)*, pages 258–269. Springer-Verlag, 2007.

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

[MLW95]   J. C. Müller, J. P. Lagrange, and R. Weibel, editors. *GIS and generalization - methodology and practice*, volume 1 of *GISDATA*. Taylor and Francis, London, 1995.

[MO85]   Z. Miller and James B. Orlin. NP-completeness for minimizing maximum edge length in grid embeddings. *Journal of Algorithms*, 6:10–16, 1985.

[MO88]   Avraham Melkman and Joseph O'Rourke. On polygonal chain approximation. In Godfried T. Toussaint, editor, *Computational Morphology*, Amsterdam, 1988.

[Mor96a]   Alastair Morrison. Designing 'octopus' bus maps for districts within a city with the aid of scale distortion. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 33(4):37–60, 1996.

[Mor96b]   Alastair Morrison. Public transport maps in Western European cities. *The Cartographic Journal*, 32(2):93–110, 1996.

[MRR+53]   Nicholas Constantine Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6):1087–1091, 1953.

[MS91]   Joe Marks and Stuart Shieber. The computational complexity of carto-
           graphic label placement. Technical Report TR-05-91, Harvard Univer-
           sity, 1991.

[MS92]   Robert B. McMaster and K. Stuart Shea. *Generalization in Digital Car-
           tography*. Association of American Geographers, Washington D.C., 1992.

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

[Mut08]   Paul Mutton. EPS Graphics2D. `http://www.jibble.org/`
           `epsgraphics/`, 15 June 2008.

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

[Nes04]   Keith V. Nesbitt. Getting to more abstract places with the metro map
           metaphor. In Ebad Banissia, Katy Brner, Chaomei Chen, Muham-
           mad 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.

[Nöl05]   Martin Nöllenburg. Automated drawing of metro maps. Master's thesis,
           Institut für Theoretische Informatik, Universität Karlsruhe, 20 August
           2005.

[Nom87]   Elliot Noma. Heuristic method for label placement in scatter plots.
           Psychometrika, 52(3):463–468, 1987.

[ORe]   O'Reilly 2003 open source route map. `http://www.oreilly.de/`
           `artikel/routemap.html`.

[OSM07]   OpenStreetMap. `http://wiki.openstreetmap.org/`, 5 August 2007.

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

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

[PAC01]   Helen C. Purchase, Jo-Anne Allder, and David A. Carrington. User
           preference of graph layout aesthetics: a uml study. In Joe Marks, edi-
           tor, *Proceedings of the 8th International Symposium on Graph Drawing
           (GD2000)*, volume 1984 of *Lecture Notes in Computer Science (LNCS)*,
           pages 127–138. Springer-Verlag, 2001.

[PCA02]   Helen C. Purchase, David A. Carrington, and Jo-Anne Allder. Empirical evaluation of aesthetics-based graph layout. *Empirical Software Engineering*, 7(3):233–255, 2002.

[PCJ95]   Helen C. Purchase, Robert F. Cohen, and M. Jones. Validating graph drawing aesthetics. In Franz J. Brandenburg, editor, *Symposium on Graph Drawing*, volume 1027 of *Lecture Notes in Computer Science (LNCS)*, pages 435–446, Passau, Germany, 1995. Springer-Verlag.

[PT97]   Achilleas Papakostas and Ioannis G. Tollis. Orthogonal drawing of high degree graphs with small area and few bends. In *Proceedings of the 5th Workshop on Algorithms and Data Structures (WADS 1997)*, pages 354–367, 1997.

[Pur97]   Helen C. Purchase. Which aesthetic has the greatest effect on human understanding? In Giuseppe di Battista, editor, *Proceedings of Graph Drawing Symposium 1997*, volume 1353 of *Lecture Notes in Computer Science (LNCS)*, pages 248–259. Springer-Verlag, 1997.

[Pur00]   Helen C. Purchase. Effective information visualisation: a study of graph drawing aesthetics and algorithms. *Interacting with Computers*, 13(2):477–506, 2000.

[Pur02]   Helen C. Purchase. Metrics for graph drawing aesthetics. *Journal of Visual Languages and Computing*, 13(5):501–516, 2002.

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

[Rec08]   Companhia Brasileira de Trens Urbanos Metrô do Recifè. `http://www.cbtu.gov.br/operadoras/sites/menuprincrec.htm`, 11 September 2008.

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

[Ros04]   Doug Rose. Personal communication, 2004.

[RSOR98]   Alejandro Rosete-Suárez and Alberto Ochoa-Rodríguez. Genetic graph drawing. In *Proceedings of the 13th International Conference of Applications of Artifical Intelligence in Engineering*, pages 37–41, Galway, 1998.

[RSORS99]   Alejandro Rosete-Suárez, Alberto Ochoa-Rodríguez, and Michele Sebag. Automatic graph drawing and stochastic hill climbing. In *Proceedings*

of the Genetic and Evolutionary Computation Conference, volume 2. Morgan Kaufmann, 1999.

[RT81]   Edward M. Reingold and John S. Tilford. Tidier drawing of trees. *IEEE Transactions on Software Engineering*, 7(2):223–228, 1981.

[Saa99]   A. Saalfield. Topologically consistent line simplification with the Douglas-Peucker algorithm. *Cartography and Geographic Information Science*, 26(1):7–18, 1999.

[SCR07]   Sydney CityRail website. `http://www.cityrail.info/`, 1 September 2007.

[Sei91]   K. Seisenberger. Termgraph: Ein system zur zeichnerischen darstellung von structurierten agenten und petrinetzen. Diploma thesis, Universität Passau, 1991.

[SGSK01]   Elmer S. Sandvad, Kaj Grønbæk, Lennert Sloth, and Jorgen Lindskov 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.

[SI03]   Eihan Shimizu and Ryo Inoue. Time-distance mapping: visualization of transportation level of service. In *Proceedings of the Symposium on Environmental Issues Related to Infrastructure Development*, pages 221–230, 2003.

[SKT00]   Janet M. Six, Konstantinos G. Kakoulis, and Ioannis G. Tollis. Techniques for the refinement of orthogonal graph drawings. *Journal of Graph Algorithms and Applications*, 4(3):75–103, 2000.

[SL008]   Storstockholms Lokaltrafik (SL) website. `http://www.sl.se/`, 13 February 2008.

[SM94]   Kozo Sugiyama and Kazuo Misue. A simple and unified method for drawing graphs: magnetic-spring algorithm. In *DIMACS International Workshop, Proceedings of Graph Drawing 1994*, number 894 in Lecture Notes in Computer Science (LNCS), pages 364–375. Springer-Verlag, 1994.

[SM95]   Kozo Sugiyama and Kazuo Misue. Graph drawing by magnetic spring model. *Journal of Visual Languages and Computing*, 6(3):217–231, 1995.

[SR04]   Jonathan Mark Stott and Peter Rodgers. Metro map layout using multicriteria optimization. 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, *Informationan Visualization (IV04)*, pages 355–362, London, England, 2004. IEEE.

[SR05]   Jonathan Mark Stott and Peter Rodgers. Automatic metro map design techniques. In *Proceedings of the 22nd International Cartographic Conference*, A Coruña, Spain, 2005.

[SRB⁺05]   Jonathan Mark Stott, Peter Rodgers, Remo Aslak Burkhard, Michael Meier, and Matthias Thomas Jelle Smis. Automatic layout of project plans using a metro map metaphor. In Ebad Banissi, Muhammad Sarfraz, Jonathan C. Roberts, Bowen Loften, Anna Ursyn, Remo Aslak Burkhard, Angela Lee, and Gennady Andrienko, editors, *Information Visualization (IV05)*, pages 203–206, London, England, 2005. IEEE.

[STT81]   Kozo Sugiyama, Shojiro Tagawa, and Mitsuhiko Toda. Methods for visual understanding of hierarchical systems. *IEEE Transactions on Systems, Man and Cybernetics*, 11(2):109–125, 1981.

[Tam87]   Roberto Tamassia. On embedding a graph in the grid with the minimum number of bends. *SIAM Journal of Computing*, 16(3):421–444, June 1987.

[Tea06]   R Development Core Team. *R: a Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2006.

[Tet98]   Andrea G. B. Tettamanzi. Drawing graphs with evolutionary algorithms. In I. Parmee, editor, *Proceedings of the 1998 Conference on Adaptive Computing in Design and Manufacture*, pages 325–338, April 1998.

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

[TfL07]   Transport for London (bus route spider maps). `http://www.tfl.gov.uk/tfl/gettingaround/maps/buses/`, 18 August 2007.

[TGP03]   Terry M. Therneau, Patricia M. Grambsch, and Vernon S. Pankratz. Penalized survival models and frailty. *Journal of Computational and Graphical Statistics*, 12(1):156–175, 2003.

[The99]   Terry M. Therneau. A package for survival analysis in S. Technical report, Mayo Foundation, 1999.

[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]   Godfried 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.

[TR95]   Robert C. Thomson and Dianne E. Richardson. A graph theory approach to road network generalization. In *Proceedings of the 17th International Cartographic Conference*, pages 1871–1880, 1995.

[TTC08]   Toronto Transit Commission (TTC) website. `http://www.toronto.ca/ttc/index.htm`, 13 February 2008.

[Tun94]   Daniel Tunkelang. A practical approach to drawing undirected graphs. Technical Report CMU-CS-94-161, Carnegie Mellon University, School of Computer Science, 1994.

[vKP98]   Marc J. van Kreveld and Jarno Peschier. On the automated generalization of road network maps. In *Proceedings of the 3rd International Conference on Geocomputation*, University of Bristol, 17 September 1998. GeoComputation CD-ROM.

[vKSW98]   Marc J. van Kreveld, Tycho Strijk, and Alexander 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.

[WB91]   C. V. Wu and Barbara P. Buttenfield. Reconsidering rules for point-feature name placement. *Cartographica: the international journal for geographic information and geovisualization*, 28(1):10–27, 1991.

[Wei96]   Robert Weibel. A typology of constraints to line simplification. In M. J. Kraak and M. Molenaar, editors, *Proceedings of the 7th International Symposium on Spatial Data Handling*, pages 9A.1–9A.14, 1996.

[WF94]   Stanley Wasserman and Katherine Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.

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

[WMA08]   Washington Metropolitan Area Transit Authority (WMATA) website. `http://www.wmata.com/metrorail/systemmap.cfm`, 13 February 2008.

[Woo82]   Donald R. Woods. *Drawing planar graphs.* PhD thesis, Department of Computer Science, Stanford University, 1982.

[WPCM02]   Colin Ware, Helen Purchase, Linda Colpoys, and Matthew McGill. Cognitive measurements of graph aesthetics. *Information Visualization*, 1(2):103–110, 2002.

[WS07]   Alexander Wolff and Tycho Strijk. The map labeling bibliography. `http://i11www.iti.uni-karlsruhe.de/~awolff/map-labeling/bibliography/`, 5 2007.

[YH98]   Paul Yates and Mark Humphries. The generation of schematic diagrams from geographic representations of networks. In *Proceedings of the Spatial Information Research Centre's 10th Colloquium*, 1998.

[Yoe72]   Pinhas Yoeli. The logic of automated map lettering. *The Cartographic Journal*, 9(2):99–108, 1972.

[Zie04]   Stephan Ziemer. *Anwendung der Metro-Map-Metapher auf die Gestaltung von Unternehmensportalen (Application of the Metro Map Metaphor to the Organization of Enterprise Portals).* PhD thesis, Fakultät für Informatik Technischen, Universität München, 2004.

[Zor86]   Steven Zoraster. Integer programming applied to the map label placement problem. *Cartographica: the international journal for geographic information and geovisualization*, 23(3):16–27, 1986.

[Zor90]   Steven Zoraster. The solution of large 0-1 integer programming problems encountered in automated cartography. *Operations Research*, 38(5):752–759, 1990.

Appendix A

Empirical Evaluation

A.1 Questions and Maps Used

A.1.1 Atlanta

- Geographic map (Figure A.1).

- Normalised published map (Figure A.2).

- Automatically-drawn map (Figure A.3).

Table A.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 A.1: Atlanta MARTA geographic map.

Figure A.2: Atlanta MARTA normalised published map.

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

A.1.2 Bucharest

- Geographic map (Figure A.4).

- Normalised published map (Figure A.5).

- Automatically-drawn map (Figure A.6).

Table A.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
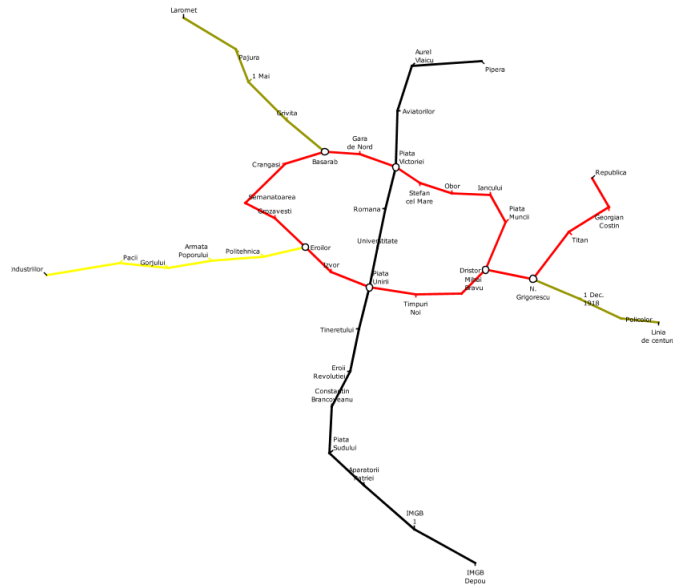
Figure A.4: Bucharest geographic map.
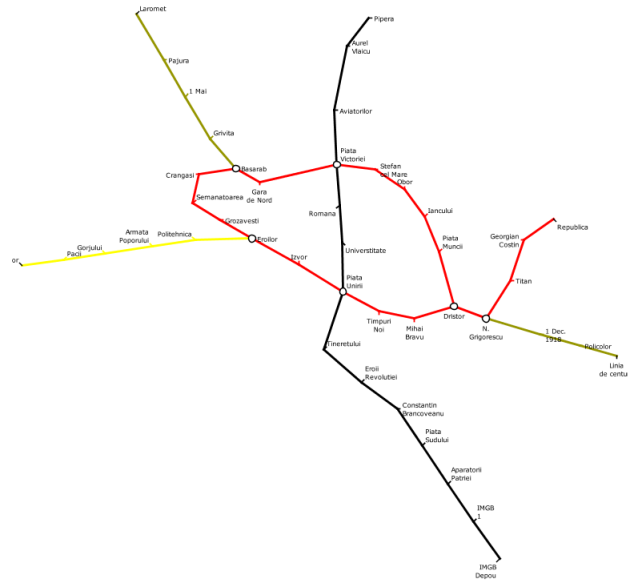
Figure A.5: Bucharest normalised published map.
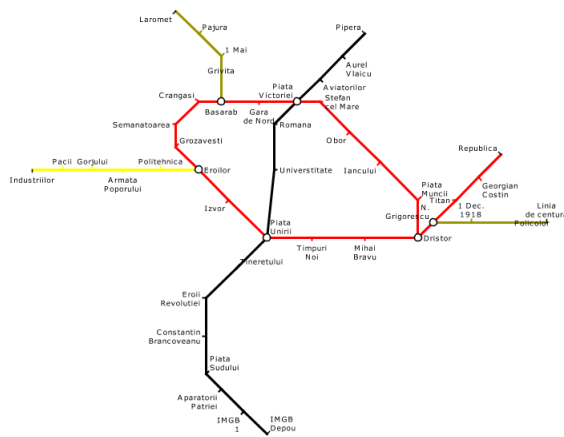
Figure A.6: Bucharest automatically-drawn map.

A.1.3 Mexico City

- Geographic map (Figure A.7).

- Normalised published map (Figure A.8).

- Automatically-drawn map (Figure A.9).

Table A.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
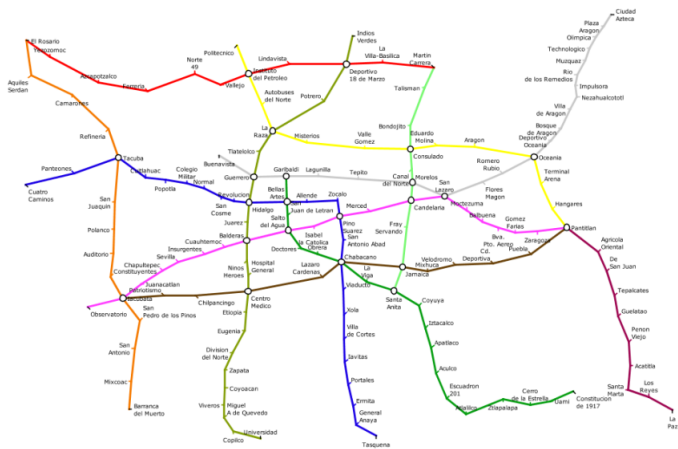
Figure A.7: Mexico City geographic map.

Figure A.8: Mexico City normalised published map.
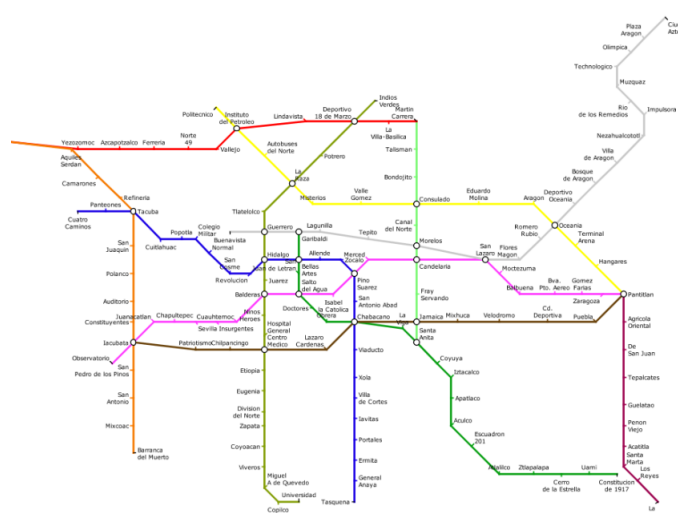
Figure A.9: Mexico City automatically-drawn map.

A.1.4 Stockholm

- Geographic map (Figure A.10).

- Normalised published map (Figure A.11).

- Automatically-drawn map (Figure A.12).

Table A.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 'Kungstrådgarden'	$\{2, 3, 4, 5, 6\}$	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 A.10: Stockholm geographic map.

Figure A.11: Stockholm normalised published map.
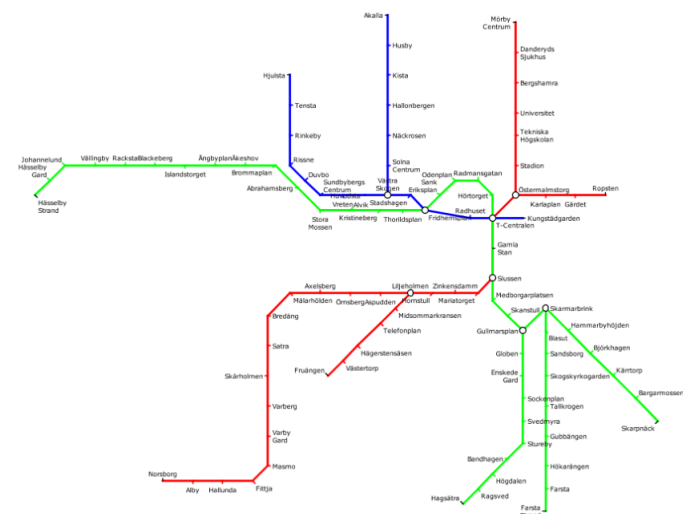
Figure A.12: Stockholm automatically-drawn map.

A.1.5 Toronto

- Geographic map (Figure A.13).

- Normalised published map (Figure A.14).

- Automatically-drawn map (Figure A.15).

Table A.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 A.13: Toronto geographic map.

Figure A.14: Toronto normalised published map.

Figure A.15: Toronto automatically-drawn map.

A.1.6 Washington D.C.

- Geographic map (Figure A.16).

- Normalised published map (Figure A.17).

- Automatically-drawn map (Figure A.18).

Table A.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
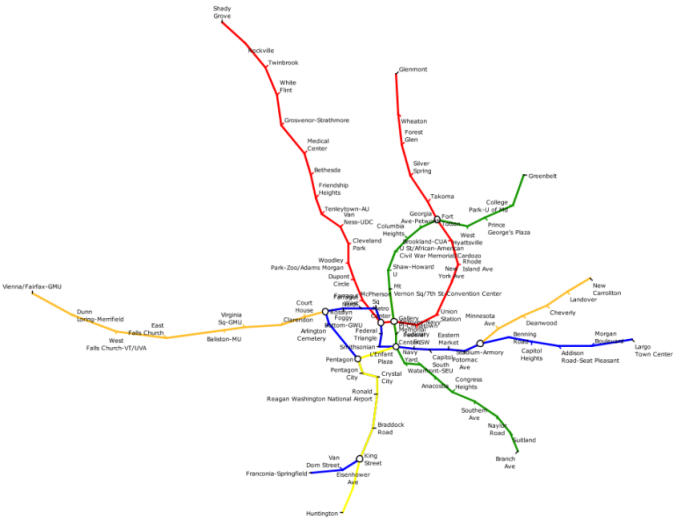
Figure A.16: Washington D.C. geographic map.
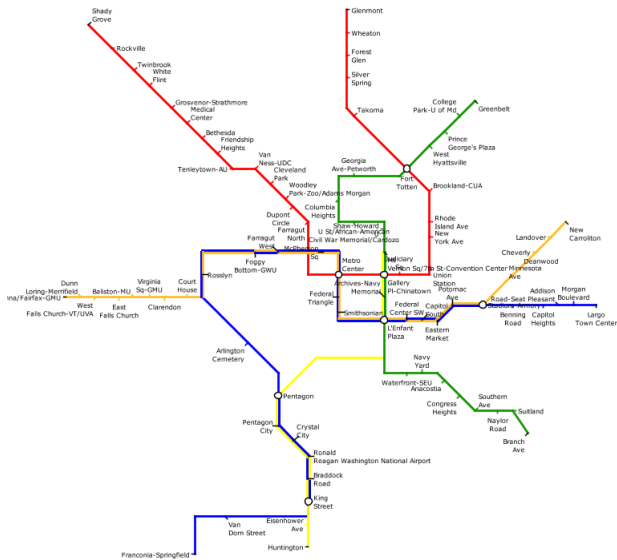
Figure A.17: Washington D.C. normalised published map.
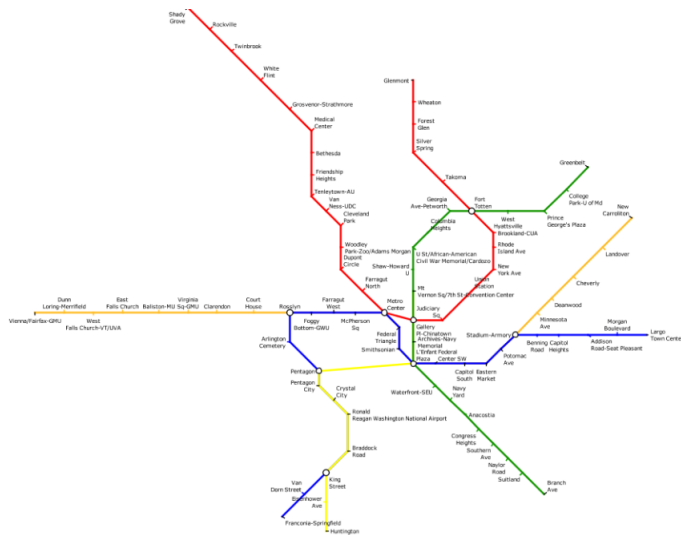
Figure A.18: Washington D.C. automatically-drawn map.

A.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.

A.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 Aeroporto 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 do not 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.

A.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].

A.2.3   Questionnaire

This is the questionnaire that each candidate was asked to fill in in relation to the postliminary script (Section A.2.2).

Your Login:

Please first fill in your login above, 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. First write down your login, then, 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.

Table A.7: Preference grid.

Slide	Preference for Map A	Preference for Map B	Preference for Map C
Slide 1			
Slide 2			
Slide 3			
Slide 4			
Slide 5			
Slide 6			

Please now take five minutes to answer the questions below

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

2. Which features of the metro map layout did you find most helpful when completing the tests?

3. Which features of the metro map layout did you find least helpful when completing the tests?

4. Did you find any of the questions ambiguous?

A.2.4   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.

A.3 Results

A.3.1 Route Planning Tasks

Table A.8: Route planning task results.

Subj.	Grp	Map[a]	Qu.	Map Type[c]	Answer	Given Answer	Correct	Ease[b]	Time (s)
1	C	1	a	F	10	11	FALSE	3	38.58
2	C	1	a	F	10	10	TRUE	2	24.06
3	C	1	a	F	10	10	TRUE	0	26.58
4	A	1	a	P	10	10	TRUE	1	17.92
5	B	1	a	G	10	10	TRUE	2	16.81

[a]1 = Atlanta; 2 = Bucharest; 3 = Mexico City; 4 = Stockholm; 5 = Toronto; 6 = Washington D.C.
[b]1 = very easy ... 5 = very hard
[c]G = geographic layout; P = normalised published layout; F = automatically-drawn layout

Subj.	Grp	Map	Qu.	Map Type	Answer	Given Answer	Correct	Ease	Time (s)
6	B	1	a	G	10	9	FALSE	2	17.7
7	B	1	a	G	10	10	TRUE	2	27.74
8	A	1	a	P	10	10	TRUE	1	17.06
9	A	1	a	P	10	10	TRUE	3	24.3
10	A	1	a	P	10	10	TRUE	2	27.86
11	C	1	a	F	10	10	TRUE	3	32.45
12	B	1	a	G	10	10	TRUE	2	26.49
13	C	1	a	F	10	10	TRUE	2	25.66
14	A	1	a	P	10	10	TRUE	2	16.92
15	A	1	a	P	10	10	TRUE	1	17.8
16	C	1	a	F	10	10	TRUE	2	18.69
17	B	1	a	G	10	11	FALSE	2	28.81
18	B	1	a	G	10	10	TRUE	0	38.89
19	C	1	a	F	10	10	TRUE	1	16.56
20	C	1	a	F	10	10	TRUE	2	19.75
21	A	1	a	P	10	10	TRUE	1	21.17
22	B	1	a	G	10	10	TRUE	2	15.31
23	A	1	a	P	10	10	TRUE	2	21.58
24	B	1	a	G	10	9	FALSE	1	26.28
25	C	1	a	F	10	10	TRUE	1	14.83
26	B	1	a	G	10	9	FALSE	1	20
27	A	1	a	P	10	10	TRUE	0	34.31
28	B	1	a	G	10	9	FALSE	2	18.05
29	A	1	a	P	10	10	TRUE	2	14.94
30	C	1	a	F	10	10	TRUE	3	19.25
31	A	1	a	P	10	10	TRUE	2	11.2
32	C	1	a	F	10	10	TRUE	2	13.49
33	B	1	a	G	10	11	FALSE	2	17.39
34	B	1	a	G	10	10	TRUE	0	27.56
35	C	1	a	F	10	10	TRUE	3	13.39
36	C	1	a	F	10	11	FALSE	3	18.11
37	B	1	a	G	10	9	FALSE	2	32.55
38	A	1	a	P	10	10	TRUE	2	15.2
39	A	1	a	P	10	12	FALSE	4	15.83
40	B	1	a	G	10	10	TRUE	0	18.97
41	B	1	a	G	10	11	FALSE	2	18.28
42	C	1	a	F	10	10	TRUE	3	20.02
43	C	1	a	F	10	10	TRUE	1	19.9
1	C	1	b	G	5	5	TRUE	2	30.66
2	C	1	b	G	5	5	TRUE	2	18.19
3	C	1	b	G	5	5	TRUE	3	13.16
4	A	1	b	F	5	4	FALSE	4	11.05
5	B	1	b	P	5	5	TRUE	2	16.36
6	B	1	b	P	5	5	TRUE	0	10.89
7	B	1	b	P	5	5	TRUE	1	13.72
8	A	1	b	F	5	5	TRUE	1	9
9	A	1	b	F	5	5	TRUE	2	20.58
10	A	1	b	F	5	5	TRUE	2	19.45

Subj.	Grp	Map	Qu.	Map Type	Answer	Given Answer	Correct	Ease	Time (s)
11	C	1	b	G	5	5	TRUE	2	17.2
12	B	1	b	P	5	5	TRUE	2	23.63
13	C	1	b	G	5	5	TRUE	2	10.81
14	A	1	b	F	5	5	TRUE	2	11.59
15	A	1	b	F	5	5	TRUE	2	12.2
16	C	1	b	G	5	5	TRUE	1	17.98
17	B	1	b	P	5	5	TRUE	1	17.17
18	B	1	b	P	5	5	TRUE	1	12.1
19	C	1	b	G	5	5	TRUE	1	19.91
20	C	1	b	G	5	5	TRUE	2	15.89
21	A	1	b	F	5	5	TRUE	1	12.12
22	B	1	b	P	5	5	TRUE	1	9.23
23	A	1	b	F	5	8	FALSE	2	19.34
24	B	1	b	P	5	5	TRUE	1	13.24
25	C	1	b	G	5	5	TRUE	2	9.8
26	B	1	b	P	5	5	TRUE	0	15.5
27	A	1	b	F	5	5	TRUE	1	19.21
28	B	1	b	P	5	5	TRUE	2	10.57
29	A	1	b	F	5	5	TRUE	3	15.56
30	C	1	b	G	5	5	TRUE	3	18.3
31	A	1	b	F	5	5	TRUE	2	10.03
32	C	1	b	G	5	5	TRUE	2	10.37
33	B	1	b	P	5	5	TRUE	1	10.56
34	B	1	b	P	5	5	TRUE	0	14.41
35	C	1	b	G	5	5	TRUE	2	13.25
36	C	1	b	G	5	5	TRUE	1	9.58
37	B	1	b	P	5	5	TRUE	3	12.69
38	A	1	b	F	5	5	TRUE	4	12.98
39	A	1	b	F	5	8	FALSE	4	10.41
40	B	1	b	P	5	5	TRUE	0	8.29
41	B	1	b	P	5	5	TRUE	1	6.79
42	C	1	b	G	5	5	TRUE	1	12.99
43	C	1	b	G	5	5	TRUE	1	15.99
1	C	1	c	F	0	0	TRUE	2	13.69
2	C	1	c	F	0	1	FALSE	2	25.27
3	C	1	c	F	0	0	TRUE	2	26.76
4	A	1	c	P	0	0	TRUE	1	13.69
5	B	1	c	G	0	0	TRUE	0	24.88
6	B	1	c	G	0	0	TRUE	0	25.63
7	B	1	c	G	0	0	TRUE	1	16.64
8	A	1	c	P	0	0	TRUE	1	12.08
9	A	1	c	P	0	0	TRUE	1	11.97
10	A	1	c	P	0	1	FALSE	0	13.61
11	C	1	c	F	0	0	TRUE	3	35.59
12	B	1	c	G	0	1	FALSE	3	86.96
13	C	1	c	F	0	0	TRUE	2	18.69
14	A	1	c	P	0	0	TRUE	1	9.45
15	A	1	c	P	0	0	TRUE	1	8.78

continued from previous page

Subj.	Grp	Map	Qu.	Map Type	Answer	Given Answer	Correct	Ease	Time (s)
16	C	1	c	F	0	0	TRUE	1	22.61
17	B	1	c	G	0	1	FALSE	3	44.19
18	B	1	c	G	0	1	FALSE	1	29.71
19	C	1	c	F	0	0	TRUE	2	44.86
20	C	1	c	F	0	1	FALSE	1	25.58
21	A	1	c	P	0	0	TRUE	0	32.6
22	B	1	c	G	0	0	TRUE	1	15.13
23	A	1	c	P	0	0	TRUE	2	30.73
24	B	1	c	G	0	0	TRUE	2	51.58
25	C	1	c	F	0	0	TRUE	0	15.25
26	B	1	c	G	0	2	FALSE	0	12.22
27	A	1	c	P	0	0	TRUE	0	11.53
28	B	1	c	G	0	0	TRUE	2	23.15
29	A	1	c	P	0	0	TRUE	1	18
30	C	1	c	F	0	0	TRUE	2	28.92
31	A	1	c	P	0	0	TRUE	1	14.54
32	C	1	c	F	0	0	TRUE	1	14.08
33	B	1	c	G	0	0	TRUE	0	13.3
34	B	1	c	G	0	1	FALSE	1	35.03
35	C	1	c	F	0	0	TRUE	1	18.58
36	C	1	c	F	0	0	TRUE	2	21.22
37	B	1	c	G	0	1	FALSE	2	33.63
38	A	1	c	P	0	0	TRUE	2	14.06
39	A	1	c	P	0	0	TRUE	1	9.84
40	B	1	c	G	0	0	TRUE	0	17.45
41	B	1	c	G	0	0	TRUE	2	15.74
42	C	1	c	F	0	0	TRUE	1	19.63
43	C	1	c	F	0	0	TRUE	1	11.98
1	C	1	d	G	0	0	TRUE	0	9.53
2	C	1	d	G	0	0	TRUE	2	14.95
3	C	1	d	G	0	0	TRUE	3	31.44
4	A	1	d	F	0	0	TRUE	0	13.81
5	B	1	d	P	0	0	TRUE	0	10.03
6	B	1	d	P	0	0	TRUE	0	11.39
7	B	1	d	P	0	0	TRUE	0	18.94
8	A	1	d	F	0	0	TRUE	1	10.94
9	A	1	d	F	0	0	TRUE	1	10.31
10	A	1	d	F	0	0	TRUE	0	11
11	C	1	d	G	0	0	TRUE	1	19.54
12	B	1	d	P	0	0	TRUE	0	29.94
13	C	1	d	G	0	0	TRUE	0	10.58
14	A	1	d	F	0	0	TRUE	0	7.64
15	A	1	d	F	0	0	TRUE	2	11.28
16	C	1	d	G	0	0	TRUE	0	11.92
17	B	1	d	P	0	0	TRUE	0	15.05
18	B	1	d	P	0	1	FALSE	1	17.28
19	C	1	d	G	0	0	TRUE	1	19.55
20	C	1	d	G	0	0	TRUE	2	14.53

continued from previous page

Subj.	Grp	Map	Qu.	Map Type	Answer	Given Answer	Correct	Ease	Time (s)
21	A	1	d	F	0	0	TRUE	0	13.58
22	B	1	d	P	0	0	TRUE	1	11.55
23	A	1	d	F	0	0	TRUE	1	17.26
24	B	1	d	P	0	0	TRUE	0	14.77
25	C	1	d	G	0	0	TRUE	0	8.74
26	B	1	d	P	0	0	TRUE	0	16.98
27	A	1	d	F	0	0	TRUE	0	14.01
28	B	1	d	P	0	0	TRUE	2	13.16
29	A	1	d	F	0	0	TRUE	1	16.08
30	C	1	d	G	0	0	TRUE	1	15.2
31	A	1	d	F	0	0	TRUE	1	10.47
32	C	1	d	G	0	0	TRUE	1	14.4
33	B	1	d	P	0	0	TRUE	0	6
34	B	1	d	P	0	0	TRUE	0	14.46
35	C	1	d	G	0	0	TRUE	0	13
36	C	1	d	G	0	0	TRUE	0	10.84
37	B	1	d	P	0	0	TRUE	3	19.81
38	A	1	d	F	0	0	TRUE	2	13.12
39	A	1	d	F	0	0	TRUE	1	20.56
40	B	1	d	P	0	0	TRUE	0	18.54
41	B	1	d	P	0	0	TRUE	0	5.73
42	C	1	d	G	0	0	TRUE	0	13.73
43	C	1	d	G	0	0	TRUE	0	12.76
1	C	2	a	G	10	10	TRUE	2	32.38
2	C	2	a	G	10	9	FALSE	2	37.41
3	C	2	a	G	10	9	FALSE	3	24.21
4	A	2	a	F	10	10	TRUE	2	41.6
5	B	2	a	P	10	10	TRUE	2	15.03
6	B	2	a	P	10	10	TRUE	1	13.81
7	B	2	a	P	10	10	TRUE	2	29.5
8	A	2	a	F	10	10	TRUE	2	34.1
9	A	2	a	F	10	10	TRUE	3	66.83
10	A	2	a	F	10	10	TRUE	2	34.36
11	C	2	a	G	10	9	FALSE	3	37.26
12	B	2	a	P	10	10	TRUE	2	31.55
13	C	2	a	G	10	10	TRUE	3	21.75
14	A	2	a	F	10	10	TRUE	3	39.02
15	A	2	a	F	10	10	TRUE	1	13.33
16	C	2	a	G	10	9	FALSE	1	15.56
17	B	2	a	P	10	10	TRUE	2	16.61
18	B	2	a	P	10	10	TRUE	0	57.44
19	C	2	a	G	10	10	TRUE	2	43.83
20	C	2	a	G	10	9	FALSE	2	13.53
21	A	2	a	F	10	10	TRUE	3	28.84
22	B	2	a	P	10	10	TRUE	2	13.2
23	A	2	a	F	10	10	TRUE	3	50.81
24	B	2	a	P	10	10	TRUE	2	13.72
25	C	2	a	G	10	9	FALSE	1	23.66

continued from previous page

Subj.	Grp	Map	Qu.	Map Type	Answer	Given Answer	Correct	Ease	Time (s)
26	B	2	a	P	10	10	TRUE	2	23.53
27	A	2	a	F	10	10	TRUE	0	45.08
28	B	2	a	P	10	10	TRUE	2	16.73
29	A	2	a	F	10	11	FALSE	3	25.43
30	C	2	a	G	10	9	FALSE	3	18.77
31	A	2	a	F	10	10	TRUE	3	35.23
32	C	2	a	G	10	10	TRUE	2	18.1
33	B	2	a	P	10	10	TRUE	2	11.44
34	B	2	a	P	10	11	FALSE	0	51.63
35	C	2	a	G	10	9	FALSE	2	37.02
36	C	2	a	G	10	10	TRUE	3	23.77
37	B	2	a	P	10	10	TRUE	2	12.97
38	A	2	a	F	10	10	TRUE	2	10.45
39	A	2	a	F	10	10	TRUE	2	12.19
40	B	2	a	P	10	10	TRUE	0	21.25
41	B	2	a	P	10	11	FALSE	2	18.17
42	C	2	a	G	10	10	TRUE	3	16.73
43	C	2	a	G	10	9	FALSE	1	22.77
1	C	2	b	P	5	5	TRUE	1	13.38
2	C	2	b	P	5	5	TRUE	2	17.33
3	C	2	b	P	5	5	TRUE	3	18.78
4	A	2	b	G	5	5	TRUE	0	9.81
5	B	2	b	F	5	5	TRUE	2	10.02
6	B	2	b	F	5	5	TRUE	2	10.63
7	B	2	b	F	5	5	TRUE	2	27.35
8	A	2	b	G	5	5	TRUE	1	15.11
9	A	2	b	G	5	5	TRUE	2	22.17
10	A	2	b	G	5	5	TRUE	1	9.27
11	C	2	b	P	5	5	TRUE	1	20.68
12	B	2	b	F	5	5	TRUE	2	32.69
13	C	2	b	P	5	5	TRUE	1	20.23
14	A	2	b	G	5	5	TRUE	2	9.64
15	A	2	b	G	5	5	TRUE	2	14.25
16	C	2	b	P	5	5	TRUE	0	9.83
17	B	2	b	F	5	5	TRUE	1	21.17
18	B	2	b	F	5	5	TRUE	2	16.97
19	C	2	b	P	5	5	TRUE	1	26.04
20	C	2	b	P	5	5	TRUE	2	13.6
21	A	2	b	G	5	3	FALSE	3	25.31
22	B	2	b	F	5	5	TRUE	1	8.3
23	A	2	b	G	5	5	TRUE	1	34.8
24	B	2	b	F	5	5	TRUE	1	10.86
25	C	2	b	P	5	5	TRUE	1	10.05
26	B	2	b	F	5	5	TRUE	0	14.2
27	A	2	b	G	5	5	TRUE	2	18.4
28	B	2	b	F	5	4	FALSE	2	8.95
29	A	2	b	G	5	5	TRUE	2	21.57
30	C	2	b	P	5	5	TRUE	1	10.67

continued from previous page

Subj.	Grp	Map	Qu.	Map Type	Answer	Given Answer	Correct	Ease	Time (s)
31	A	2	b	G	5	5	TRUE	3	32.85
32	C	2	b	P	5	5	TRUE	1	14.7
33	B	2	b	F	5	5	TRUE	1	12.67
34	B	2	b	F	5	5	TRUE	0	30.66
35	C	2	b	P	5	5	TRUE	1	16.52
36	C	2	b	P	5	5	TRUE	1	7.38
37	B	2	b	F	5	5	TRUE	2	9.59
38	A	2	b	G	5	5	TRUE	3	10
39	A	2	b	G	5	5	TRUE	4	20.25
40	B	2	b	F	5	5	TRUE	0	10.27
41	B	2	b	F	5	5	TRUE	1	7.79
42	C	2	b	P	5	5	TRUE	1	19.57
43	C	2	b	P	5	5	TRUE	1	11.88
1	C	2	c	G	0	0	TRUE	1	12.75
2	C	2	c	G	0	0	TRUE	2	26.02
3	C	2	c	G	0	0	TRUE	2	22.15
4	A	2	c	F	0	0	TRUE	0	14.13
5	B	2	c	P	0	0	TRUE	1	19.09
6	B	2	c	P	0	0	TRUE	0	22.81
7	B	2	c	P	0	0	TRUE	1	34.25
8	A	2	c	F	0	0	TRUE	1	12.95
9	A	2	c	F	0	0	TRUE	1	29.66
10	A	2	c	F	0	0	TRUE	1	12.83
11	C	2	c	G	0	0	TRUE	3	48.67
12	B	2	c	P	0	0	TRUE	2	52.5
13	C	2	c	G	0	2	FALSE	3	23.83
14	A	2	c	F	0	0	TRUE	1	8.03
15	A	2	c	F	0	0	TRUE	1	18.67
16	C	2	c	G	0	0	TRUE	0	9.64
17	B	2	c	P	0	0	TRUE	2	35.5
18	B	2	c	P	0	3	FALSE	2	45.7
19	C	2	c	G	0	0	TRUE	2	27.77
20	C	2	c	G	0	0	TRUE	1	19.21
21	A	2	c	F	0	0	TRUE	1	17.86
22	B	2	c	P	0	0	TRUE	1	22.52
23	A	2	c	F	0	0	TRUE	2	44.15
24	B	2	c	P	0	1	FALSE	3	42.82
25	C	2	c	G	0	0	TRUE	1	10.88
26	B	2	c	P	0	2	FALSE	1	12.77
27	A	2	c	F	0	0	TRUE	0	10.42
28	B	2	c	P	0	0	TRUE	2	34.99
29	A	2	c	F	0	0	TRUE	1	12.67
30	C	2	c	G	0	0	TRUE	2	19.41
31	A	2	c	F	0	0	TRUE	1	16.62
32	C	2	c	G	0	0	TRUE	1	9.41
33	B	2	c	P	0	0	TRUE	0	11.03
34	B	2	c	P	0	0	TRUE	0	38.6
35	C	2	c	G	0	0	TRUE	1	18.13

continued from previous page

Subj.	Grp	Map	Qu.	Map Type	Answer	Given Answer	Correct	Ease	Time (s)
36	C	2	c	G	0	0	TRUE	2	23.59
37	B	2	c	P	0	0	TRUE	1	12.63
38	A	2	c	F	0	0	TRUE	3	24.25
39	A	2	c	F	0	0	TRUE	1	9.23
40	B	2	c	P	0	1	FALSE	0	14.5
41	B	2	c	P	0	0	TRUE	2	14.72
42	C	2	c	G	0	0	TRUE	1	29.35
43	C	2	c	G	0	0	TRUE	0	7.28
1	C	2	d	P	2	3	FALSE	1	15.89
2	C	2	d	P	2	2	TRUE	2	23.31
3	C	2	d	P	2	2	TRUE	2	20.6
4	A	2	d	G	2	2	TRUE	0	15.09
5	B	2	d	F	2	2	TRUE	2	15.81
6	B	2	d	F	2	2	TRUE	1	10.14
7	B	2	d	F	2	2	TRUE	2	24.58
8	A	2	d	G	2	2	TRUE	1	13.8
9	A	2	d	G	2	2	TRUE	2	31.27
10	A	2	d	G	2	2	TRUE	3	37.78
11	C	2	d	P	2	2	TRUE	2	26.7
12	B	2	d	F	2	3	FALSE	2	42.46
13	C	2	d	P	2	2	TRUE	2	27.95
14	A	2	d	G	2	2	TRUE	2	17.31
15	A	2	d	G	2	2	TRUE	2	9.55
16	C	2	d	P	2	2	TRUE	1	15.19
17	B	2	d	F	2	2	TRUE	1	19.3
18	B	2	d	F	2	3	FALSE	1	15.24
19	C	2	d	P	2	2	TRUE	2	44.14
20	C	2	d	P	2	2	TRUE	2	23.82
21	A	2	d	G	2	2	TRUE	2	25.59
22	B	2	d	F	2	2	TRUE	1	14.05
23	A	2	d	G	2	2	TRUE	3	56.6
24	B	2	d	F	2	2	TRUE	1	14.55
25	C	2	d	P	2	2	TRUE	0	11.38
26	B	2	d	F	2	2	TRUE	1	21.19
27	A	2	d	G	2	2	TRUE	0	22.13
28	B	2	d	F	2	2	TRUE	2	19.93
29	A	2	d	G	2	2	TRUE	2	19.7
30	C	2	d	P	2	2	TRUE	2	22.34
31	A	2	d	G	2	2	TRUE	2	19.56
32	C	2	d	P	2	2	TRUE	2	14.51
33	B	2	d	F	2	2	TRUE	2	20.05
34	B	2	d	F	2	2	TRUE	2	35.46
35	C	2	d	P	2	2	TRUE	1	16.72
36	C	2	d	P	2	3	FALSE	2	21.63
37	B	2	d	F	2	2	TRUE	3	21.41
38	A	2	d	G	2	2	TRUE	2	14.23
39	A	2	d	G	2	2	TRUE	2	16.5
40	B	2	d	F	2	2	TRUE	0	16.23

Subj.	Grp	Map	Qu.	Map Type	Answer	Given Answer	Correct	Ease	Time (s)
41	B	2	d	F	2	2	TRUE	1	11.75
42	C	2	d	P	2	2	TRUE	2	13.54
43	C	2	d	P	2	2	TRUE	1	19.52
1	C	3	a	P	7	8	FALSE	2	25.19
2	C	3	a	P	7	7	TRUE	2	76.92
3	C	3	a	P	7	6	FALSE	3	22.82
4	A	3	a	G	7	7	TRUE	0	22.19
5	B	3	a	F	7	7	TRUE	2	14.41
6	B	3	a	F	7	6	FALSE	2	20.64
7	B	3	a	F	7	7	TRUE	2	23.11
8	A	3	a	G	7	7	TRUE	2	17.74
9	A	3	a	G	7	7	TRUE	3	28.7
10	A	3	a	G	7	7	TRUE	2	29.11
11	C	3	a	P	7	7	TRUE	4	23.75
12	B	3	a	F	7	6	FALSE	2	26.05
13	C	3	a	P	7	6	FALSE	3	17.27
14	A	3	a	G	7	7	TRUE	3	25.67
15	A	3	a	G	7	7	TRUE	1	16.33
16	C	3	a	P	7	5	FALSE	1	26.06
17	B	3	a	F	7	7	TRUE	2	37.99
18	B	3	a	F	7	6	FALSE	2	59.47
19	C	3	a	P	7	6	FALSE	2	50.61
20	C	3	a	P	7	7	TRUE	2	18.55
21	A	3	a	G	7	7	TRUE	3	34.97
22	B	3	a	F	7	7	TRUE	2	22
23	A	3	a	G	7	7	TRUE	2	33.78
24	B	3	a	F	7	7	TRUE	2	54.94
25	C	3	a	P	7	7	TRUE	1	34.97
26	B	3	a	F	7	7	TRUE	2	42.17
27	A	3	a	G	7	8	FALSE	1	37.19
28	B	3	a	F	7	7	TRUE	2	15.66
29	A	3	a	G	7	7	TRUE	3	16.89
30	C	3	a	P	7	7	TRUE	3	18.61
31	A	3	a	G	7	7	TRUE	3	29.13
32	C	3	a	P	7	8	FALSE	2	16.2
33	B	3	a	F	7	7	TRUE	3	39.03
34	B	3	a	F	7	6	FALSE	0	25.92
35	C	3	a	P	7	7	TRUE	2	19.27
36	C	3	a	P	7	6	FALSE	3	24.99
37	B	3	a	F	7	7	TRUE	3	13.85
38	A	3	a	G	7	5	FALSE	2	13.03
39	A	3	a	G	7	7	TRUE	2	9.31
40	B	3	a	F	7	6	FALSE	0	21.19
41	B	3	a	F	7	6	FALSE	1	15.74
42	C	3	a	P	7	7	TRUE	4	18.03
43	C	3	a	P	7	7	TRUE	1	16.84
1	C	3	b	F	6	6	TRUE	2	16.89
2	C	3	b	F	6	6	TRUE	2	30.98

Subj.	Grp	Map	Qu.	Map Type	Answer	Given Answer	Correct	Ease	Time (s)
3	C	3	b	F	6	6	TRUE	3	19.6
4	A	3	b	P	6	5	FALSE	0	11.84
5	B	3	b	G	6	6	TRUE	1	15.59
6	B	3	b	G	6	6	TRUE	2	12.05
7	B	3	b	G	6	6	TRUE	1	16.3
8	A	3	b	P	6	6	TRUE	1	11.86
9	A	3	b	P	6	6	TRUE	2	23.05
10	A	3	b	P	6	6	TRUE	1	12.42
11	C	3	b	F	6	6	TRUE	2	12.25
12	B	3	b	G	6	6	TRUE	2	34.39
13	C	3	b	F	6	6	TRUE	3	18.25
14	A	3	b	P	6	6	TRUE	1	13.7
15	A	3	b	P	6	6	TRUE	2	13.52
16	C	3	b	F	6	8	FALSE	2	26.55
17	B	3	b	G	6	6	TRUE	2	20.41
18	B	3	b	G	6	6	TRUE	2	12.42
19	C	3	b	F	6	5	FALSE	2	41.38
20	C	3	b	F	6	6	TRUE	2	12.69
21	A	3	b	P	6	6	TRUE	1	17.81
22	B	3	b	G	6	6	TRUE	2	9.47
23	A	3	b	P	6	7	FALSE	1	16.95
24	B	3	b	G	6	6	TRUE	1	15.94
25	C	3	b	F	6	6	TRUE	1	17.23
26	B	3	b	G	6	6	TRUE	1	12.91
27	A	3	b	P	6	6	TRUE	1	16.07
28	B	3	b	G	6	6	TRUE	2	11.5
29	A	3	b	P	6	6	TRUE	2	11.26
30	C	3	b	F	6	6	TRUE	1	15.73
31	A	3	b	P	6	6	TRUE	2	18.48
32	C	3	b	F	6	6	TRUE	2	12.8
33	B	3	b	G	6	6	TRUE	1	7.44
34	B	3	b	G	6	6	TRUE	0	51.39
35	C	3	b	F	6	6	TRUE	1	11.81
36	C	3	b	F	6	6	TRUE	1	10.56
37	B	3	b	G	6	6	TRUE	3	11.71
38	A	3	b	P	6	6	TRUE	3	11.7
39	A	3	b	P	6	6	TRUE	3	11.98
40	B	3	b	G	6	6	TRUE	0	15.47
41	B	3	b	G	6	6	TRUE	2	12.21
42	C	3	b	F	6	6	TRUE	3	17.12
43	C	3	b	F	6	6	TRUE	1	13.93
1	C	3	c	P	2	2	TRUE	3	27.19
2	C	3	c	P	2	2	TRUE	3	129.64
3	C	3	c	P	2	1	FALSE	2	20.68
4	A	3	c	G	2	1	FALSE	1	14.05
5	B	3	c	F	2	1	FALSE	1	17.56
6	B	3	c	F	2	2	TRUE	2	18.23
7	B	3	c	F	2	1	FALSE	2	23.77

continued from previous page

Subj.	Grp	Map	Qu.	Map Type	Answer	Given Answer	Correct	Ease	Time (s)
8	A	3	c	G	2	1	FALSE	1	14.17
9	A	3	c	G	2	2	TRUE	2	21.62
10	A	3	c	G	2	1	FALSE	1	11.92
11	C	3	c	P	2	2	TRUE	2	51.46
12	B	3	c	F	2	2	TRUE	3	45.05
13	C	3	c	P	2	2	TRUE	1	24.84
14	A	3	c	G	2	1	FALSE	2	23.5
15	A	3	c	G	2	3	FALSE	2	13.08
16	C	3	c	P	2	2	TRUE	1	22.39
17	B	3	c	F	2	1	FALSE	1	36.31
18	B	3	c	F	2	1	FALSE	0	28.61
19	C	3	c	P	2	1	FALSE	2	26.91
20	C	3	c	P	2	2	TRUE	2	17.6
21	A	3	c	G	2	1	FALSE	2	26.41
22	B	3	c	F	2	1	FALSE	1	18.94
23	A	3	c	G	2	2	TRUE	2	37.69
24	B	3	c	F	2	2	TRUE	2	42.57
25	C	3	c	P	2	1	FALSE	1	15.48
26	B	3	c	F	2	2	TRUE	2	170.64
27	A	3	c	G	2	1	FALSE	2	38.49
28	B	3	c	F	2	2	TRUE	2	40.19
29	A	3	c	G	2	1	FALSE	2	21.89
30	C	3	c	P	2	1	FALSE	2	24
31	A	3	c	G	2	1	FALSE	2	28.07
32	C	3	c	P	2	1	FALSE	1	13.14
33	B	3	c	F	2	2	TRUE	2	41.77
34	B	3	c	F	2	1	FALSE	0	30.15
35	C	3	c	P	2	1	FALSE	2	22.74
36	C	3	c	P	2	1	FALSE	2	33.17
37	B	3	c	F	2	1	FALSE	1	19.94
38	A	3	c	G	2	1	FALSE	2	13.9
39	A	3	c	G	2	2	TRUE	2	14.66
40	B	3	c	F	2	2	TRUE	0	12.78
41	B	3	c	F	2	2	TRUE	2	22.74
42	C	3	c	P	2	1	FALSE	2	15.12
43	C	3	c	P	2	1	FALSE	2	17.93
1	C	3	d	F	2	3	FALSE	2	53.73
2	C	3	d	F	2	3	FALSE	2	30.96
3	C	3	d	F	2	2	TRUE	3	30.39
4	A	3	d	P	2	2	TRUE	0	15.42
5	B	3	d	G	2	3	FALSE	1	14.74
6	B	3	d	G	2	2	TRUE	2	18.25
7	B	3	d	G	2	2	TRUE	2	19.63
8	A	3	d	P	2	2	TRUE	2	23.25
9	A	3	d	P	2	2	TRUE	2	39.76
10	A	3	d	P	2	1	FALSE	4	42.13
11	C	3	d	F	2	2	TRUE	2	18.37
12	B	3	d	G	2	3	FALSE	2	41.8

Subj.	Grp	Map	Qu.	Map Type	Answer	Given Answer	Correct	Ease	Time (s)
13	C	3	d	F	2	2	TRUE	2	29.72
14	A	3	d	P	2	3	FALSE	2	28.74
15	A	3	d	P	2	2	TRUE	2	12.92
16	C	3	d	F	2	2	TRUE	0	23.95
17	B	3	d	G	2	2	TRUE	4	66.03
18	B	3	d	G	2	4	FALSE	2	16
19	C	3	d	F	2	2	TRUE	1	46.8
20	C	3	d	F	2	3	FALSE	2	36.52
21	A	3	d	P	2	2	TRUE	2	33.89
22	B	3	d	G	2	2	TRUE	1	14.47
23	A	3	d	P	2	2	TRUE	2	28.4
24	B	3	d	G	2	3	FALSE	1	18.91
25	C	3	d	F	2	2	TRUE	0	24.3
26	B	3	d	G	2	2	TRUE	1	26.87
27	A	3	d	P	2	2	TRUE	0	40.95
28	B	3	d	G	2	1	FALSE	2	27.44
29	A	3	d	P	2	2	TRUE	3	34.38
30	C	3	d	F	2	2	TRUE	1	26.81
31	A	3	d	P	2	2	TRUE	2	25.67
32	C	3	d	F	2	2	TRUE	3	19.59
33	B	3	d	G	2	2	TRUE	3	26.67
34	B	3	d	G	2	2	TRUE	2	67.62
35	C	3	d	F	2	2	TRUE	1	21.69
36	C	3	d	F	2	2	TRUE	2	12.52
37	B	3	d	G	2	2	TRUE	3	21.6
38	A	3	d	P	2	3	FALSE	2	19.08
39	A	3	d	P	2	2	TRUE	2	21.61
40	B	3	d	G	2	2	TRUE	1	24.93
41	B	3	d	G	2	2	TRUE	3	19.94
42	C	3	d	F	2	2	TRUE	2	18.92
43	C	3	d	F	2	2	TRUE	2	20.33
1	C	4	a	F	10	10	TRUE	2	31.13
2	C	4	a	F	10	7	FALSE	2	101.92
3	C	4	a	F	10	10	TRUE	4	21.1
4	A	4	a	P	10	10	TRUE	0	15.39
5	B	4	a	G	10	7	FALSE	2	16.92
6	B	4	a	G	10	7	FALSE	3	14.89
7	B	4	a	G	10	6	FALSE	2	23.42
8	A	4	a	P	10	9	FALSE	3	29.19
9	A	4	a	P	10	7	FALSE	3	56.17
10	A	4	a	P	10	9	FALSE	3	31.8
11	C	4	a	F	10	7	FALSE	2	30.9
12	B	4	a	G	10	7	FALSE	2	26.72
13	C	4	a	F	10	7	FALSE	2	20.36
14	A	4	a	P	10	9	FALSE	2	25.69
15	A	4	a	P	10	7	FALSE	2	24.44
16	C	4	a	F	10	10	TRUE	1	17.83
17	B	4	a	G	10	7	FALSE	3	28.36

continued from previous page

Subj.	Grp	Map	Qu.	Map Type	Answer	Given Answer	Correct	Ease	Time (s)
18	B	4	a	G	10	7	FALSE	2	18.36
19	C	4	a	F	10	7	FALSE	1	41.08
20	C	4	a	F	10	7	FALSE	2	19.27
21	A	4	a	P	10	9	FALSE	2	16.89
22	B	4	a	G	10	7	FALSE	2	23.22
23	A	4	a	P	10	8	FALSE	3	54.4
24	B	4	a	G	10	7	FALSE	1	25.6
25	C	4	a	F	10	6	FALSE	1	13.77
26	B	4	a	G	10	7	FALSE	2	31.36
27	A	4	a	P	10	10	TRUE	2	34.11
28	B	4	a	G	10	7	FALSE	2	26.03
29	A	4	a	P	10	10	TRUE	3	15.13
30	C	4	a	F	10	7	FALSE	3	21.25
31	A	4	a	P	10	10	TRUE	4	25.1
32	C	4	a	F	10	7	FALSE	3	22.91
33	B	4	a	G	10	7	FALSE	2	18.14
34	B	4	a	G	10	7	FALSE	1	34.72
35	C	4	a	F	10	7	FALSE	2	21.42
36	C	4	a	F	10	7	FALSE	3	35.72
37	B	4	a	G	10	8	FALSE	3	13.94
38	A	4	a	P	10	9	FALSE	3	18.43
39	A	4	a	P	10	7	FALSE	2	23.38
40	B	4	a	G	10	7	FALSE	0	15.67
41	B	4	a	G	10	7	FALSE	2	10.7
42	C	4	a	F	10	7	FALSE	3	18.72
43	C	4	a	F	10	7	FALSE	2	27.61
1	C	4	b	G	6	5	FALSE	2	17.58
2	C	4	b	G	6	5	FALSE	2	16.33
3	C	4	b	G	6	4	FALSE	2	10.89
4	A	4	b	F	6	6	TRUE	0	7.44
5	B	4	b	P	6	6	TRUE	3	20.31
6	B	4	b	P	6	6	TRUE	1	11
7	B	4	b	P	6	6	TRUE	2	33.21
8	A	4	b	F	6	6	TRUE	2	19.81
9	A	4	b	F	6	6	TRUE	2	12.33
10	A	4	b	F	6	6	TRUE	2	15.95
11	C	4	b	G	6	5	FALSE	3	25.85
12	B	4	b	P	6	6	TRUE	4	82.65
13	C	4	b	G	6	4	FALSE	2	10.64
14	A	4	b	F	6	6	TRUE	2	10.31
15	A	4	b	F	6	6	TRUE	1	9.38
16	C	4	b	G	6	4	FALSE	0	16.53
17	B	4	b	P	6	6	TRUE	3	23.33
18	B	4	b	P	6	5	FALSE	1	10.37
19	C	4	b	G	6	5	FALSE	1	19.47
20	C	4	b	G	6	5	FALSE	1	12.41
21	A	4	b	F	6	6	TRUE	1	12.22
22	B	4	b	P	6	6	TRUE	3	22.5

continued from previous page

Subj.	Grp	Map	Qu.	Map Type	Answer	Given Answer	Correct	Ease	Time (s)
23	A	4	b	F	6	6	TRUE	2	33.55
24	B	4	b	P	6	2	FALSE	2	26.33
25	C	4	b	G	6	5	FALSE	1	8.84
26	B	4	b	P	6	6	TRUE	2	32.08
27	A	4	b	F	6	6	TRUE	1	16.23
28	B	4	b	P	6	6	TRUE	2	9.37
29	A	4	b	F	6	6	TRUE	3	7.48
30	C	4	b	G	6	5	FALSE	3	11.22
31	A	4	b	F	6	6	TRUE	3	10.64
32	C	4	b	G	6	5	FALSE	2	11.61
33	B	4	b	P	6	6	TRUE	2	14.08
34	B	4	b	P	6	6	TRUE	4	65.72
35	C	4	b	G	6	5	FALSE	1	15.91
36	C	4	b	G	6	5	FALSE	1	9.16
37	B	4	b	P	6	6	TRUE	2	25.03
38	A	4	b	F	6	6	TRUE	2	11.31
39	A	4	b	F	6	6	TRUE	2	8.44
40	B	4	b	P	6	6	TRUE	0	25.91
41	B	4	b	P	6	6	TRUE	3	28.76
42	C	4	b	G	6	5	FALSE	4	14.12
43	C	4	b	G	6	4	FALSE	1	8.64
1	C	4	c	F	0	0	TRUE	2	21.7
2	C	4	c	F	0	1	FALSE	2	28.2
3	C	4	c	F	0	0	TRUE	3	20.11
4	A	4	c	P	0	0	TRUE	0	10.31
5	B	4	c	G	0	0	TRUE	1	9.95
6	B	4	c	G	0	0	TRUE	2	10.47
7	B	4	c	G	0	0	TRUE	1	16.21
8	A	4	c	P	0	0	TRUE	1	9.44
9	A	4	c	P	0	0	TRUE	2	15.33
10	A	4	c	P	0	0	TRUE	0	8.31
11	C	4	c	F	0	1	FALSE	3	37.82
12	B	4	c	G	0	0	TRUE	2	26.1
13	C	4	c	F	0	0	TRUE	1	23.31
14	A	4	c	P	0	0	TRUE	0	6.97
15	A	4	c	P	0	0	TRUE	1	12.06
16	C	4	c	F	0	0	TRUE	1	39.44
17	B	4	c	G	0	1	FALSE	3	35.94
18	B	4	c	G	0	2	FALSE	2	22.39
19	C	4	c	F	0	0	TRUE	2	21.2
20	C	4	c	F	0	0	TRUE	2	22.28
21	A	4	c	P	0	0	TRUE	1	11.88
22	B	4	c	G	0	0	TRUE	1	13.19
23	A	4	c	P	0	0	TRUE	2	18.07
24	B	4	c	G	0	0	TRUE	2	16.86
25	C	4	c	F	0	1	FALSE	2	46.33
26	B	4	c	G	0	0	TRUE	1	17.44
27	A	4	c	P	0	0	TRUE	0	14.56

Subj.	Grp	Map	Qu.	Map Type	Answer	Given Answer	Correct	Ease	Time (s)
28	B	4	c	G	0	0	TRUE	2	16.94
29	A	4	c	P	0	0	TRUE	1	6.58
30	C	4	c	F	0	0	TRUE	2	17.03
31	A	4	c	P	0	0	TRUE	1	9.43
32	C	4	c	F	0	0	TRUE	1	6.08
33	B	4	c	G	0	0	TRUE	0	9.92
34	B	4	c	G	0	0	TRUE	1	31.99
35	C	4	c	F	0	0	TRUE	1	10.5
36	C	4	c	F	0	0	TRUE	2	26.84
37	B	4	c	G	0	0	TRUE	1	11.7
38	A	4	c	P	0	0	TRUE	2	10.83
39	A	4	c	P	0	0	TRUE	2	10.66
40	B	4	c	G	0	0	TRUE	0	11.38
41	B	4	c	G	0	0	TRUE	2	16.2
42	C	4	c	F	0	0	TRUE	1	9.86
43	C	4	c	F	0	0	TRUE	0	6.9
1	C	4	d	G	1	3	FALSE	3	24.8
2	C	4	d	G	1	1	TRUE	2	28.32
3	C	4	d	G	1	1	TRUE	3	21.28
4	A	4	d	F	1	3	FALSE	2	13.92
5	B	4	d	P	1	1	TRUE	2	37.47
6	B	4	d	P	1	1	TRUE	2	16.13
7	B	4	d	P	1	1	TRUE	2	57.62
8	A	4	d	F	1	1	TRUE	2	12.58
9	A	4	d	F	1	1	TRUE	2	35.23
10	A	4	d	F	1	1	TRUE	2	16.11
11	C	4	d	G	1	2	FALSE	3	38.27
12	B	4	d	P	1	2	FALSE	3	95.47
13	C	4	d	G	1	1	TRUE	2	17.25
14	A	4	d	F	1	1	TRUE	2	35.21
15	A	4	d	F	1	1	TRUE	1	13.83
16	C	4	d	G	1	1	TRUE	0	20.05
17	B	4	d	P	1	1	TRUE	4	39.94
18	B	4	d	P	1	2	FALSE	1	12.13
19	C	4	d	G	1	1	TRUE	1	17.56
20	C	4	d	G	1	1	TRUE	2	14.45
21	A	4	d	F	1	1	TRUE	2	15.66
22	B	4	d	P	1	1	TRUE	3	39.63
23	A	4	d	F	1	1	TRUE	3	59.89
24	B	4	d	P	1	1	TRUE	0	21.07
25	C	4	d	G	1	1	TRUE	0	8.67
26	B	4	d	P	1	1	TRUE	1	48.66
27	A	4	d	F	1	1	TRUE	0	12.21
28	B	4	d	P	1	1	TRUE	2	46.44
29	A	4	d	F	1	1	TRUE	3	31.8
30	C	4	d	G	1	4	FALSE	4	10.38
31	A	4	d	F	1	1	TRUE	2	17.98
32	C	4	d	G	1	1	TRUE	2	12.13

Subj.	Grp	Map	Qu.	Map Type	Answer	Given Answer	Correct	Ease	Time (s)
33	B	4	d	P	1	1	TRUE	2	17.67
34	B	4	d	P	1	0	FALSE	3	52.16
35	C	4	d	G	1	1	TRUE	1	6.59
36	C	4	d	G	1	1	TRUE	2	11.88
37	B	4	d	P	1	2	FALSE	4	37.66
38	A	4	d	F	1	1	TRUE	2	12.43
39	A	4	d	F	1	4	FALSE	4	28.44
40	B	4	d	P	1	2	FALSE	2	20.48
41	B	4	d	P	1	2	FALSE	2	17.31
42	C	4	d	G	1	1	TRUE	2	10.77
43	C	4	d	G	1	1	TRUE	1	16.15
1	C	5	a	G	7	6	FALSE	2	17.95
2	C	5	a	G	7	6	FALSE	2	20.47
3	C	5	a	G	7	7	TRUE	2	21.62
4	A	5	a	F	7	6	FALSE	1	14.11
5	B	5	a	P	7	6	FALSE	3	27.14
6	B	5	a	P	7	6	FALSE	4	39.75
7	B	5	a	P	7	6	FALSE	2	55.24
8	A	5	a	F	7	6	FALSE	3	23.45
9	A	5	a	F	7	6	FALSE	2	24.72
10	A	5	a	F	7	6	FALSE	2	26.5
11	C	5	a	G	7	6	FALSE	4	45.51
12	B	5	a	P	7	6	FALSE	2	70.93
13	C	5	a	G	7	6	FALSE	3	25.48
14	A	5	a	F	7	6	FALSE	1	11.89
15	A	5	a	F	7	6	FALSE	1	11.94
16	C	5	a	G	7	6	FALSE	4	54.92
17	B	5	a	P	7	6	FALSE	3	27.56
18	B	5	a	P	7	6	FALSE	3	96.4
19	C	5	a	G	7	6	FALSE	2	49.14
20	C	5	a	G	7	6	FALSE	2	16.63
21	A	5	a	F	7	6	FALSE	1	11.95
22	B	5	a	P	7	6	FALSE	2	27.42
23	A	5	a	F	7	7	TRUE	2	27.93
24	B	5	a	P	7	6	FALSE	2	36.5
25	C	5	a	G	7	6	FALSE	3	85.13
26	B	5	a	P	7	10	FALSE	3	103.14
27	A	5	a	F	7	6	FALSE	0	16.34
28	B	5	a	P	7	6	FALSE	2	25.97
29	A	5	a	F	7	6	FALSE	2	9.31
30	C	5	a	G	7	6	FALSE	3	22.34
31	A	5	a	F	7	6	FALSE	2	24.06
32	C	5	a	G	7	6	FALSE	2	9.05
33	B	5	a	P	7	6	FALSE	4	68
34	B	5	a	P	7	6	FALSE	2	33.59
35	C	5	a	G	7	6	FALSE	4	47.42
36	C	5	a	G	7	6	FALSE	4	71.85
37	B	5	a	P	7	6	FALSE	4	20.63

continued from previous page

Subj.	Grp	Map	Qu.	Map Type	Answer	Given Answer	Correct	Ease	Time (s)
38	A	5	a	F	7	6	FALSE	3	15.17
39	A	5	a	F	7	6	FALSE	2	11.42
40	B	5	a	P	7	6	FALSE	0	63.36
41	B	5	a	P	7	6	FALSE	3	32.08
42	C	5	a	G	7	6	FALSE	4	32.1
43	C	5	a	G	7	6	FALSE	3	55.08
1	C	5	b	P	14	12	FALSE	2	29.75
2	C	5	b	P	14	11	FALSE	2	21.39
3	C	5	b	P	14	11	FALSE	4	21.72
4	A	5	b	G	14	13	FALSE	2	9.77
5	B	5	b	F	14	14	TRUE	2	15.99
6	B	5	b	F	14	14	TRUE	3	18.2
7	B	5	b	F	14	14	TRUE	2	13.67
8	A	5	b	G	14	13	FALSE	3	32.63
9	A	5	b	G	14	13	FALSE	2	26.23
10	A	5	b	G	14	13	FALSE	3	33.52
11	C	5	b	P	14	14	TRUE	4	30.74
12	B	5	b	F	14	13	FALSE	3	53.22
13	C	5	b	P	14	13	FALSE	3	19.12
14	A	5	b	G	14	13	FALSE	2	13.25
15	A	5	b	G	14	13	FALSE	2	12.56
16	C	5	b	P	14	13	FALSE	2	19.5
17	B	5	b	F	14	14	TRUE	3	35.75
18	B	5	b	F	14	13	FALSE	1	14.44
19	C	5	b	P	14	12	FALSE	3	27.06
20	C	5	b	P	14	13	FALSE	3	20.92
21	A	5	b	G	14	10	FALSE	2	16.39
22	B	5	b	F	14	14	TRUE	3	15.14
23	A	5	b	G	14	13	FALSE	2	35.42
24	B	5	b	F	14	14	TRUE	2	19.66
25	C	5	b	P	14	13	FALSE	1	14.91
26	B	5	b	F	14	14	TRUE	2	30.08
27	A	5	b	G	14	13	FALSE	1	12.37
28	B	5	b	F	14	14	TRUE	2	14.61
29	A	5	b	G	14	13	FALSE	3	13.93
30	C	5	b	P	14	13	FALSE	3	20.64
31	A	5	b	G	14	13	FALSE	2	13.9
32	C	5	b	P	14	14	TRUE	2	15.57
33	B	5	b	F	14	14	TRUE	2	14
34	B	5	b	F	14	14	TRUE	0	29.21
35	C	5	b	P	14	13	FALSE	1	14.95
36	C	5	b	P	14	13	FALSE	2	17.05
37	B	5	b	F	14	14	TRUE	3	12.07
38	A	5	b	G	14	13	FALSE	3	25.29
39	A	5	b	G	14	13	FALSE	2	12.28
40	B	5	b	F	14	14	TRUE	0	12.16
41	B	5	b	F	14	14	TRUE	3	15.15
42	C	5	b	P	14	13	FALSE	4	16.78

Subj.	Grp	Map	Qu.	Map Type	Answer	Given Answer	Correct	Ease	Time (s)
43	C	5	b	P	14	13	FALSE	1	13.95
1	C	5	c	G	1	1	TRUE	1	13.42
2	C	5	c	G	1	1	TRUE	2	26.44
3	C	5	c	G	1	1	TRUE	2	13.68
4	A	5	c	F	1	1	TRUE	1	9.36
5	B	5	c	P	1	1	TRUE	1	7.83
6	B	5	c	P	1	1	TRUE	2	10.63
7	B	5	c	P	1	1	TRUE	2	13.85
8	A	5	c	F	1	1	TRUE	2	13.16
9	A	5	c	F	1	1	TRUE	2	13.41
10	A	5	c	F	1	1	TRUE	1	15.11
11	C	5	c	G	1	1	TRUE	2	26.81
12	B	5	c	P	1	2	FALSE	1	20.77
13	C	5	c	G	1	1	TRUE	2	16.52
14	A	5	c	F	1	1	TRUE	0	8.38
15	A	5	c	F	1	1	TRUE	2	13.78
16	C	5	c	G	1	1	TRUE	1	15
17	B	5	c	P	1	1	TRUE	0	28.99
18	B	5	c	P	1	2	FALSE	0	23.31
19	C	5	c	G	1	1	TRUE	1	21.45
20	C	5	c	G	1	1	TRUE	1	13.27
21	A	5	c	F	1	1	TRUE	1	8.52
22	B	5	c	P	1	1	TRUE	1	10.28
23	A	5	c	F	1	1	TRUE	2	11.01
24	B	5	c	P	1	1	TRUE	1	12.92
25	C	5	c	G	1	1	TRUE	1	9.8
26	B	5	c	P	1	1	TRUE	1	10.13
27	A	5	c	F	1	1	TRUE	0	11.38
28	B	5	c	P	1	1	TRUE	2	22.77
29	A	5	c	F	1	1	TRUE	2	4.44
30	C	5	c	G	1	1	TRUE	2	11.12
31	A	5	c	F	1	1	TRUE	2	8.67
32	C	5	c	G	1	1	TRUE	1	9.57
33	B	5	c	P	1	1	TRUE	1	13.59
34	B	5	c	P	1	1	TRUE	0	21.42
35	C	5	c	G	1	1	TRUE	2	7.77
36	C	5	c	G	1	1	TRUE	2	11.27
37	B	5	c	P	1	1	TRUE	2	12
38	A	5	c	F	1	1	TRUE	2	10.83
39	A	5	c	F	1	1	TRUE	2	9.11
40	B	5	c	P	1	1	TRUE	0	9.28
41	B	5	c	P	1	1	TRUE	1	8.19
42	C	5	c	G	1	1	TRUE	1	20.35
43	C	5	c	G	1	1	TRUE	1	10.24
1	C	5	d	P	2	3	FALSE	2	10.64
2	C	5	d	P	2	1	FALSE	2	11.87
3	C	5	d	P	2	2	TRUE	3	12.69
4	A	5	d	G	2	1	FALSE	2	7.05

Subj.	Grp	Map	Qu.	Map Type	Answer	Given Answer	Correct	Ease	Time (s)
5	B	5	d	F	2	2	TRUE	2	11.2
6	B	5	d	F	2	1	FALSE	2	21.13
7	B	5	d	F	2	1	FALSE	1	15.97
8	A	5	d	G	2	1	FALSE	2	19.22
9	A	5	d	G	2	2	TRUE	2	20.33
10	A	5	d	G	2	1	FALSE	4	40.17
11	C	5	d	P	2	2	TRUE	3	21.32
12	B	5	d	F	2	1	FALSE	1	19.61
13	C	5	d	P	2	2	TRUE	1	15.5
14	A	5	d	G	2	2	TRUE	2	34.28
15	A	5	d	G	2	2	TRUE	2	15.22
16	C	5	d	P	2	1	FALSE	4	28.64
17	B	5	d	F	2	2	TRUE	4	20.05
18	B	5	d	F	2	1	FALSE	0	15.96
19	C	5	d	P	2	2	TRUE	2	38.86
20	C	5	d	P	2	1	FALSE	1	13.33
21	A	5	d	G	2	1	FALSE	1	19.06
22	B	5	d	F	2	1	FALSE	1	16.5
23	A	5	d	G	2	1	FALSE	3	26.69
24	B	5	d	F	2	2	TRUE	2	29.5
25	C	5	d	P	2	2	TRUE	4	19.44
26	B	5	d	F	2	1	FALSE	1	14.67
27	A	5	d	G	2	2	TRUE	3	47.1
28	B	5	d	F	2	1	FALSE	2	8.82
29	A	5	d	G	2	1	FALSE	3	25.26
30	C	5	d	P	2	1	FALSE	1	17.34
31	A	5	d	G	2	2	TRUE	3	23.08
32	C	5	d	P	2	2	TRUE	2	10.37
33	B	5	d	F	2	2	TRUE	3	19.43
34	B	5	d	F	2	1	FALSE	0	17.05
35	C	5	d	P	2	2	TRUE	2	15.59
36	C	5	d	P	2	2	TRUE	3	25.88
37	B	5	d	F	2	2	TRUE	3	10.42
38	A	5	d	G	2	1	FALSE	1	9.04
39	A	5	d	G	2	1	FALSE	3	22.2
40	B	5	d	F	2	2	TRUE	2	25.58
41	B	5	d	F	2	1	FALSE	2	10.34
42	C	5	d	P	2	1	FALSE	1	10.8
43	C	5	d	P	2	2	TRUE	2	23.67
1	C	6	a	P	2	1	FALSE	1	12.34
2	C	6	a	P	2	2	TRUE	2	55.77
3	C	6	a	P	2	3	FALSE	0	28.71
4	A	6	a	G	2	1	FALSE	2	10.88
5	B	6	a	F	2	2	TRUE	1	15.61
6	B	6	a	F	2	1	FALSE	1	16.03
7	B	6	a	F	2	2	TRUE	0	20.75
8	A	6	a	G	2	2	TRUE	3	14.97
9	A	6	a	G	2	1	FALSE	1	22.05

Subj.	Grp	Map	Qu.	Map Type	Answer	Given Answer	Correct	Ease	Time (s)
10	A	6	a	G	2	1	FALSE	1	12.27
11	C	6	a	P	2	5	FALSE	4	61.13
12	B	6	a	F	2	2	TRUE	1	24.77
13	C	6	a	P	2	2	TRUE	3	12.23
14	A	6	a	G	2	2	TRUE	1	8.97
15	A	6	a	G	2	2	TRUE	2	13.92
16	C	6	a	P	2	5	FALSE	4	72.85
17	B	6	a	F	2	2	TRUE	0	16.13
18	B	6	a	F	2	2	TRUE	0	21.71
19	C	6	a	P	2	2	TRUE	2	48.1
20	C	6	a	P	2	1	FALSE	1	11.41
21	A	6	a	G	2	1	FALSE	1	10.8
22	B	6	a	F	2	2	TRUE	1	7.13
23	A	6	a	G	2	2	TRUE	1	16.64
24	B	6	a	F	2	2	TRUE	1	33.33
25	C	6	a	P	2	5	FALSE	4	102.06
26	B	6	a	F	2	2	TRUE	1	16.97
27	A	6	a	G	2	1	FALSE	0	15.16
28	B	6	a	F	2	1	FALSE	2	18.72
29	A	6	a	G	2	5	FALSE	2	12.94
30	C	6	a	P	2	5	FALSE	4	42.42
31	A	6	a	G	2	1	FALSE	1	20.56
32	C	6	a	P	2	5	FALSE	4	76.19
33	B	6	a	F	2	2	TRUE	1	9.11
34	B	6	a	F	2	2	TRUE	1	14.52
35	C	6	a	P	2	2	TRUE	2	28.03
36	C	6	a	P	2	3	FALSE	4	82.63
37	B	6	a	F	2	2	TRUE	1	27.3
38	A	6	a	G	2	1	FALSE	4	9.94
39	A	6	a	G	2	1	FALSE	2	8.2
40	B	6	a	F	2	2	TRUE	0	10.18
41	B	6	a	F	2	2	TRUE	1	6.31
42	C	6	a	P	2	2	TRUE	3	28.64
43	C	6	a	P	2	1	FALSE	0	10.9
1	C	6	b	F	4	4	TRUE	2	11.92
2	C	6	b	F	4	4	TRUE	2	14.94
3	C	6	b	F	4	4	TRUE	3	20.48
4	A	6	b	P	4	4	TRUE	2	9.75
5	B	6	b	G	4	4	TRUE	3	20.05
6	B	6	b	G	4	4	TRUE	3	15.33
7	B	6	b	G	4	4	TRUE	3	26.39
8	A	6	b	P	4	4	TRUE	2	19.49
9	A	6	b	P	4	4	TRUE	2	27.39
10	A	6	b	P	4	5	FALSE	4	28.44
11	C	6	b	F	4	4	TRUE	1	23.36
12	B	6	b	G	4	4	TRUE	3	54.49
13	C	6	b	F	4	4	TRUE	1	13.94
14	A	6	b	P	4	4	TRUE	3	12.69

continued from previous page

Subj.	Grp	Map	Qu.	Map Type	Answer	Given Answer	Correct	Ease	Time (s)
15	A	6	b	P	4	4	TRUE	2	14.02
16	C	6	b	F	4	4	TRUE	0	9.42
17	B	6	b	G	4	4	TRUE	3	25.52
18	B	6	b	G	4	4	TRUE	2	31.15
19	C	6	b	F	4	4	TRUE	1	24.39
20	C	6	b	F	4	4	TRUE	2	12.97
21	A	6	b	P	4	4	TRUE	3	22.53
22	B	6	b	G	4	4	TRUE	3	8.48
23	A	6	b	P	4	4	TRUE	2	28.18
24	B	6	b	G	4	4	TRUE	1	21.69
25	C	6	b	F	4	4	TRUE	1	6.78
26	B	6	b	G	4	5	FALSE	4	21.3
27	A	6	b	P	4	4	TRUE	2	27.21
28	B	6	b	G	4	6	FALSE	2	20.12
29	A	6	b	P	4	4	TRUE	3	14.01
30	C	6	b	F	4	4	TRUE	2	10.34
31	A	6	b	P	4	4	TRUE	2	11.88
32	C	6	b	F	4	4	TRUE	1	10.49
33	B	6	b	G	4	4	TRUE	4	34.82
34	B	6	b	G	4	5	FALSE	1	34.39
35	C	6	b	F	4	4	TRUE	1	11.7
36	C	6	b	F	4	4	TRUE	1	11.88
37	B	6	b	G	4	4	TRUE	2	11.51
38	A	6	b	P	4	4	TRUE	3	12.9
39	A	6	b	P	4	5	FALSE	4	16.47
40	B	6	b	G	4	4	TRUE	0	21.12
41	B	6	b	G	4	4	TRUE	2	12.21
42	C	6	b	F	4	4	TRUE	3	18.05
43	C	6	b	F	4	4	TRUE	0	11.71
1	C	6	c	P	0	0	TRUE	2	20.48
2	C	6	c	P	0	0	TRUE	2	32.13
3	C	6	c	P	0	0	TRUE	0	17.25
4	A	6	c	G	0	0	TRUE	1	11.23
5	B	6	c	F	0	0	TRUE	0	7.02
6	B	6	c	F	0	2	FALSE	3	14.13
7	B	6	c	F	0	0	TRUE	1	13.6
8	A	6	c	G	0	0	TRUE	2	22.41
9	A	6	c	G	0	0	TRUE	3	36.36
10	A	6	c	G	0	0	TRUE	1	12.97
11	C	6	c	P	0	0	TRUE	2	29.91
12	B	6	c	F	0	0	TRUE	2	36
13	C	6	c	P	0	1	FALSE	2	22.66
14	A	6	c	G	0	0	TRUE	1	12.09
15	A	6	c	G	0	0	TRUE	1	14.58
16	C	6	c	P	0	0	TRUE	0	15.24
17	B	6	c	F	0	0	TRUE	1	31.28
18	B	6	c	F	0	2	FALSE	1	32.63
19	C	6	c	P	0	0	TRUE	1	21.48

continued from previous page

Subj.	Grp	Map	Qu.	Map Type	Answer	Given Answer	Correct	Ease	Time (s)
20	C	6	c	P	0	0	TRUE	2	19.55
21	A	6	c	G	0	0	TRUE	1	12.33
22	B	6	c	F	0	0	TRUE	1	12.73
23	A	6	c	G	0	2	FALSE	2	16.92
24	B	6	c	F	0	4	FALSE	2	28.91
25	C	6	c	P	0	0	TRUE	1	8.05
26	B	6	c	F	0	0	TRUE	1	16.66
27	A	6	c	G	0	0	TRUE	0	11.59
28	B	6	c	F	0	0	TRUE	2	9.48
29	A	6	c	G	0	0	TRUE	1	18.36
30	C	6	c	P	0	0	TRUE	2	11.89
31	A	6	c	G	0	0	TRUE	1	13.35
32	C	6	c	P	0	0	TRUE	2	12.97
33	B	6	c	F	0	0	TRUE	1	10.27
34	B	6	c	F	0	0	TRUE	0	16.63
35	C	6	c	P	0	0	TRUE	1	13.41
36	C	6	c	P	0	0	TRUE	1	14.64
37	B	6	c	F	0	0	TRUE	1	13.73
38	A	6	c	G	0	0	TRUE	4	14.32
39	A	6	c	G	0	0	TRUE	2	11.78
40	B	6	c	F	0	0	TRUE	4	7.77
41	B	6	c	F	0	0	TRUE	1	12.75
42	C	6	c	P	0	0	TRUE	1	13.58
43	C	6	c	P	0	0	TRUE	1	22.1
1	C	6	d	F	1	2	FALSE	2	25.09
2	C	6	d	F	1	2	FALSE	2	28.09
3	C	6	d	F	1	1	TRUE	2	24.04
4	A	6	d	P	1	1	TRUE	0	15.72
5	B	6	d	G	1	1	TRUE	2	11.94
6	B	6	d	G	1	2	FALSE	2	23.72
7	B	6	d	G	1	1	TRUE	2	14.03
8	A	6	d	P	1	1	TRUE	4	27.49
9	A	6	d	P	1	1	TRUE	4	79.26
10	A	6	d	P	1	0	FALSE	1	17.42
11	C	6	d	F	1	1	TRUE	2	31.54
12	B	6	d	G	1	1	TRUE	2	46.27
13	C	6	d	F	1	1	TRUE	3	26.45
14	A	6	d	P	1	1	TRUE	2	31.13
15	A	6	d	P	1	1	TRUE	2	23.36
16	C	6	d	F	1	2	FALSE	2	22.22
17	B	6	d	G	1	1	TRUE	4	41.63
18	B	6	d	G	1	2	FALSE	2	20.52
19	C	6	d	F	1	1	TRUE	2	32.35
20	C	6	d	F	1	3	FALSE	2	20.44
21	A	6	d	P	1	1	TRUE	2	34.13
22	B	6	d	G	1	1	TRUE	1	17.73
23	A	6	d	P	1	1	TRUE	2	31.69
24	B	6	d	G	1	1	TRUE	1	20.52

continued from previous page

Subj.	Grp	Map	Qu.	Map Type	Answer	Given Answer	Correct	Ease	Time (s)
25	C	6	d	F	1	1	TRUE	1	13.14
26	B	6	d	G	1	1	TRUE	1	14.28
27	A	6	d	P	1	1	TRUE	2	20.96
28	B	6	d	G	1	1	TRUE	2	22.85
29	A	6	d	P	1	1	TRUE	4	57.18
30	C	6	d	F	1	1	TRUE	2	15.94
31	A	6	d	P	1	1	TRUE	3	48.23
32	C	6	d	F	1	1	TRUE	3	13.07
33	B	6	d	G	1	1	TRUE	2	14.51
34	B	6	d	G	1	1	TRUE	2	35.46
35	C	6	d	F	1	1	TRUE	1	23.59
36	C	6	d	F	1	1	TRUE	2	15.08
37	B	6	d	G	1	1	TRUE	3	11.23
38	A	6	d	P	1	2	FALSE	4	18.66
39	A	6	d	P	1	1	TRUE	4	24.05
40	B	6	d	G	1	1	TRUE	1	19.17
41	B	6	d	G	1	1	TRUE	2	11.45
42	C	6	d	F	1	2	FALSE	2	21.08
43	C	6	d	F	1	1	TRUE	2	22.69

A.3.2 Design Preferences

Table A.9: 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

continued from previous page

Candidate	Atlanta	Bucharest	Mexico	Stockholm	Toronto	Washington
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
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 A.10: 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

continued from previous page

Candidate	Atlanta	Bucharest	Mexico	Stockholm	Toronto	Washington
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
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 A.11: 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

Candidate	Atlanta	Bucharest	Mexico	Stockholm	Toronto	Washington
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
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

A.3.3 Questionnaire Feedback

Table A.12: 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

Candidate	Answer
3	Yes Slide 6
4	No
5	No
6	No
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

Candidate	Answer
43	Washington DC

Table A.13: 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

continued from previous page

Candidate	Answer
24	The blue or red coloured lines (made it easier to see the stations)
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 A.14: 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

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.
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

Candidate	Answer
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 A.15: 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.
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

Candidate	Answer
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
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

Appendix B

Map Input Data

This Appendix contains the input data for all of the maps presented in Chapter 6. The data for each map consists of a set of node definitions and a set of edge definitions. Each node definition contains the label for that node together with the starting x and y co-ordinates for that node. The starting co-ordinates are usually the latitude or longitude in degress of that station, but sometimes other reference systems are used. The edge definitions consist of a label (the name of the line that that edge is part of), a colour (red, green, blue), followed by a list of nodes. The list of nodes can either be just two nodes for a single edge or it can be a longer list. Longer lists are just shorthand for writing lots of individual edge definitions.

The format is also explained in Section 5.2.

B.1   Atlanta

```
NODE: label=Five Points : x=−84.3916:y=33.7539
NODE: label=Bankhead : x=−84.4288:y=33.4288
NODE: label=Indian Creek : x=−84.2295:y=33.7697
NODE: label=North Avenue : x=−84.3863:y=33.7719
NODE: label=Buckhead : x=−84.3673:y=33.8479
NODE: label=Medical Center : x=−84.3527:y=33.9108
NODE: label=Dunwoody : x=−84.3444:y=33.9213
NODE: label=Sandy Springs : x=−84.3531:y=33.9326
NODE: label=North Springs : x=−84.3574:y=33.9439
NODE: label=Brookhaven/Oglethorpe University : x=−84.3389:y
    =33.8605
```

NODE: label=Chamblee : x=−84.3075:y=33.8869
NODE: label=Doraville : x=−84.2805:y=33.9028
NODE: label=West End : x=−84.4135:y=33.7361
NODE: label=Oakland City : x=−84.4256:y=33.7153
NODE: label=Lakewood/Ft. McPherson : x=−84.4287:y=33.7003
NODE: label=East Point : x=−84.4409:y=33.6773
NODE: label=College Park : x=−84.4487:y=33.6516
NODE: label=Airport : x=−84.4462:y=33.6395
NODE: label=Georgia State : x=−84.3863:y=33.7507
NODE: label=King Memorial : x=−84.3746:y=33.7501
NODE: label=Inman Park/Reynoldstown : x=−84.3529:y=33.7574
NODE: label=Edgwood/Candler Park : x=−84.3390:y=33.7621
NODE: label=East Lake : x=−84.3130:y=33.7653
NODE: label=Decatur : x=−84.2954:y=33.7752
NODE: label=Avondale : x=−84.2828:y=33.7749
NODE: label=Kensington : x=−84.2502:y=33.7723
NODE: label=Dome/GWCC/Philips Arena/CNN Center : x=−84.3988:y
    =33.7565
NODE: label=Vine City : x=−84.4061:y=33.7565
NODE: label=Ashby : x=−84.4183:y=33.7566
NODE: label=West Lake : x=−84.4449:y=33.7537
NODE: label=Hamilton E. Holmes : x=−84.4698:y=33.755
NODE: label=Peachtree Center : x=−84.3876:y=33.7598
NODE: label=Civic Center : x=−84.3873:y=33.7674
NODE: label=Midtown : x=−84.3864:y=33.7809
NODE: label=Arts Center : x=−84.3867:y=33.7893
NODE: label=Garnett : x=−84.3963:y=33.7481
NODE: label=Lenox : x=−84.3583:y=33.8454
NODE: label=Lindburgh Center : x=−84.3697:y=33.8212
EDGE: label=:color=255,171,25: adjlist=Five Points, Peachtree
    Center
EDGE: label=:color=255,171,25: adjlist=Peachtree Center, Civic
    Center
EDGE: label=:color=255,171,25: adjlist=Civic Center, North Avenue
EDGE: label=:color=255,171,25: adjlist=North Avenue, Midtown
EDGE: label=:color=255,171,25: adjlist=Midtown, Arts Center
EDGE: label=:color=255,171,25: adjlist=Arts Center, Lindburgh
    Center
EDGE: label=:color=255,171,25: adjlist=Lindburgh Center, Buckhead
EDGE: label=:color=255,171,25: adjlist=Buckhead, Medical Center
EDGE: label=:color=255,171,25: adjlist=Medical Center, Dunwoody
EDGE: label=:color=255,171,25: adjlist=Dunwoody, Sandy Springs
EDGE: label=:color=255,171,25: adjlist=Sandy Springs, North
    Springs
EDGE: label=:color=255,171,25: adjlist=Lindburgh Center, Lenox
EDGE: label=:color=255,171,25: adjlist=Lenox, Brookhaven/
    Oglethorpe University

EDGE: label=:color=255,171,25: adjlist=Brookhaven/Oglethorpe
    University, Chamblee
EDGE: label=:color=255,171,25: adjlist=Chamblee, Doraville
EDGE: label=:color=255,171,25: adjlist=Five Points, Garnett
EDGE: label=:color=255,171,25: adjlist=Garnett, West End
EDGE: label=:color=255,171,25: adjlist=West End, Oakland City
EDGE: label=:color=255,171,25: adjlist=Oakland City, Lakewood/Ft.
    McPherson
EDGE: label=:color=255,171,25: adjlist=Lakewood/Ft. McPherson,
    East Point
EDGE: label=:color=255,171,25: adjlist=East Point, College Park
EDGE: label=:color=255,171,25: adjlist=College Park, Airport
EDGE: label=:color=50,103,254: adjlist=Five Points, Dome/GWCC/
    Philips Arena/CNN Center
EDGE: label=:color=50,103,254: adjlist=Dome/GWCC/Philips Arena/
    CNN Center, Vine City
EDGE: label=:color=50,103,254: adjlist=Vine City, Ashby
EDGE: label=:color=50,103,254: adjlist=Ashby, West Lake
EDGE: label=:color=50,103,254: adjlist=West Lake, Hamilton E.
    Holmes
EDGE: label=:color=50,103,254: adjlist=Ashby, Bankhead
EDGE: label=:color=50,103,254: adjlist=Five Points, Georgia State
EDGE: label=:color=50,103,254: adjlist=Georgia State, King
    Memorial
EDGE: label=:color=50,103,254: adjlist=King Memorial, Inman Park/
    Reynoldstown
EDGE: label=:color=50,103,254: adjlist=Inman Park/Reynoldstown,
    Edgwood/Candler Park
EDGE: label=:color=50,103,254: adjlist=Edgwood/Candler Park, East
    Lake
EDGE: label=:color=50,103,254: adjlist=East Lake, Decatur
EDGE: label=:color=50,103,254: adjlist=Decatur, Avondale
EDGE: label=:color=50,103,254: adjlist=Avondale, Kensington
EDGE: label=:color=50,103,254: adjlist=Kensington, Indian Creek

B.2 Auckland

NODE: label=Boston Road:x=174.7688:y=−36.8661
NODE: label=Newmarket:x=174.7791:y=−36.8696
NODE: label=Remuera:x=174.7851:y=−36.8812
NODE: label=Panmure:x=174.8459:y=−36.9012
NODE: label=Glen Innes:x=174.8540:y=−36.8791
NODE: label=Greenlane:x=174.7987:y=−36.8905
NODE: label=Ellerslie:x=174.8075:y=−36.8978
NODE: label=Penrose:x=174.8152:y=−36.9096
NODE: label=Meadowbank:x=174.8208:y=−36.8663
NODE: label=Orakei:x=174.8099:y=36.8627
NODE: label=Britomart:x=174.7769:y=−36.8484

NODE: label=Mt Eden:x=174.7587:y=−36.8680
NODE: label=Kingsland:x=174.7452:y=−36.8722
NODE: label=Morningside:x=174.7353:y=−36.8751
NODE: label=Baldwin Avenue:x=174.7203:y=−36.8778
NODE: label=Mt Albert:x=174.7138:y=−36.8850
NODE: label=Avondale:x=174.7000:y=−36.8950
NODE: label=New Lynn:x=174.6840:y=−36.9094
NODE: label=Fruitvale Road:x=174.6698:y=−36.9115
NODE: label=Glen Eden:x=174.6532:y=−36.9103
NODE: label=Sunnyvale:x=174.6318:y=−36.8967
NODE: label=Henderson:x=174.6308:y=−36.8820
NODE: label=Sturges Road:x=174.6203:y=−36.8734
NODE: label=Ranui:x=174.6015:y=−36.8677
NODE: label=Swanson:x=174.5769:y=−36.8664
NODE: label=Waitakere:x=174.5436:y=−36.8495
NODE: label=Westfield:x=174.8319:y=−36.9381
NODE: label=Otahuhu:x=174.8334:y=−36.9475
NODE: label=Mangere:x=174.8348:y=−36.9558
NODE: label=Middlemore:x=174.8387:y=−36.9625
NODE: label=Papatoetoe:x=174.8502:y=−36.9781
NODE: label=Puhinui:x=174.8561:y=−36.9897
NODE: label=Wiri:x=174.8621:y=−37.0014
NODE: label=Homai:x=174.8748:y=−37.0134
NODE: label=Manurewa:x=174.8944:y=−37.0219
NODE: label=Te Mahia:x=174.9062:y=−37.0311
NODE: label=Takanini:x=174.9194:y=−37.0409
NODE: label=Papakura:x=174.9469:y=−37.0655
NODE: label=Pukekohe:x=174.9095:y=−37.2026
EDGE: label=Western Line:color=0,92,72:adjlist=Waitakere,
    Swanson, Ranui, Sturges Road, Henderson, Sunnyvale, Glen
    Eden, Fruitvale Road, New Lynn, Avondale, Mt Albert, Baldwin
     Avenue, Morningside, Kingsland, Mt Eden, Boston Road,
    Newmarket, Britomart
EDGE: label=Eastern Line:color=243,171,54:adjlist=Britomart,
    Orakei, Meadowbank, Glen Innes, Panmure, Westfield, Otahuhu,
     Mangere, Middlemore, Papatoetoe, Puhinui, Wiri, Homai,
    Manurewa, Te Mahia, Takanini, Papakura, Pukekohe
EDGE: label=Southern Line:color=177,12,62:adjlist=Britomart,
    Newmarket, Remuera, Greenlane, Ellerslie, Penrose, Westfield
    , Otahuhu, Mangere, Middlemore, Papatoetoe, Puhinui, Wiri,
    Homai, Manurewa, Te Mahia, Takanini, Papakura, Pukekohe

B.3   Bucharest

NODE: label=Piata Victoriei:x=606.2641208680187:y
    =433.4458635842211650.0
NODE: label=Dristor:x=699.4167363134591:y=327.2029155555039650.0

NODE:label=Piata Unirii:x=578.7945197005204:y
=308.89474473157026650.0
NODE:label=Romana:x=595.8992206749997:y=390.72712717213653650.0
NODE:label=Universtitate:x=587.3001487295211:y
=349.4681239948213650.0
NODE:label=Tineretului:x=567.9181141917643:y
=265.57292620518646 50.0
NODE:label=1 Dec. 1918:x=795.5529528727775:y
=297.6031767250176650.0
NODE:label=IMGB Depou:x=688.6739943368088:y
=23.548315150782628650.0
NODE:label=IMGB 1:x=625.7856274954758:y=58.491363797161284650.0
NODE:label=Aparatorii Patriei:x=573.0212935317273:y
=103.50194880468644650.0
NODE:label=Piata Sudului:x=537.7057906017257:y
=136.73568692331958650.0
NODE:label=Aviatorilor:x=607.8849036661383:y
=492.0932653311289650.0
NODE:label=Aurel Vlaicu:x=622.9127400559196:y
=538.5766722489166650.0
NODE:label=Pipera:x=695.4357268219205:y=543.4525360325331650.0
NODE:label=Timpuri Noi:x=626.7698426393716:y
=301.9859921747986650.0
NODE:label=Mihai Bravu:x=674.3500628238654:y
=302.55148518615425650.0
NODE:label=Piata Muncii:x=720.3201197569026:y
=376.09459199584273650.0
NODE:label=Iancului:x=703.9964048318012:y
=404.4897319649993650.0
NODE:label=Obor:x=664.8025624545041:y=406.2798714641121650.0
NODE:label=Stefan cel Mare:x=630.848708702024:y
=417.0431819804452650.0
NODE:label=Gara de Nord:x=569.0172269077977:y
=446.9626176284118650.0
NODE:label=Basarab:x=532.7371498641589:y=449.5201817867361650.0
NODE:label=N. Grigorescu:x=748.8211532118221:y
=317.77808986914954650.0
NODE:label=Titan:x=785.5155985493016:y=366.5569523810624650.0
NODE:label=Policolor:x=839.4682956310235:y
=276.3458991189872650.0
NODE:label=Linia de centura:x=878.9261783614058:y
=272.75290714201026650.0
NODE:label=Georgian Costin:x=827.7182933412583:y
=391.90139226871395650.0
NODE:label=Republica:x=809.8523393793713:y
=422.180049195584650.0
NODE:label=Izvor:x=539.123383562581:y=324.51407802453457650.0
NODE:label=Eroilor:x=512.2918849790794:y=350.787979879442650.0

NODE: label=Grozavesti:x=480.5348177109894:y
    =380.3371981621366650.0
NODE: label=Semanatoarea:x=449.90122363171787:y
    =396.66621304905823650.0
NODE: label=Crangasi:x=491.58463517253836:y
    =437.0507659028972650.0
NODE: label=Politehnica:x=468.93002929764674:y
    =340.4230528849302650.0
NODE: label=Armata Poporului:x=414.9624677540683:y
    =336.6804151048766650.0
NODE: label=Gorjului:x=370.2854804763591:y
    =328.7175726369838650.0
NODE: label=Pacii:x=319.7311229576297:y=333.88582465139547650.0
NODE: label=Industriilor:x=243.5814299687721:y
    =321.96607733381757650.0
NODE: label=Grivita:x=493.41042747549534:y
    =481.96593777024844650.0
NODE: label=1 Mai:x=453.5388069604622:y=522.0202599289454650.0
NODE: label=Pajura:x=439.8043258337593:y=555.9277958381861650.0
NODE: label=Laromet:x=386.22359689968124:y
    =588.7350791948212650.0
NODE: label=Eroii Revolutiei:x=559.1733019512307:y
    =221.8283558438893650.0
NODE: label=Constantin Brancoveanu:x=540.1549379143057:y
    =185.10983139081907650.0
EDGE: label=M1:color=255,0,0:adjlist=Stefan cel Mare, Piata
    Victoriei
EDGE: label=M1:color=255,0,0:adjlist=Stefan cel Mare, Obor
EDGE: label=M1:color=255,0,0:adjlist=Obor, Iancului
EDGE: label=M1:color=255,0,0:adjlist=Iancului, Piata Muncii
EDGE: label=M1:color=255,0,0:adjlist=Piata Muncii, Dristor
EDGE: label=M1:color=255,0,0:adjlist=Mihai Bravu, Dristor
EDGE: label=M1:color=255,0,0:adjlist=Mihai Bravu, Timpuri Noi
EDGE: label=M1:color=255,0,0:adjlist=Timpuri Noi, Piata Unirii
EDGE: label=M1:color=255,0,0:adjlist=Izvor, Piata Unirii
EDGE: label=M1:color=255,0,0:adjlist=Eroilor, Izvor
EDGE: label=M1:color=255,0,0:adjlist=Grozavesti, Eroilor
EDGE: label=M1:color=255,0,0:adjlist=Semanatoarea, Grozavesti
EDGE: label=M1:color=255,0,0:adjlist=Crangasi, Semanatoarea
EDGE: label=M1:color=255,0,0:adjlist=Crangasi, Basarab
EDGE: label=M1:color=255,0,0:adjlist=Basarab, Gara de Nord
EDGE: label=M1:color=255,0,0:adjlist=Gara de Nord, Piata
    Victoriei
EDGE: label=M1:color=255,0,0:adjlist=N. Grigorescu, Dristor
EDGE: label=M1:color=255,0,0:adjlist=Titan, N. Grigorescu
EDGE: label=M1:color=255,0,0:adjlist=Georgian Costin, Titan
EDGE: label=M1:color=255,0,0:adjlist=Georgian Costin, Republica
EDGE: label=M2:color=0,0,0:adjlist=Aurel Vlaicu, Pipera

EDGE: label=M2: color = 0,0,0: adjlist=Aurel Vlaicu , Aviatorilor
EDGE: label=M2: color = 0,0,0: adjlist=Aviatorilor , Piata Victoriei
EDGE: label=M2: color = 0,0,0: adjlist=Romana, Piata Victoriei
EDGE: label=M2: color = 0,0,0: adjlist=Universtitate , Romana
EDGE: label=M2: color = 0,0,0: adjlist=Universtitate , Piata Unirii
EDGE: label=M2: color = 0,0,0: adjlist=Tineretului , Piata Unirii
EDGE: label=M2: color = 0,0,0: adjlist=Eroii Revolutiei , Tineretului
EDGE: label=M2: color = 0,0,0: adjlist=Constantin Brancoveanu , Eroii
      Revolutiei
EDGE: label=M2: color = 0,0,0: adjlist=Constantin Brancoveanu , Piata
      Sudului
EDGE: label=M2: color = 0,0,0: adjlist=Piata Sudului , Aparatorii
      Patriei
EDGE: label=M2: color = 0,0,0: adjlist=Aparatorii Patriei , IMGB 1
EDGE: label=M2: color = 0,0,0: adjlist=IMGB 1, IMGB Depou
EDGE: label=M3: color = 255,255,0: adjlist=Politehnica , Eroilor
EDGE: label=M3: color = 255,255,0: adjlist=Armata Poporului ,
      Politehnica
EDGE: label=M3: color = 255,255,0: adjlist=Gorjului , Armata
      Poporului
EDGE: label=M3: color = 255,255,0: adjlist=Pacii , Gorjului
EDGE: label=M3: color = 255,255,0: adjlist=Pacii , Industriilor
EDGE: label=T1: color = 150,150,0: adjlist=Grivita , Basarab
EDGE: label=T1: color = 150,150,0: adjlist=1 Mai, Grivita
EDGE: label=T1: color = 150,150,0: adjlist=Pajura , 1 Mai
EDGE: label=T1: color = 150,150,0: adjlist=Pajura , Laromet
EDGE: label=T2: color = 150,150,0: adjlist=N. Grigorescu , 1 Dec.
      1918
EDGE: label=T2: color = 150,150,0: adjlist=Policolor , 1 Dec. 1918
EDGE: label=T2: color = 150,150,0: adjlist=Policolor , Linia de
      centura

B.4   London

NODE: label=Acton Central: x=1206:y=802802
NODE: label=Acton Town: x=1194:y=795795
NODE: label=Aldgate : x=1335:y=812812
NODE: label=Aldgate East : x=1338:y=813813
NODE: label=All Saints : x=1379:y=809809
NODE: label=Alperton : x=1180:y=837837
NODE: label=Amersham: x=964:y=981981
NODE: label=Angel: x=1314:y=832832
NODE: label=Archway: x=1293:y=868868
NODE: label=Arnos Grove: x=1293:y=925925
NODE: label=Arsenal: x=1313:y=860860
NODE: label=Baker Street : x=1279:y=820820
NODE: label=Balham: x=1285:y=732732
NODE: label=Bank: x=1327:y=811811

NODE: label=Barbican:x=1320:y=818818
NODE: label=Barking:x=1444:y=843843
NODE: label=Barkingside:x=1447:y=895895
NODE: label=Barons Court:x=1240:y=783783
NODE: label=Bayswater:x=1258:y=808808
NODE: label=Beckton:x=1431:y=815815
NODE: label=Beckton Park:x=1427:y=808808
NODE: label=Becontree:x=1476:y=845845
NODE: label=Belsize Park:x=1273:y=851851
NODE: label=Bermondsey:x=1345:y=794794
NODE: label=Bethnal Green:x=1349:y=827827
NODE: label=Blackfriars:x=1310:y=813813
NODE: label=Blackhorse Road:x=1358:y=893893
NODE: label=Blackwall:x=1384:y=806806
NODE: label=Bond Street:x=1285:y=811811
NODE: label=Borough:x=1323:y=797797
NODE: label=Boston Manor:x=1163:y=787787
NODE: label=Bounds Green:x=1299:y=914914
NODE: label=Bow Church:x=1373:y=828828
NODE: label=Bow Road:x=1370:y=827827
NODE: label=Brent Cross:x=1238:y=879879
NODE: label=Brixton:x=1312:y=755755
NODE: label=Bromley−by−Bow:x=1379:y=825825
NODE: label=Brondesbury:x=1247:y=844844
NODE: label=Brondesbury Park:x=1242:y=839839
NODE: label=Buckhurst Hill:x=1417:y=939939
NODE: label=Burnt Oak:x=1203:y=907907
NODE: label=Caledonian Road:x=1305:y=849849
NODE: label=Caledonian Road \& Barnsbury:x=1308:y=844844
NODE: label=Camden Road:x=1291:y=842842
NODE: label=Camden Town:x=1289:y=839839
NODE: label=Canada Water:x=1354:y=794794
NODE: label=Canary Wharf:x=1374:y=803803
NODE: label=Canning Town:x=1394:y=813813
NODE: label=Cannon Street:x=1325:y=809809
NODE: label=Canonbury:x=1323:y=850850
NODE: label=Canons Park:x=1181:y=912912
NODE: label=Chalfont \& Latimer:x=996:y=975975
NODE: label=Chalk Farm:x=1281:y=844844
NODE: label=Chancery Lane:x=1311:y=816816
NODE: label=Charing Cross:x=1303:y=803803
NODE: label=Chesham:x=960:y=10161016
NODE: label=Chigwell:x=1437:y=930930
NODE: label=Chiswick Park:x=1203:y=787787
NODE: label=Chorleywood:x=1026:y=961961
NODE: label=Clapham Common:x=1294:y=753753
NODE: label=Clapham North:x=1300:y=756756
NODE: label=Clapham South:x=1287:y=742742

NODE: label=Cockfosters:x=1281:y=963963
NODE: label=Colindale:x=1213:y=899899
NODE: label=Colliers Wood:x=1268:y=704704
NODE: label=Covent Garden:x=1302:y=810810
NODE: label=Crossharbour \& London Arena:x=1379:y=793793
NODE: label=Croxley:x=1079:y=954954
NODE: label=Custom House:x=1406:y=809809
NODE: label=Cutty Sark:x=1382:y=777777
NODE: label=Cyprus:x=1433:y=808808
NODE: label=Dagenham East:x=1501:y=850850
NODE: label=Dagenham Heathway:x=1489:y=847847
NODE: label=Dalston Kingsland:x=1335:y=850850
NODE: label=Debden:x=1442:y=961961
NODE: label=Deptford Bridge:x=1374:y=769769
NODE: label=Devons Road:x=1376:y=822822
NODE: label=Dollis Hill:x=1221:y=851851
NODE: label=Ealing Broadway:x=1179:y=809809
NODE: label=Ealing Common:x=1188:y=804804
NODE: label=Earl's Court:x=1254:y=784784
NODE: label=East Acton:x=1216:y=812812
NODE: label=East Finchley:x=1272:y=892892
NODE: label=East Ham:x=1424:y=842842
NODE: label=East India:x=1387:y=808808
NODE: label=East Putney:x=1243:y=748748
NODE: label=Eastcote:x=1111:y=876876
NODE: label=Edgware:x=1195:y=919919
NODE: label=Edgware Road:x=1272:y=817817
NODE: label=Elephant \& Castle:x=1319:y=791791
NODE: label=Elm Park:x=1525:y=856856
NODE: label=Elverson Road:x=1378:y=762762
NODE: label=Embankment:x=1304:y=803803
NODE: label=Epping:x=1461:y=10151015
NODE: label=Euston:x=1295:y=827827
NODE: label=Euston Square:x=1294:y=823823
NODE: label=Fairlop:x=1449:y=906906
NODE: label=Farringdon:x=1315:y=818818
NODE: label=Finchley Central:x=1253:y=906906
NODE: label=Finchley Road:x=1262:y=847847
NODE: label=Finchley Road \& Frognal:x=1260:y=850850
NODE: label=Finsbury Park:x=1313:y=867867
NODE: label=Fulham Broadway:x=1254:y=772772
NODE: label=Gallions Reach:x=1438:y=809809
NODE: label=Gants Hill:x=1432:y=884884
NODE: label=Gloucester Road:x=1262:y=788788
NODE: label=Golders Green:x=1252:y=874874
NODE: label=Goldhawk Road:x=1231:y=795795
NODE: label=Goodge Street:x=1295:y=818818
NODE: label=Gospel Oak:x=1282:y=856856

NODE: label=Grange Hill:x=1449:y=925925
NODE: label=Great Portland Street:x=1288:y=821821
NODE: label=Green Park:x=1289:y=802802
NODE: label=Greenford:x=1147:y=839839
NODE: label=Greenwich:x=1379:y=773773
NODE: label=Gunnersbury:x=1197:y=783783
NODE: label=Hackney Central:x=1349:y=849849
NODE: label=Hackney Wick:x=1370:y=845845
NODE: label=Hainault:x=1450:y=914914
NODE: label=Hammersmith:x=1233:y=786786
NODE: label=Hampstead:x=1263:y=857857
NODE: label=Hampstead Heath:x=1272:y=856856
NODE: label=Hanger Lane:x=1184:y=826826
NODE: label=Harlesden:x=1209:y=833833
NODE: label=Harrow \& Wealdstone:x=1154:y=894894
NODE: label=Harrow−on−the−Hill:x=1153:y=880880
NODE: label=Hatton Cross:x=1096:y=753753
NODE: label=Heathrow Terminals 1, 2 \& 3:x=1075:y=758758
NODE: label=Heathrow Terminal 4:x=1080:y=744744
NODE: label=Hendon Central:x=1229:y=886886
NODE: label=Heron Quays:x=1374:y=801801
NODE: label=High Barnet:x=1249:y=962962
NODE: label=High Street Kensington:x=1255:y=794794
NODE: label=Highbury \& Islington:x=1315:y=847847
NODE: label=Highgate:x=1285:y=881881
NODE: label=Hillingdon:x=1074:y=850850
NODE: label=Holborn:x=1305:y=815815
NODE: label=Holland Park:x=1246:y=802802
NODE: label=Holloway Road:x=1309:y=854854
NODE: label=Homerton:x=1358:y=849849
NODE: label=Hornchurch:x=1538:y=862862
NODE: label=Hounslow Central:x=1135:y=759759
NODE: label=Hounslow East:x=1142:y=762762
NODE: label=Hounslow West:x=1122:y=761761
NODE: label=Hyde Park Corner:x=1282:y=797797
NODE: label=Ickenham:x=1081:y=859859
NODE: label=Island Gardens:x=1382:y=784784
NODE: label=Kennington:x=1316:y=782782
NODE: label=Kensal Green:x=1232:y=827827
NODE: label=Kensal Rise:x=1228:y=835835
NODE: label=Kensington (Olympia):x=1243:y=792792
NODE: label=Kentish Town:x=1290:y=851851
NODE: label=Kentish Town West:x=1286:y=847847
NODE: label=Kenton:x=1167:y=883883
NODE: label=Kew Gardens:x=1192:y=767767
NODE: label=Kilburn:x=1245:y=846846
NODE: label=Kilburn Park:x=1253:y=833833
NODE: label=King's Cross St. Pancras:x=1302:y=830830

NODE:label=Kingsbury:x=1193:y=887887
NODE:label=Knightsbridge:x=1276:y=795795
NODE:label=Ladbroke Grove:x=1242:y=813813
NODE:label=Lambeth North:x=1312:y=794794
NODE:label=Lancaster Gate:x=1266:y=807807
NODE:label=Latimer Road:x=1237:y=809809
NODE:label=Leicester Square:x=1299:y=808808
NODE:label=Lewisham:x=1381:y=759759
NODE:label=Leyton:x=1383:y=861861
NODE:label=Leytonstone:x=1392:y=874874
NODE:label=Limehouse:x=1361:y=811811
NODE:label=Liverpool Street:x=1331:y=815815
NODE:label=London Bridge:x=1327:y=802802
NODE:label=Loughton:x=1423:y=956956
NODE:label=Maida Vale:x=1259:y=827827
NODE:label=Manor House:x=1320:y=874874
NODE:label=Mansion House:x=1323:y=809809
NODE:label=Marble Arch:x=1277:y=810810
NODE:label=Marylebone:x=1275:y=819819
NODE:label=Mile End:x=1365:y=825825
NODE:label=Mill Hill East:x=1240:y=914914
NODE:label=Monument:x=1328:y=808808
NODE:label=Moor Park:x=1085:y=934934
NODE:label=Moorgate:x=1326:y=816816
NODE:label=Morden:x=1256:y=686686
NODE:label=Mornington Crescent:x=1291:y=833833
NODE:label=Mudchute:x=1379:y=788788
NODE:label=Neasden:x=1213:y=853853
NODE:label=New Cross:x=1368:y=770770
NODE:label=New Cross Gate:x=1362:y=769769
NODE:label=Newbury Park:x=1449:y=883883
NODE:label=North Acton:x=1208:y=819819
NODE:label=North Ealing:x=1188:y=804804
NODE:label=North Greenwich:x=1391:y=798798
NODE:label=North Harrow:x=1135:y=886886
NODE:label=North Wembley:x=1176:y=862862
NODE:label=North Woolwich:x=1432:y=798798
NODE:label=Northfields:x=1171:y=792792
NODE:label=Northolt:x=1132:y=845845
NODE:label=Northwick Park:x=1166:y=879879
NODE:label=Northwood:x=1091:y=913913
NODE:label=Northwood Hills:x=1102:y=902902
NODE:label=Notting Hill Gate:x=1252:y=804804
NODE:label=Oakwood:x=1293:y=959959
NODE:label=Old Street:x=1326:y=825825
NODE:label=Osterley:x=1145:y=771771
NODE:label=Oval:x=1310:y=775775
NODE:label=Oxford Circus:x=1290:y=812812

NODE: label=Paddington : x=1266:y=812812
NODE: label=Park Royal : x=1191:y=822822
NODE: label=Parsons Green : x=1250:y=766766
NODE: label=Perivale : x=1163:y=833833
NODE: label=Piccadilly Circus : x=1294:y=805809
NODE: label=Pimlico : x=1296:y=784784
NODE: label=Pinner : x=1122:y=894894
NODE: label=Plaistow : x=1399:y=833833
NODE: label=Poplar : x=1376:y=806806
NODE: label=Preston Road : x=1182:y=873873
NODE: label=Prince Regent : x=1412:y=809809
NODE: label=Pudding Mill Lane : x=1378:y=835835
NODE: label=Putney Bridge : x=1244:y=758758
NODE: label=Queen's Park : x=1245:y=832832
NODE: label=Queensbury : x=1188:y=897897
NODE: label=Queensway : x=1258:y=806806
NODE: label=Ravenscourt Park : x=1225:y=787787
NODE: label=Rayners Lane : x=1129:y=875875
NODE: label=Redbridge : x=1418:y=883883
NODE: label=Regent's Park : x=1286:y=821821
NODE: label=Richmond : x=1181:y=751751
NODE: label=Rickmansworth : x=1056:y=945945
NODE: label=Roding Valley : x=1415:y=928928
NODE: label=Rotherhithe : x=1352:y=798798
NODE: label=Royal Albert : x=1420:y=808808
NODE: label=Royal Oak : x=1257:y=815815
NODE: label=Royal Victoria : x=1401:y=808808
NODE: label=Ruislip : x=1095:y=870870
NODE: label=Ruislip Gardens : x=1103:y=858858
NODE: label=Ruislip Manor : x=1101:y=872872
NODE: label=Russell Square : x=1301:y=821821
NODE: label=Seven Sisters : x=1335:y=888888
NODE: label=Shadwell : x=1350:y=809809
NODE: label=Shepherd's Bush : x=1231:y=800800
NODE: label=Shepherd's Bush (Central) : x=1231:y=801801
NODE: label=Shoreditch : x=1339:y=822822
NODE: label=Silvertown : x=1420:y=801801
NODE: label=Sloane Square : x=1281:y=786786
NODE: label=Snaresbrook : x=1401:y=887887
NODE: label=South Acton : x=1201:y=792792
NODE: label=South Ealing : x=1175:y=793793
NODE: label=South Harrow : x=1143:y=863863
NODE: label=South Kensington : x=1269:y=788788
NODE: label=South Kenton : x=1173:y=870870
NODE: label=South Quay : x=1375:y=798798
NODE: label=South Ruislip : x=1110:y=854854
NODE: label=South Wimbledon : x=1258:y=700700
NODE: label=South Woodford : x=1405:y=900900

NODE: label=Southfields:x=1247:y=733733
NODE: label=Southgate:x=1296:y=942942
NODE: label=Southwark:x=1315:y=800800
NODE: label=St. James's Park:x=1296:y=794794
NODE: label=St. John's Wood:x=1267:y=833833
NODE: label=St. Paul's:x=1321:y=815815
NODE: label=Stamford Brook:x=1218:y=787787
NODE: label=Stanmore:x=1175:y=925925
NODE: label=Stepney Green:x=1356:y=821821
NODE: label=Stockwell:x=1304:y=765765
NODE: label=Stonebridge Park:x=1196:y=842842
NODE: label=Stratford:x=1385:y=843843
NODE: label=Sudbury Hill:x=1154:y=855855
NODE: label=Sudbury Town:x=1168:y=848848
NODE: label=Surrey Quays:x=1356:y=789789
NODE: label=Swiss Cottage:x=1266:y=843843
NODE: label=Temple:x=1309:y=808808
NODE: label=Theydon Bois:x=1455:y=991991
NODE: label=Tooting Bec:x=1280:y=723723
NODE: label=Tooting Broadway:x=1274:y=713713
NODE: label=Tottenham Court Road:x=1297:y=814809
NODE: label=Tottenham Hale:x=1345:y=895895
NODE: label=Totteridge \& Whetstone:x=1261:y=939939
NODE: label=Tower Gateway:x=1336:y=808808
NODE: label=Tower Hill:x=1335:y=807807
NODE: label=Tufnell Park:x=1291:y=858858
NODE: label=Turnham Green:x=1212:y=787787
NODE: label=Turnpike Lane:x=1315:y=896896
NODE: label=Upminster:x=1560:y=868868
NODE: label=Upminster Bridge:x=1551:y=867867
NODE: label=Upney:x=1458:y=843843
NODE: label=Upton Park:x=1412:y=837837
NODE: label=Uxbridge:x=1056:y=842842
NODE: label=Vauxhall:x=1304:y=780780
NODE: label=Victoria:x=1288:y=789789
NODE: label=Walthamstow Central:x=1373:y=889889
NODE: label=Wanstead:x=1406:y=882882
NODE: label=Wapping:x=1350:y=801801
NODE: label=Warren Street:x=1292:y=822822
NODE: label=Warwick Avenue:x=1260:y=821821
NODE: label=Waterloo:x=1310:y=799799
NODE: label=Watford:x=1095:y=965965
NODE: label=Wembley Central:x=1182:y=851851
NODE: label=Wembley Park:x=1192:y=863863
NODE: label=West Acton:x=1193:y=813813
NODE: label=West Brompton:x=1254:y=779779
NODE: label=West Finchley:x=1255:y=916916
NODE: label=West Ham:x=1392:y=829829

NODE: label=West Hampstead:x=1255:y=846846
NODE: label=West Harrow:x=1141:y=880880
NODE: label=West India Quay:x=1374:y=805805
NODE: label=West Kensington:x=1246:y=783783
NODE: label=West Ruislip:x=1084:y=867867
NODE: label=Westbourne Park:x=1248:y=817817
NODE: label=Westferry:x=1370:y=807807
NODE: label=Westminster:x=1302:y=797797
NODE: label=White City:x=1233:y=807807
NODE: label=Whitechapel:x=1346:y=818818
NODE: label=Willesden Green:x=1233:y=848848
NODE: label=Willesden Junction:x=1218:y=829829
NODE: label=Wimbledon:x=1248:y=706706
NODE: label=Wimbledon Park:x=1252:y=721721
NODE: label=Wood Green:x=1310:y=904904
NODE: label=Woodford:x=1409:y=917917
NODE: label=Woodside Park:x=1257:y=925925
EDGE: label=X:color=255,255,255:adjlist=Tower Hill, Tower
    Gateway
EDGE: label=X:color=255,255,255:adjlist=Bank, Monument
EDGE: label=X:color=255,255,255:adjlist=Bow Road, Bow Church
EDGE: label=Victoria:color=60,140,255:adjlist=Brixton, Stockwell
    , Vauxhall, Pimlico, Victoria, Green Park, Oxford Circus,
    Warren Street, Euston, King's Cross St. Pancras, Highbury \&
     Islington, Finsbury Park, Seven Sisters, Tottenham Hale,
    Blackhorse Road, Walthamstow Central
EDGE: label=District:color=20,150,0:adjlist=Kensington (Olympia)
    , Earl's Court
EDGE: label=District:color=20,150,0:adjlist=Ealing Broadway,
    Ealing Common, Acton Town, Chiswick Park, Turnham Green
EDGE: label=District:color=20,150,0:adjlist=Richmond, Kew
    Gardens, Gunnersbury, Turnham Green, Stamford Brook,
    Ravenscourt Park, Hammersmith, Barons Court, West Kensington
    , Earl's Court, South Kensington, Sloane Square, Victoria,
    St. James's Park, Westminster, Embankment, Temple,
    Blackfriars, Mansion House, Cannon Street, Monument, Tower
    Hill, Aldgate East, Whitechapel, Stepney Green, Mile End,
    Bow Road, Bow Church, Bromley−by−Bow, West Ham, Plaistow,
    Upton Park, East Ham, Barking, Upney, Becontree, Dagenham
    Heathway, Dagenham East, Elm Park, Hornchurch, Upminster
    Bridge, Upminster
EDGE: label=District:color=20,150,0:adjlist=Wimbledon, Wimbledon
     Park, Southfields, East Putney, Putney Bridge, Parsons
    Green, Fulham Broadway, West Brompton, Earl's Court, High
    Street Kensington, Notting Hill Gate, Bayswater, Paddington,
     Edgware Road

EDGE: label=Bakerloo : color =100,80,10: adjlist=Harrow \&
Wealdstone , Kenton , South Kenton , North Wembley , Wembley
Central , Stonebridge Park , Harlesden , Willesden Junction ,
Kensal Green , Queen's Park , Kilburn Park , Maida Vale ,
Warwick Avenue , Paddington , Edgware Road , Marylebone , Baker
Street , Regent's Park , Oxford Circus , Piccadilly Circus ,
Charing Cross , Embankment , Waterloo , Lambeth North , Elephant
\& Castle

EDGE: label=Metropolitan : color =140,80,90: adjlist=Uxbridge ,
Hillingdon , Ickenham , Ruislip , Ruislip Manor , Eastcote ,
Rayners Lane , West Harrow , Harrow−on−the−Hill , Northwick
Park , Preston Road , Wembley Park , Finchley Road , Baker
Street , Great Portland Street , Euston Square , King's Cross
St . Pancras , Farringdon , Barbican , Moorgate , Liverpool
Street , Aldgate

EDGE: label=Metropolitan : color =140,80,90: adjlist=Amersham ,
Chalfont \& Latimer , Chorleywood , Rickmansworth , Moor Park ,
Northwood , Northwood Hills , Pinner , North Harrow , Harrow−on−
the−Hill

EDGE: label=Metropolitan : color =140,80,90: adjlist=Chesham ,
Chalfont \& Latimer

EDGE: label=Metropolitan : color =140,80,90: adjlist=Watford ,
Croxley , Moor Park

EDGE: label=Hammersmith \& City : color =255,100,160: adjlist=
Hammersmith , Goldhawk Road , Shepherd's Bush , Latimer Road ,
Ladbroke Grove , Westbourne Park , Royal Oak , Paddington ,
Edgware Road , Baker Street , Great Portland Street , Euston
Square , King's Cross St . Pancras , Farringdon , Barbican ,
Moorgate , Liverpool Street , Aldgate East , Whitechapel ,
Stepney Green , Mile End , Bow Road , Bow Church , Bromley−by−
Bow , West Ham , Plaistow , Upton Park , East Ham , Barking

EDGE: label=Circle : color =255,255,0: adjlist=Paddington , Bayswater
, Notting Hill Gate , High Street Kensington , Gloucester Road
, South Kensington , Sloane Square , Victoria , St . James's
Park , Westminster , Embankment , Temple , Blackfriars , Mansion
House , Cannon Street , Monument , Tower Hill , Aldgate ,
Liverpool Street , Moorgate , Barbican , Farringdon , King's
Cross St . Pancras , Euston Square , Great Portland Street ,
Baker Street , Edgware Road , Paddington

EDGE: label=Central : color =255,0,0: adjlist=West Ruislip , Ruislip
Gardens , South Ruislip , Northolt , Greenford , Perivale ,
Hanger Lane , North Acton , East Acton , White City , Shepherd's
Bush (Central) , Holland Park , Notting Hill Gate , Queensway ,
Lancaster Gate , Marble Arch , Bond Street , Oxford Circus ,
Tottenham Court Road , Holborn , Chancery Lane , St . Paul's ,
Bank , Liverpool Street , Bethnal Green , Mile End , Stratford ,
Leyton , Leytonstone , Snaresbrook , South Woodford , Woodford ,
Buckhurst Hill , Loughton , Debden , Theydon Bois , Epping

EDGE: label=Central:color=255,0,0:adjlist=Ealing Broadway, West
    Acton, North Acton
EDGE: label=Central:color=255,0,0:adjlist=Leytonstone, Wanstead,
    Redbridge, Gants Hill, Newbury Park, Barkingside, Fairlop,
    Hainault, Grange Hill, Chigwell, Roding Valley, Woodford
EDGE: label=Piccadilly:color=0,0,200:adjlist=Uxbridge,
    Hillingdon, Ickenham, Ruislip, Ruislip Manor, Eastcote,
    Rayners Lane, South Harrow, Sudbury Hill, Sudbury Town,
    Alperton, Park Royal, North Ealing, Ealing Common, Acton
    Town, Turnham Green, Hammersmith, Barons Court, Earl's Court
    , Gloucester Road, South Kensington, Knightsbridge, Hyde
    Park Corner, Green Park, Piccadilly Circus, Leicester Square
    , Covent Garden, Holborn, Russell Square, King's Cross St.
    Pancras, Caledonian Road, Holloway Road, Arsenal, Finsbury
    Park, Manor House, Turnpike Lane, Wood Green, Bounds Green,
    Arnos Grove, Southgate, Oakwood, Cockfosters
EDGE: label=Piccadilly:color=0,0,200:adjlist=Hatton Cross,
    Heathrow Terminals 1, 2 \& 3, Heathrow Terminal 4, Hatton
    Cross, Hounslow West, Hounslow Central, Hounslow East,
    Osterley, Boston Manor, Northfields, South Ealing, Acton
    Town
EDGE: label=Jubilee:color=128,128,128:adjlist=Stanmore, Canons
    Park, Queensbury, Kingsbury, Wembley Park, Neasden, Dollis
    Hill, Willesden Green, Kilburn, West Hampstead, Finchley
    Road, Swiss Cottage, St. John's Wood, Baker Street, Bond
    Street, Green Park, Westminster, Waterloo, Southwark, London
    Bridge, Bermondsey, Canada Water, Canary Wharf, North
    Greenwich, Canning Town, West Ham, Stratford
EDGE: label=Northern:color=0,0,0:adjlist=Edgware, Burnt Oak,
    Colindale, Hendon Central, Brent Cross, Golders Green,
    Hampstead, Belsize Park, Chalk Farm, Camden Town, Mornington
    Crescent, Euston, Warren Street, Goodge Street, Tottenham
    Court Road, Leicester Square, Charing Cross, Embankment,
    Waterloo, Kennington, Oval, Stockwell, Clapham North,
    Clapham Common, Clapham South, Balham, Tooting Bec, Tooting
    Broadway, Colliers Wood, South Wimbledon, Morden
EDGE: label=Northern:color=0,0,0:adjlist=High Barnet, Totteridge
    \& Whetstone, Woodside Park, West Finchley, Finchley
    Central, East Finchley, Highgate, Archway, Tufnell Park,
    Kentish Town, Camden Town, Euston, King's Cross St. Pancras,
    Angel, Old Street, Moorgate, Bank, Monument, London Bridge,
    Borough, Elephant \& Castle, Kennington
EDGE: label=Northern:color=0,0,0:adjlist=Mill Hill East,
    Finchley Central
EDGE: label=Waterloo \& City:color=0,255,160:adjlist=Waterloo,
    Bank

EDGE: label=East London: color=255,190,40: adjlist=Shoreditch,
    Whitechapel, Shadwell, Wapping, Rotherhithe, Canada Water,
    Surrey Quays, New Cross Gate
EDGE: label=East London: color=255,190,40: adjlist=Surrey Quays,
    New Cross
EDGE: label=Docklands Light Rail: color=0,255,255: adjlist=
    Monument, Shadwell, Limehouse, Westferry, Poplar, Blackwall,
     East India, Canning Town, Royal Victoria, Custom House,
    Prince Regent, Royal Albert, Beckton Park, Cyprus, Gallions
    Reach, Beckton
EDGE: label=Docklands Light Rail: color=0,255,255: adjlist=
    Stratford, Pudding Mill Lane, Bow Church, Devons Road, All
    Saints, Poplar, West India Quay, Canary Wharf, Heron Quays,
    South Quay, Crossharbour \& London Arena, Mudchute, Island
    Gardens, Cutty Sark, Greenwich, Deptford Bridge, Elverson
    Road, Lewisham
EDGE: label=Docklands Light Rail: color=0,255,255: adjlist=
    Westferry, West India Quay
EDGE: label=Docklands Light Rail: color=0,255,255: adjlist=Tower
    Gateway, Shadwell
EDGE: label=Docklands Light Rail: color=0,255,255: adjlist=Canning
     Town
EDGE: label=National Rail: color=0,255,0: adjlist=Richmond, Kew
    Gardens, Gunnersbury, South Acton, Acton Central, Willesden
    Junction, Kensal Rise, Brondesbury Park, Brondesbury, West
    Hampstead, Finchley Road \& Frognal, Hampstead Heath, Gospel
     Oak, Kentish Town West, Camden Road, Caledonian Road \&
    Barnsbury, Highbury \& Islington, Canonbury, Dalston
    Kingsland, Hackney Central, Homerton, Hackney Wick,
    Stratford, West Ham, Canning Town, Custom House, Silvertown,
     North Woolwich

B.5 Madrid

NODE: label=Congosto: x=−3.6186: y=40.3713
NODE: label=Villa de Vallecas: x=−3.6274: y=40.3815
NODE: label=Sierra de Guadalupe: x=−3.6348: y=40.3833
NODE: label=Miguel Hernandez: x=−3.6406: y=40.3876
NODE: label=Alto del Arenal: x=−3.6465: y=40.3900
NODE: label=Buenos Aires: x=−3.6529: y=40.3914
NODE: label=Portazgo: x=−3.6601: y=40.3933
NODE: label=Nueva Numancia: x=−3.6630: y=40.3950
NODE: label=Puente de Vallecas: x=−3.6686: y=40.3979
NODE: label=Pacifico: x=−3.6750: y=40.4013
NODE: label=Menendez Pelayo: x=−3.6809: y=40.4043
NODE: label=Atocha Renfe: x=−3.6865: y=40.4058
NODE: label=Atocha: x=−3.6926: y=40.4088
NODE: label=Anton Martin: x=−3.6987: y=40.4121

NODE:label=Tirso de Molina:x=−3.704:y=40.4123
NODE:label=Sol:x=−3.7035:y=40.4167
NODE:label=Gran Via:x=−3.7016:y=40.4200
NODE:label=Tribunal:x=−3.7009:y=40.4254
NODE:label=Bilbao:x=−3.7023:y=40.4291
NODE:label=Iglesia:x=−3.6987:y=40.4349
NODE:label=Rios Rosas:x=−3.7018:y=40.4420
NODE:label=Cuatro Caminos:x=−3.7038:y=40.4468
NODE:label=Alvarado:x=−3.7034:y=40.4502
NODE:label=Estrecho:x=−3.7027:y=40.4543
NODE:label=Tetuan:x=−3.6981:y=40.4613
NODE:label=Valdeacederas:x=−3.6957:y=40.4643
NODE:label=Plaza de Castilla:x=−3.6893:y=40.4660
NODE:label=Canal:x=−3.7043:y=40.4385
NODE:label=Quevedo:x=−3.7048:y=40.4334
NODE:label=San Bernardo:x=−3.7062:y=40.4297
NODE:label=Noviciado:x=−3.7073:y=40.4255
NODE:label=Santo Domingo:x=−3.7084:y=40.4202
NODE:label=Opera:x=−3.7101:y=40.4178
NODE:label=Sevilla:x=−3.6977:y=40.4184
NODE:label=Banco de Espana:x=−3.6935:y=40.4194
NODE:label=Retiro:x=−3.6857:y=40.4206
NODE:label=Principe de Vergara:x=−3.6802:y=40.4227
NODE:label=Goya:x=−3.6756:y=40.4248
NODE:label=Manuel Becerra:x=−3.6690:y=40.4279
NODE:label=Ventas:x=−3.6632:y=40.4320
NODE:label=Legazpi:x=−3.6950:y=40.3912
NODE:label=Delicias:x=−3.6942:y=40.3984
NODE:label=Palos de la Frontera:x=−3.6960:y=40.4031
NODE:label=Embajadores:x=−3.7022:y=40.4049
NODE:label=Lavapies:x=−3.7011:y=40.4087
NODE:label=Callao:x=−3.7058:y=40.4202
NODE:label=Plaza de Espana:x=−3.7109:y=40.4234
NODE:label=Ventura Rodriguez:x=−3.7139:y=40.4272
NODE:label=Arguelles:x=−3.7163:y=40.4307
NODE:label=Moncloa:x=−3.7193:y=40.4347
NODE:label=Alonso Martinez:x=−3.6956:y=40.4275
NODE:label=Colon:x=−3.6907:y=40.4252
NODE:label=Serrano:x=−3.6878:y=40.4253
NODE:label=Velazquez:x=−3.6836:y=40.4251
NODE:label=Lista:x=−3.6754:y=40.4299
NODE:label=Diego de Leon:x=−3.6747:y=40.4347
NODE:label=Avda de America:x=−3.6772:y=40.4374
NODE:label=Prosperidad:x=−3.6745:y=40.4439
NODE:label=Alfonso XIII:x=−3.6678:y=40.4484
NODE:label=Avda de la Paz:x=−3.6625:y=40.4534
NODE:label=Arturo Soria:x=−3.6563:y=40.4560
NODE:label=Esperanza:x=−3.6457:y=40.4593

NODE: label=Canillas:x=−3.6354:y=40.4646
NODE: label=Mar de Cristal:x=−3.6389:y=40.4695
NODE: label=San Lorenzo:x=−3.6384:y=40.4765
NODE: label=Parque de Santa Maria:x=−3.6460:y=40.4796
NODE: label=Casa de Campo:x=−3.7668:y=40.3992
NODE: label=Campamento:x=−3.7680:y=40.3943
NODE: label=Empalme:x=−3.7665:y=40.3908
NODE: label=Aluche:x=−3.7609:y=40.3864
NODE: label=Eugenia de Montijo:x=−3.7494:y=40.3852
NODE: label=Carabanchel:x=−3.7447:y=40.3878
NODE: label=Vista Alegre:x=−3.7400:y=40.3889
NODE: label=Oporto:x=−3.7321:y=40.3883
NODE: label=Urgel:x=−3.7242:y=40.3930
NODE: label=Marques de Vadillo:x=−3.7163:y=40.3974
NODE: label=Piramides:x=−3.7146:y=40.4008
NODE: label=Acacias:x=−3.7064:y=40.4037
NODE: label=Puerto de Toledo:x=−3.7113:y=40.4065
NODE: label=La Latina:x=−3.7083:y=40.4111
NODE: label=Chueca:x=−3.6973:y=40.4228
NODE: label=Ruben Dario:x=−3.6890:y=40.4334
NODE: label=Nunez de Balboa:x=−3.6830:y=40.4323
NODE: label=El Carmen:x=−3.6555:y=40.4321
NODE: label=Quintana:x=−3.6490:y=40.4329
NODE: label=Pueblo Nuevo:x=−3.6443:y=40.4352
NODE: label=Ciudad Lineal:x=−3.6384:y=40.4377
NODE: label=Suanzes:x=−3.6269:y=40.4409
NODE: label=Torre Arias:x=−3.6172:y=40.4436
NODE: label=Canillejas:x=−3.6080:y=40.4493
NODE: label=Mendez Alvaro:x=−3.6804:y=40.3957
NODE: label=Conde de Casal:x=−3.6700:y=40.4084
NODE: label=Sainz de Baranda:x=−3.6692:y=40.4163
NODE: label=O'Donnell:x=−3.6689:y=40.4230
NODE: label=Republica Argentina:x=−3.6853:y=40.4450
NODE: label=Nuevos Ministerios:x=−3.6921:y=40.4465
NODE: label=Guzman el Bueno:x=−3.7125:y=40.4465
NODE: label=Metropolitano:x=−3.7183:y=40.4466
NODE: label=Ciudad Universitaria:x=−3.72750:y=40.4438
NODE: label=Principe Rio:x=−3.72010:y=40.4203
NODE: label=Puerta del Angel:x=−3.72690:y=40.4138
NODE: label=Alto de Extremadura:x=−3.73780:y=40.4101
NODE: label=Lucero:x=−3.74530:y=40.4046
NODE: label=Laguna:x=−3.74450:y=40.3990
NODE: label=Carpetana:x=−3.74160:y=40.3923
NODE: label=Opanel:x=−3.72350:y=40.3853
NODE: label=Plaza Eliptica:x=−3.71650:y=40.3851
NODE: label=Usera:x=−3.70750:y=40.3868
NODE: label=Pitis:x=−3.72760:y=40.4940
NODE: label=Lacoma:x=−3.72360:y=40.4851

NODE: label=Avda Ilustracion : x=−3.71610:y=40.4769
NODE: label=Penagrande : x=−3.72080:y=40.4744
NODE: label=Antonio Machado : x=−3.71960:y=40.4679
NODE: label=Valdezarza : x=−3.7158:y=40.4665
NODE: label=Francos Rodriguez : x=−3.7115:y=40.4571
NODE: label=Islas Filipinas : x=−3.7131:y=40.4390
NODE: label=Alonso Cano : x=−3.6982:y=40.4383
NODE: label=Gregorio Maranon : x=−3.6912:y=40.4379
NODE: label=Cartagena : x=−3.6726:y=40.4396
NODE: label=Parque de las Avenidas : x=−3.6630:y=40.4390
NODE: label=Barrio de la Concepcion : x=−3.6520:y=40.4385
NODE: label=Ascao : x=−3.6416:y=40.4301
NODE: label=Garcia Noblejas : x=−3.6320:y=40.4291
NODE: label=Simancas : x=−3.6254:y=40.4279
NODE: label=San Blas : x=−3.6178:y=40.4270
NODE: label=Las Musas : x=−3.6077:y=40.4329
NODE: label=Barajas : x=−3.5826:y=40.4764
NODE: label=Aeropuerto : x=−3.5710:y=40.4682
NODE: label=Campo de las Naciones : x=−3.6164:y=40.4639
NODE: label=Colombia : x=−3.6768:y=40.4575
NODE: label=Herrera Oria : x=−3.7089:y=40.4852
NODE: label=Barrio del Pilar : x=−3.7038:y=40.4773
NODE: label=Ventilla : x=−3.6953:y=40.4701
NODE: label=Duque de Pastrana : x=−3.6791:y=40.4675
NODE: label=Pio XII : x=−3.6762:y=40.4627
NODE: label=Concha Espina : x=−3.6779:y=40.4523
NODE: label=Cruz del Rayo : x=−3.6786:y=40.4427
NODE: label=Ibiza : x=−3.6778:y=40.4183
NODE: label=Estrella : x=−3.6598:y=40.4106
NODE: label=Vinateros : x=−3.6524:y=40.4103
NODE: label=Artilleros : x=−3.6443:y=40.4069
NODE: label=Pavones : x=−3.6353:y=40.4007
NODE: label=Valdebernardo : x=−3.6215:y=40.3999
NODE: label=Vicalvaro : x=−3.6090:y=40.4041
NODE: label=San Cipriano : x=−3.6038:y=40.4039
NODE: label=Puerta de Arganda : x=−3.5968:y=40.4017
NODE: label=Rivas Urbanizaciones : x=−3.5443:y=40.3644
NODE: label=Rivas Vaciamadrid : x=−3.5149:y=40.3222
NODE: label=La Poveda : x=−3.4786:y=40.3163
NODE: label=Arganda del Rey : x=−3.4540:y=40.3017
NODE: label=Fuencarral : x=−3.6931:y=40.4951
NODE: label=Begona : x=−3.6859:y=40.4803
NODE: label=Chamartin : x=−3.6825:y=40.4720
NODE: label=Cuzco : x=−3.6902:y=40.4585
NODE: label=Santiago Bernabeu : x=−3.6903:y=40.4521
NODE: label=Lago : x=−3.7347:y=40.4174
NODE: label=Batan : x=−3.7524:y=40.4082
NODE: label=Colonia Jardin : x=−3.7744:y=40.3970

NODE: label=Cuatro Vientos : x=−3.7912:y=40.3779
NODE: label=Joaquin Vilumbrales : x=−3.8077:y=40.3498
NODE: label=Puerta del Sur : x=−3.8122:y=40.3452
NODE: label=Abrantes : x=−3.7273:y=40.3816
NODE: label=Pan Bendito : x=−3.7346:y=40.3757
NODE: label=Leganes Central : x=−3.7712:y=40.3279
NODE: label=San Nicasio : x=−3.7729:y=40.3361
NODE: label=Hospital Severo Ochoa : x=−3.7672:y=40.3223
NODE: label=Casa del Reloj : x=−3.7596:y=40.3268
NODE: label=Julian Besteiro : x=−3.7528:y=40.3346
NODE: label=El Carrascal : x=−3.7421:y=40.3362
NODE: label=El Bercial : x=−3.7361:y=40.3271
NODE: label=Los Espartales : x=−3.7187:y=40.3247
NODE: label=El Casar : x=−3.7110:y=40.3188
NODE: label=Juan de la Cierva : x=−3.7227:y=40.3115
NODE: label=Getafe Central : x=−3.7335:y=40.3099
NODE: label=Alonso de Mendoza : x=−3.7358:y=40.3010
NODE: label=Conservatorio : x=−3.7453:y=40.2934
NODE: label=Arroyo Culebro : x=−3.7568:y=40.2891
NODE: label=Parque de los Estados : x=−3.7875:y=40.2869
NODE: label=Fuenlabrada Central : x=−3.7992:y=40.2833
NODE: label=Parque Europa : x=−3.8059:y=40.2852
NODE: label=Hospital de Fuenlabrada : x=−3.8132:y=40.2853
NODE: label=Loranca : x=−3.8351:y=40.2966
NODE: label=Manuela Malasana : x=−3.8648:y=40.3091
NODE: label=Hospital de Mostoles : x=−3.8744:y=40.3161
NODE: label=Pradillo : x=−3.8653:y=40.3217
NODE: label=Mostoles Central : x=−3.8635:y=40.3288
NODE: label=Universidad Rey Juan Carlos : x=−3.8732:y=40.3346
NODE: label=Parque Oeste : x=−3.8496:y=40.3459
NODE: label=Alcorcon Central : x=−3.8319:y=40.3502
NODE: label=Parque Lisboa : x=−3.8208:y=40.3492
EDGE: label=1:color=51,153,204: adjlist=Plaza de Castilla ,
    Valdeacederas , Tetuan , Estrecho , Alvarado , Cuatro Caminos ,
    Rios Rosas , Iglesia , Bilbao , Tribunal , Gran Via , Sol , Tirso
    de Molina , Anton Martin , Atocha , Atocha Renfe , Menendez
    Pelayo , Pacifico , Puente de Vallecas , Nueva Numancia ,
    Portazgo , Buenos Aires , Alto del Arenal , Miguel Hernandez ,
    Sierra de Guadalupe , Villa de Vallecas , Congosto
EDGE: label=2:color=255,34,0: adjlist=Cuatro Caminos , Canal ,
    Quevedo , San Bernardo , Noviciado , Santo Domingo , Opera , Sol ,
     Sevilla , Banco de Espana , Retiro , Principe de Vergara , Goya
    , Manuel Becerra , Ventas
EDGE: label=3:color=255,204,0: adjlist=Moncloa , Arguelles ,
    Ventura Rodriguez , Plaza de Espana , Callao , Sol , Lavapies ,
    Embajadores , Palos de la Frontera , Delicias , Legazpi

EDGE: label=4:color=102,0,0:adjlist=Arguelles, San Bernardo, Bilbao, Alonso Martinez, Colon, Serrano, Velazquez, Goya, Lista, Diego de Leon, Avda de America, Prosperidad, Alfonso XIII, Avda de la Paz, Arturo Soria, Esperanza, Canillas, Mar de Cristal, San Lorenzo, Parque de Santa Maria

EDGE: label=5:color=153,204,51:adjlist=Casa de Campo, Campamento, Empalme, Aluche, Eugenia de Montijo, Carabanchel, Vista Alegre, Oporto, Urgel, Marques de Vadillo, Piramides, Acacias, Puerto de Toledo, La Latina, Opera, Callao, Gran Via, Chueca, Alonso Martinez, Ruben Dario, Nunez de Balboa, Diego de Leon, Ventas, El Carmen, Quintana, Pueblo Nuevo, Ciudad Lineal, Suanzes, Torre Arias, Canillejas

EDGE: label=6:color=204,204,204:adjlist=Legazpi, Mendez Alvaro, Pacifico, Conde de Casal, Sainz de Baranda, O'Donnell, Manuel Becerra, Diego de Leon, Avda de America, Republica Argentina, Nuevos Ministerios, Cuatro Caminos, Guzman el Bueno, Metropolitano, Ciudad Universitaria, Moncloa, Arguelles, Principe Rio, Puerta del Angel, Alto de Extremadura, Lucero, Laguna, Carpetana, Oporto, Opanel, Plaza Eliptica, Usera, Legazpi

EDGE: label=7:color=255,153,0:adjlist=Pitis, Lacoma, Avda Ilustracion, Penagrande, Antonio Machado, Valdezarza, Francos Rodriguez, Guzman el Bueno, Islas Filipinas, Canal, Alonso Cano, Gregorio Maranon, Avda de America, Cartagena, Parque de las Avenidas, Barrio de la Concepcion, Pueblo Nuevo, Ascao, Garcia Noblejas, Simancas, San Blas, Las Musas

EDGE: label=8:color=255,102,153:adjlist=Barajas, Aeropuerto, Campo de las Naciones, Mar de Cristal, Colombia, Nuevos Ministerios

EDGE: label=9:color=153,51,102:adjlist=Herrera Oria, Barrio del Pilar, Ventilla, Plaza de Castilla, Duque de Pastrana, Pio XII, Colombia, Concha Espina, Cruz del Rayo, Avda de America, Nunez de Balboa, Principe de Vergara, Ibiza, Sainz de Baranda, Estrella, Vinateros, Artilleros, Pavones, Valdebernardo, Vicalvaro, San Cipriano, Puerta de Arganda, Rivas Urbanizaciones, Rivas Vaciamadrid, La Poveda, Arganda del Rey

EDGE: label=10:color=0,0,102:adjlist=Fuencarral, Begona, Chamartin, Plaza de Castilla, Cuzco, Santiago Bernabeu, Nuevos Ministerios, Gregorio Maranon, Alonso Martinez, Tribunal, Plaza de Espana, Principe Rio, Lago, Batan, Casa de Campo, Colonia Jardin, Cuatro Vientos, Joaquin Vilumbrales, Puerta del Sur

EDGE: label=11:color=0,102,0:adjlist=Plaza Eliptica, Abrantes, Pan Bendito

EDGE: label=12:color=221,204,51:adjlist=Puerta del Sur, San
    Nicasio, Leganes Central, Hospital Severo Ochoa, Casa del
    Reloj, Julian Besteiro, El Carrascal, El Bercial, Los
    Espartales, El Casar, Juan de la Cierva, Getafe Central,
    Alonso de Mendoza, Conservatorio, Arroyo Culebro, Parque de
    los Estados, Fuenlabrada Central, Parque Europa, Hospital de
     Fuenlabrada, Loranca, Manuela Malasana, Hospital de
    Mostoles, Pradillo, Mostoles Central, Universidad Rey Juan
    Carlos, Parque Oeste, Alcorcon Central, Parque Lisboa,
    Puerta del Sur
EDGE: label=R:color=0,0,0:adjlist=Principe Rio, Opera

B.6 Mexico City

NODE: label=General Anaya:x=690.1962319167424:y
    =56.058467001033705650.0
NODE: label=Oceania:x=940.3502369657688:y=432.3965591864863650.0
NODE: label=Terminal Arena:x=950.1078310769819:y
    =394.4713352638818650.0
NODE: label=La Paz:x=1134.7749604373348:y=77.1908618651961650.0
NODE: label=Hangares:x=965.4931385827108:y
    =359.4471071291963650.0
NODE: label=Pantitlan:x=985.9116976680533:y
    =333.53450508591675650.0
NODE: label=Agricola Oriental:x=1030.1440231447646:y
    =299.0024221352593650.0
NODE: label=De San Juan:x=1042.1091576036492:y
    =271.6547649235314650.0
NODE: label=Tepalcates:x=1053.5613631325073:y
    =238.01204706252685650.0
NODE: label=Guelatao:x=1058.617398590932:y
    =208.69890542341938650.0
NODE: label=Penon Viejo:x=1072.8480727447636:y
    =173.0661885673411650.0
NODE: label=Los Reyes:x=1101.2718600521146:y
    =95.86844759108476650.0
NODE: label=Santa Marta:x=1072.7428294364531:y
    =105.91539352650011650.0
NODE: label=Acatitla:x=1075.4690504525017:y
    =139.75240347255203650.0
NODE: label=Puebla:x=935.9100000952067:y=296.41524117850395650.0
NODE: label=Aragon:x=856.1857883404023:y=446.99974340006133650.0
NODE: label=Romero Rubio:x=896.00276758247:y
    =409.4949952355877650.0
NODE: label=Flores Magon:x=867.576285092002:y
    =397.18850897064794650.0
NODE: label=Eduardo Molina:x=803.3824889861783:y
    =448.3661175995914650.0

NODE: label=Consulado : x=766.1498229872799 : y
    =443.6433435239503650.0
NODE: label=Bondojito : x=766.286701799067 : y=476.317239866421650.0
NODE: label=Deportivo Oceania : x=964.3307797278712 : y
    =455.31243973907965650.0
NODE: label=Bosque de Aragon : x=977.9464793780767 : y
    =472.36642747282275650.0
NODE: label=Villa de Aragon : x=990.8547338633714 : y
    =499.59755706749706650.0
NODE: label=Nezahualcototl : x=1000.3390370617108 : y
    =516.5431117667663650.0
NODE: label=Impulsora : x=997.5489547543752 : y
    =532.5055236137931650.0
NODE: label=Rio de los Remedios : x=1000.5679026314724 : y
    =547.6603793744362650.0
NODE: label=Muzquaz : x=1012.4373866530839 : y
    =568.1026386905862650.0
NODE: label=Technologico : x=1018.1634064339185 : y
    =584.8630748762263650.0
NODE: label=Olimpica : x=1026.7100981035865 : y
    =600.8988873683857650.0
NODE: label=Plaza Aragon : x=1037.9690008190223 : y
    =616.0281610671931650.0
NODE: label=Ciudad Azteca : x=1048.6013714190424 : y
    =632.5728622370842650.0
NODE: label=Talisman : x=786.1707146025242 : y
    =529.7202357000715650.0
NODE: label=Martin Carrera : x=799.3689900204648 : y
    =557.5666209631888650.0
NODE: label=La Villa−Basilica : x=726.794946699478 : y
    =565.18047114417650.0
NODE: label=Deportivo 18 de Marzo : x=676.3000140909905 : y
    =562.7214192418963650.0
NODE: label=Indios Verdes : x=685.0369006188668 : y
    =602.005187664207650.0
NODE: label=Potrero : x=645.0759244923179 : y=510.7858646186287650.0
NODE: label=Lindavista : x=597.2037301692355 : y
    =563.3466532922939650.0
NODE: label=Instituto del Petroleo : x=538.870567962673 : y
    =549.3838747123286650.0
NODE: label=Politecnico : x=522.5316901468319 : y
    =588.7265267272081650.0
NODE: label=Autobuses del Norte : x=557.1875881750041 : y
    =505.815719457291650.0
NODE: label=La Raza : x=572.6710104117305 : y
    =469.11656371892883650.0
NODE: label=Valle Gomez : x=701.8599805101569 : y
    =444.81092939823577650.0

NODE: label=Misterios : x=620.7875947196756 : y
    =452.7747399281598650.0
NODE: label=Canal del Norte : x=768.6474959324212 : y
    =415.6103664793877650.0
NODE: label=Morelos : x=769.081043677851 : y=396.76246552434424650.0
NODE: label=San Lazaro : x=814.2613867314793 : y
    =377.2666308870394650.0
NODE: label=Tepito : x=693.5537232703792 : y=401.56728382917987650.0
NODE: label=Lagunilla : x=636.8105998909869 : y
    =405.4665739121127650.0
NODE: label=Garibaldi : x=591.822372666386 : y
    =405.74683230284495650.0
NODE: label=Buenavista : x=497.8501431313742 : y
    =435.03393195717206650.0
NODE: label=Cd. Deportiva : x=883.0400211413211 : y
    =282.83269659345365650.0
NODE: label=Velodromo : x=829.2218413264995 : y
    =281.4982099721949650.0
NODE: label=Moctezuma : x=845.9103909220333 : y
    =357.82039780462975650.0
NODE: label=Balbuena : x=867.6198362226827 : y
    =341.53453955404643650.0
NODE: label=Zaragoza : x=945.0509970872313 : y
    =327.97343967446295650.0
NODE: label=Gomez Farias : x=913.5477040393869 : y
    =325.91190884312874650.0
NODE: label=Bva. Pto. Aereo : x=889.9864815655966 : y
    =332.8650479882195650.0
NODE: label=Candelaria : x=767.1931856313887 : y
    =371.7051559696891650.0
NODE: label=Fray Servando : x=761.4143222420631 : y
    =334.0315697821942650.0
NODE: label=Constitucion de 1917 : x=995.317124089652 : y
    =103.50759487724679650.0
NODE: label=Cerro de la Estrella : x=934.5345267537435 : y
    =84.26737865933819650.0
NODE: label=Ztlapalapa : x=896.3013587483638 : y
    =92.9733942640031650.0
NODE: label=Uami : x=962.4489157019518 : y=89.95242994758814650.0
NODE: label=Atlalilco : x=844.9821996588755 : y
    =71.09925291026866650.0
NODE: label=Escuadron 201 : x=817.1791392759446 : y
    =93.30209490861102650.0
NODE: label=Aculco : x=802.2789713858667 : y=128.58126096837123650.0
NODE: label=Apatlaco : x=795.8640950191474 : y
    =164.54774867553732650.0
NODE: label=Iztacalco : x=787.7733206872942 : y
    =192.30247870791055650.0

NODE: label=Mixhuca:x=789.8854590135923:y=271.6835155959798650.0
NODE: label=Jamaica:x=755.099826984903:y=278.25417483640365650.0
NODE: label=Coyuya:x=778.3846713062158:y=224.17244477402062650.0
NODE: label=Santa Anita:x=743.2942348050627:y
    =244.7171345768217650.0
NODE: label=Merced:x=709.628699793079:y=356.76308505728025650.0
NODE: label=Zocalo:x=663.8286900662175:y=371.1162596233209650.0
NODE: label=Pino Suarez:x=667.1754622662816:y
    =349.4407369718181650.0
NODE: label=Allende:x=620.7857311906694:y
    =369.13331360426963650.0
NODE: label=Bellas Artes:x=593.2251170916165:y
    =369.64404832005664650.0
NODE: label=Guerrero:x=546.3165250417469:y
    =404.51332581103617650.0
NODE: label=Tlatelolco:x=554.371985778633:y
    =441.011505196711650.0
NODE: label=Hidalgo:x=539.7957163078484:y
    =368.73487977086387650.0
NODE: label=Revolucion:x=518.0148416837726:y
    =370.1878647871303650.0
NODE: label=San Cosme:x=500.27510457199486:y
    =376.3177768544568650.0
NODE: label=Normal:x=475.876086744038:y=388.19856087655234650.0
NODE: label=Colegio Militar:x=452.0647217099176:y
    =395.59284445512833650.0
NODE: label=Popotla:x=421.80428350307466:y
    =400.25656925213826650.0
NODE: label=San Juaquin:x=351.17623172926346:y
    =369.15191452798604650.0
NODE: label=Polanco:x=350.1598712868657:y=330.788738465861650.0
NODE: label=Auditorio:x=348.4935191761575:y
    =296.6589047433717650.0
NODE: label=Constituyentes:x=345.00989675046924:y
    =264.5319016691648650.0
NODE: label=Iacubata:x=362.27165066310636:y
    =234.2630601185437650.0
NODE: label=Observatorio:x=312.49682960498774:y
    =221.9559976086067650.0
NODE: label=San Pedro de los Pinos:x=386.3000633547856:y
    =202.9824460941909650.0
NODE: label=San Antonio:x=378.5614148764704:y
    =164.00463448704795650.0
NODE: label=Mixcoac:x=374.54330902320294:y
    =117.61667399358976650.0
NODE: label=Barranca del Muerto:x=371.24575025609306:y
    =78.37708640616893650.0

NODE: label=Panteones : x=294.1375166095565 : y
=409.1378910279736650.0
NODE: label=Cuatro Caminos : x=225.2047277950776 : y
=393.7659306549358650.0
NODE: label=Cuitlahuac : x=393.7459990347896 : y
=403.9780261244692650.0
NODE: label=Tacuba : x=356.1203908410537 : y=431.4519811269679650.0
NODE: label=Vallejo : x=499.7944848023392 : y=533.9620327558863650.0
NODE: label=Norte 49 : x=463.1077184560619 : y=549.4430267278866650.0
NODE: label=Refineria : x=342.1713510927749 : y
=471.3770203365727650.0
NODE: label=Camarones : x=290.55188126805604 : y
=520.3324091986826650.0
NODE: label=Aquiles Serdan : x=233.12755368763635 : y
=547.161494580002650.0
NODE: label=El Rosario : x=226.67979018388434 : y
=595.6426804223457650.0
NODE: label=Yezozomoc : x=259.2724909545603 : y
=573.1813199311617650.0
NODE: label=Azcapotzalco : x=328.08898769817085 : y
=536.5832172860836650.0
NODE: label=Ferreria : x=397.082899157643 : y=524.6010084257157650.0
NODE: label=Juarez : x=537.1363304844184 : y=341.13523998073407650.0
NODE: label=Balderas : x=536.1895262493688 : y
=314.98379999974804650.0
NODE: label=Cuauhtemoc : x=505.17880903446894 : y
=306.4869065943534650.0
NODE: label=Juanacatlan : x=387.6205083556217 : y
=254.32469406288715650.0
NODE: label=Chapultepec : x=418.37585716013166 : y
=273.50414814684046650.0
NODE: label=Sevilla : x=440.10370787897125 : y
=285.24590390586474650.0
NODE: label=Insurgentes : x=477.0903725732559 : y
=296.0301611890389650.0
NODE: label=San Juan de Letran : x=593.6395399051573 : y
=350.27393666504287650.0
NODE: label=Salto del Agua : x=594.6873459929035 : y
=329.5881663159852650.0
NODE: label=Isabel la Catolica : x=630.4104238219322 : y
=335.45305521842295650.0
NODE: label=La Viga : x=705.5634498860911 : y=257.1328964931833650.0
NODE: label=Chabacano : x=669.25797058584 : y=284.8940835264632650.0
NODE: label=Lazaro Cardenas : x=643.3103155898646 : y
=264.1057034307296650.0
NODE: label=Viaducto : x=671.6794646798804 : y
=245.7358702841007650.0

NODE: label=San Antonio Abad:x=671.6371108019874:y
    =316.16630389679026 50.0
NODE: label=Xola:x=672.5374586206407:y=210.63743956441567650.0
NODE: label=Villa de Cortes:x=672.8004071347966:y
    =176.42760037240555650.0
NODE: label=Iavitas:x=674.4701256654256:y=144.93292158506296 50.0
NODE: label=Ermita:x=685.9422490688188:y=82.87451060439832650.0
NODE: label=Portales:x=678.6380122545044:y
    =112.22582060471518650.0
NODE: label=Tasquena:x=718.2808686238508:y
    =40.97521210647949650.0
NODE: label=Universidad:x=555.6227268668898:y
    =36.85899290924158650.0
NODE: label=Copilco:x=517.5446840628206:y=47.44939640780285650.0
NODE: label=Miguel A de Quevedo:x=504.2832216209025:y
    =66.27853747378867650.0
NODE: label=Viveros:x=504.36684378218297:y
    =84.54052759592457650.0
NODE: label=Coyoacan:x=505.7737474340197:y
    =108.09908735696342650.0
NODE: label=Patriotismo:x=420.49738751095197:y
    =238.28029987372554650.0
NODE: label=Chilpancingo:x=463.84803694469474:y
    =238.117746223256650.0
NODE: label=Centro Medico:x=538.0754246420652:y
    =244.13860755396337650.0
NODE: label=Obrera:x=635.660624991185:y=296.26724806361955650.0
NODE: label=Doctores:x=612.4215364035742:y
    =304.1274804688043650.0
NODE: label=Ninos Heroes:x=536.2166732386614:y
    =287.0520348178272650.0
NODE: label=Hospital General:x=538.7943380318693:y
    =266.4177784518864650.0
NODE: label=Etiopia:x=535.917635924966:y=215.13388735905932650.0
NODE: label=Eugenia:x=533.9280961771702:y
    =188.79007070050915650.0
NODE: label=Division del Norte:x=516.8483092535726:y
    =157.42025508481606650.0
NODE: label=Zapata:x=505.20301067426135:y
    =133.59339306069205650.0
EDGE: label=1:color=255,59,255:adjlist=Observatorio , Iacubata
EDGE: label=1:color=255,59,255:adjlist=Iacubata , Juanacatlan
EDGE: label=1:color=255,59,255:adjlist=Juanacatlan , Chapultepec
EDGE: label=1:color=255,59,255:adjlist=Chapultepec , Sevilla
EDGE: label=1:color=255,59,255:adjlist=Sevilla , Insurgentes
EDGE: label=1:color=255,59,255:adjlist=Insurgentes , Cuauhtemoc
EDGE: label=1:color=255,59,255:adjlist=Cuauhtemoc , Balderas
EDGE: label=1:color=255,59,255:adjlist=Balderas , Salto del Agua

EDGE: label=1: color=255,59,255: adjlist=Salto del Agua, Isabel la
    Catolica
EDGE: label=1: color=255,59,255: adjlist=Isabel la Catolica, Pino
    Suarez
EDGE: label=1: color=255,59,255: adjlist=Pino Suarez, Merced
EDGE: label=1: color=255,59,255: adjlist=Merced, Candelaria
EDGE: label=1: color=255,59,255: adjlist=Candelaria, San Lazaro
EDGE: label=1: color=255,59,255: adjlist=San Lazaro, Moctezuma
EDGE: label=1: color=255,59,255: adjlist=Moctezuma, Balbuena
EDGE: label=1: color=255,59,255: adjlist=Balbuena, Bva. Pto. Aereo
EDGE: label=1: color=255,59,255: adjlist=Bva. Pto. Aereo, Gomez
    Farias
EDGE: label=1: color=255,59,255: adjlist=Gomez Farias, Zaragoza
EDGE: label=1: color=255,59,255: adjlist=Zaragoza, Pantitlan
EDGE: label=2: color=27,1,229: adjlist=Cuatro Caminos, Panteones
EDGE: label=2: color=27,1,229: adjlist=Panteones, Tacuba
EDGE: label=2: color=27,1,229: adjlist=Tacuba, Cuitlahuac
EDGE: label=2: color=27,1,229: adjlist=Cuitlahuac, Popotla
EDGE: label=2: color=27,1,229: adjlist=Popotla, Colegio Militar
EDGE: label=2: color=27,1,229: adjlist=Colegio Militar, Normal
EDGE: label=2: color=27,1,229: adjlist=Normal, San Cosme
EDGE: label=2: color=27,1,229: adjlist=San Cosme, Revolucion
EDGE: label=2: color=27,1,229: adjlist=Revolucion, Hidalgo
EDGE: label=2: color=27,1,229: adjlist=Hidalgo, Bellas Artes
EDGE: label=2: color=27,1,229: adjlist=Bellas Artes, Allende
EDGE: label=2: color=27,1,229: adjlist=Allende, Zocalo
EDGE: label=2: color=27,1,229: adjlist=Zocalo, Pino Suarez
EDGE: label=2: color=27,1,229: adjlist=Pino Suarez, San Antonio
    Abad
EDGE: label=2: color=27,1,229: adjlist=San Antonio Abad, Chabacano
EDGE: label=2: color=27,1,229: adjlist=Chabacano, Viaducto
EDGE: label=2: color=27,1,229: adjlist=Viaducto, Xola
EDGE: label=2: color=27,1,229: adjlist=Xola, Villa de Cortes
EDGE: label=2: color=27,1,229: adjlist=Villa de Cortes, Iavitas
EDGE: label=2: color=27,1,229: adjlist=Iavitas, Portales
EDGE: label=2: color=27,1,229: adjlist=Portales, Ermita
EDGE: label=2: color=27,1,229: adjlist=Ermita, General Anaya
EDGE: label=2: color=27,1,229: adjlist=General Anaya, Tasquena
EDGE: label=3: color=131,158,6: adjlist=Universidad, Copilco
EDGE: label=3: color=131,158,6: adjlist=Copilco, Miguel A de
    Quevedo
EDGE: label=3: color=131,158,6: adjlist=Miguel A de Quevedo,
    Viveros
EDGE: label=3: color=131,158,6: adjlist=Viveros, Coyoacan
EDGE: label=3: color=131,158,6: adjlist=Coyoacan, Zapata
EDGE: label=3: color=131,158,6: adjlist=Zapata, Division del Norte
EDGE: label=3: color=131,158,6: adjlist=Division del Norte,
    Eugenia

EDGE: label=3:color=131,158,6:adjlist=Eugenia, Etiopia
EDGE: label=3:color=131,158,6:adjlist=Etiopia, Centro Medico
EDGE: label=3:color=131,158,6:adjlist=Centro Medico, Hospital
    General
EDGE: label=3:color=131,158,6:adjlist=Hospital General, Ninos
    Heroes
EDGE: label=3:color=131,158,6:adjlist=Ninos Heroes, Balderas
EDGE: label=3:color=131,158,6:adjlist=Balderas, Juarez
EDGE: label=3:color=131,158,6:adjlist=Juarez, Hidalgo
EDGE: label=3:color=131,158,6:adjlist=Hidalgo, Guerrero
EDGE: label=3:color=131,158,6:adjlist=Guerrero, Tlatelolco
EDGE: label=3:color=131,158,6:adjlist=Tlatelolco, La Raza
EDGE: label=3:color=131,158,6:adjlist=La Raza, Potrero
EDGE: label=3:color=131,158,6:adjlist=Potrero, Deportivo 18 de
    Marzo
EDGE: label=3:color=131,158,6:adjlist=Deportivo 18 de Marzo,
    Indios Verdes
EDGE: label=4:color=120,249,113:adjlist=Santa Anita, Jamaica
EDGE: label=4:color=120,249,113:adjlist=Jamaica, Fray Servando
EDGE: label=4:color=120,249,113:adjlist=Fray Servando,
    Candelaria
EDGE: label=4:color=120,249,113:adjlist=Candelaria, Morelos
EDGE: label=4:color=120,249,113:adjlist=Morelos, Canal del Norte
EDGE: label=4:color=120,249,113:adjlist=Canal del Norte,
    Consulado
EDGE: label=4:color=120,249,113:adjlist=Consulado, Bondojito
EDGE: label=4:color=120,249,113:adjlist=Bondojito, Talisman
EDGE: label=4:color=120,249,113:adjlist=Talisman, Martin Carrera
EDGE: label=5:color=255,255,0:adjlist=Politecnico, Instituto del
    Petroleo
EDGE: label=5:color=255,255,0:adjlist=Instituto del Petroleo,
    Autobuses del Norte
EDGE: label=5:color=255,255,0:adjlist=Autobuses del Norte, La
    Raza
EDGE: label=5:color=255,255,0:adjlist=La Raza, Misterios
EDGE: label=5:color=255,255,0:adjlist=Misterios, Valle Gomez
EDGE: label=5:color=255,255,0:adjlist=Valle Gomez, Consulado
EDGE: label=5:color=255,255,0:adjlist=Consulado, Eduardo Molina
EDGE: label=5:color=255,255,0:adjlist=Eduardo Molina, Aragon
EDGE: label=5:color=255,255,0:adjlist=Aragon, Oceania
EDGE: label=5:color=255,255,0:adjlist=Oceania, Terminal Arena
EDGE: label=5:color=255,255,0:adjlist=Terminal Arena, Hangares
EDGE: label=5:color=255,255,0:adjlist=Hangares, Pantitlan
EDGE: label=6:color=255,0,0:adjlist=El Rosario, Yezozomoc
EDGE: label=6:color=255,0,0:adjlist=Yezozomoc, Azcapotzalco
EDGE: label=6:color=255,0,0:adjlist=Azcapotzalco, Ferreria
EDGE: label=6:color=255,0,0:adjlist=Ferreria, Norte 49
EDGE: label=6:color=255,0,0:adjlist=Norte 49, Vallejo

EDGE: label=6:color=255,0,0: adjlist=Vallejo , Instituto del
    Petroleo
EDGE: label=6:color=255,0,0: adjlist=Instituto del Petroleo ,
    Lindavista
EDGE: label=6:color=255,0,0: adjlist=Lindavista , Deportivo 18 de
    Marzo
EDGE: label=6:color=255,0,0: adjlist=Deportivo 18 de Marzo, La
    Villa−Basilica
EDGE: label=6:color=255,0,0: adjlist=La Villa−Basilica , Martin
    Carrera
EDGE: label=7:color=246,122,0: adjlist=Barranca del Muerto ,
    Mixcoac
EDGE: label=7:color=246,122,0: adjlist=Mixcoac , San Antonio
EDGE: label=7:color=246,122,0: adjlist=San Antonio , San Pedro de
    los Pinos
EDGE: label=7:color=246,122,0: adjlist=San Pedro de los Pinos ,
    Iacubata
EDGE: label=7:color=246,122,0: adjlist=Iacubata , Constituyentes
EDGE: label=7:color=246,122,0: adjlist=Constituyentes , Auditorio
EDGE: label=7:color=246,122,0: adjlist=Auditorio , Polanco
EDGE: label=7:color=246,122,0: adjlist=Polanco , San Juaquin
EDGE: label=7:color=246,122,0: adjlist=San Juaquin , Tacuba
EDGE: label=7:color=246,122,0: adjlist=Tacuba , Refineria
EDGE: label=7:color=246,122,0: adjlist=Refineria , Camarones
EDGE: label=7:color=246,122,0: adjlist=Camarones , Aquiles Serdan
EDGE: label=7:color=246,122,0: adjlist=Aquiles Serdan , El Rosario
EDGE: label=8:color=6,158,17: adjlist=Garibaldi , Bellas Artes
EDGE: label=8:color=6,158,17: adjlist=Bellas Artes , San Juan de
    Letran
EDGE: label=8:color=6,158,17: adjlist=San Juan de Letran , Salto
    del Agua
EDGE: label=8:color=6,158,17: adjlist=Salto del Agua, Doctores
EDGE: label=8:color=6,158,17: adjlist=Doctores , Obrera
EDGE: label=8:color=6,158,17: adjlist=Obrera , Chabacano
EDGE: label=8:color=6,158,17: adjlist=Chabacano , La Viga
EDGE: label=8:color=6,158,17: adjlist=La Viga , Santa Anita
EDGE: label=8:color=6,158,17: adjlist=Santa Anita , Coyuya
EDGE: label=8:color=6,158,17: adjlist=Coyuya , Iztacalco
EDGE: label=8:color=6,158,17: adjlist=Iztacalco , Apatlaco
EDGE: label=8:color=6,158,17: adjlist=Apatlaco , Aculco
EDGE: label=8:color=6,158,17: adjlist=Aculco , Escuadron 201
EDGE: label=8:color=6,158,17: adjlist=Escuadron 201 , Atlalilco
EDGE: label=8:color=6,158,17: adjlist=Atlalilco , Ztlapalapa
EDGE: label=8:color=6,158,17: adjlist=Ztlapalapa , Cerro de la
    Estrella
EDGE: label=8:color=6,158,17: adjlist=Cerro de la Estrella , Uami
EDGE: label=8:color=6,158,17: adjlist=Uami, Constitucion de 1917
EDGE: label=9:color=106,66,10: adjlist=Iacubata , Patriotismo

EDGE: label=9:color=106,66,10:adjlist=Patriotismo, Chilpancingo
EDGE: label=9:color=106,66,10:adjlist=Chilpancingo, Centro
    Medico
EDGE: label=9:color=106,66,10:adjlist=Centro Medico, Lazaro
    Cardenas
EDGE: label=9:color=106,66,10:adjlist=Lazaro Cardenas, Chabacano
EDGE: label=9:color=106,66,10:adjlist=Chabacano, Jamaica
EDGE: label=9:color=106,66,10:adjlist=Jamaica, Mixhuca
EDGE: label=9:color=106,66,10:adjlist=Mixhuca, Velodromo
EDGE: label=9:color=106,66,10:adjlist=Velodromo, Cd. Deportiva
EDGE: label=9:color=106,66,10:adjlist=Cd. Deportiva, Puebla
EDGE: label=9:color=106,66,10:adjlist=Puebla, Pantitlan
EDGE: label=A:color=158,6,81:adjlist=Pantitlan, Agricola
    Oriental
EDGE: label=A:color=158,6,81:adjlist=Agricola Oriental, De San
    Juan
EDGE: label=A:color=158,6,81:adjlist=De San Juan, Tepalcates
EDGE: label=A:color=158,6,81:adjlist=Tepalcates, Guelatao
EDGE: label=A:color=158,6,81:adjlist=Guelatao, Penon Viejo
EDGE: label=A:color=158,6,81:adjlist=Penon Viejo, Acatitla
EDGE: label=A:color=158,6,81:adjlist=Acatitla, Santa Marta
EDGE: label=A:color=158,6,81:adjlist=Santa Marta, Los Reyes
EDGE: label=A:color=158,6,81:adjlist=Los Reyes, La Paz
EDGE: label=B:color=200,200,200:adjlist=Buenavista, Guerrero
EDGE: label=B:color=200,200,200:adjlist=Guerrero, Garibaldi
EDGE: label=B:color=200,200,200:adjlist=Garibaldi, Lagunilla
EDGE: label=B:color=200,200,200:adjlist=Lagunilla, Tepito
EDGE: label=B:color=200,200,200:adjlist=Tepito, Morelos
EDGE: label=B:color=200,200,200:adjlist=Morelos, San Lazaro
EDGE: label=B:color=200,200,200:adjlist=San Lazaro, Flores Magon
EDGE: label=B:color=200,200,200:adjlist=Flores Magon, Romero
    Rubio
EDGE: label=B:color=200,200,200:adjlist=Romero Rubio, Oceania
EDGE: label=B:color=200,200,200:adjlist=Oceania, Deportivo
    Oceania
EDGE: label=B:color=200,200,200:adjlist=Deportivo Oceania,
    Bosque de Aragon
EDGE: label=B:color=200,200,200:adjlist=Bosque de Aragon, Villa
    de Aragon
EDGE: label=B:color=200,200,200:adjlist=Villa de Aragon,
    Nezahualcototl
EDGE: label=B:color=200,200,200:adjlist=Nezahualcototl,
    Impulsora
EDGE: label=B:color=200,200,200:adjlist=Impulsora, Rio de los
    Remedios
EDGE: label=B:color=200,200,200:adjlist=Rio de los Remedios,
    Muzquaz
EDGE: label=B:color=200,200,200:adjlist=Muzquaz, Technologico

EDGE: label=B: color =200 ,200 ,200: adjlist=Technologico , Olimpica
EDGE: label=B: color =200 ,200 ,200: adjlist=Olimpica , Plaza Aragon
EDGE: label=B: color =200 ,200 ,200: adjlist=Plaza Aragon , Ciudad
    Azteca

B.7 Recife

NODE: label=Recife : x=1201.0644337180665 : y=431.6944884725592715.0
NODE: label=Joana Bezerra : x=1135.9077461228494 : y
    =410.45874565853364715.0
NODE: label=Largo Da Paz : x=1099.7128850411473 : y
    =387.49878969858446715.0
NODE: label=Imbiribeira : x=1084.715070045847 : y
    =344.84736795623326715.0
NODE: label=Boa Viagem : x=1091.838271925862 : y
    =290.8563291108914715.0
NODE: label=Shopping : x=1086.8402682538417 : y
    =250.46964425608172715.0
NODE: label=Tancredo Neves : x=1082.2739976291705 : y
    =196.98018228614274715.0
NODE: label=Aeroporto : x=1073.0332819117302 : y
    =152.08367187454576715.0
NODE: label=Porta Larga : x=1060.8420362877125 : y
    =117.52062179791744715.0
NODE: label=Monte Dos Guararpes : x=1028.352433423444 : y
    =75.90633408124972715.0
NODE: label=Prazeres : x=983.0696591851895 : y
    =42.891302243654536715.0
NODE: label=Cajueiro Seco : x=945.9069868898079 : y
    =9.33002148337971715.0
NODE: label=Afogadas : x=1091.499674549303 : y
    =407.7895818241699715.0
NODE: label=Werneck : x=923.8783845210328 : y
    =389.28471126622486715.0
NODE: label=Ipiranga : x=1059.3719861694362 : y
    =405.6054874357932715.0
NODE: label=Santa Luzia : x=972.2562314094442 : y
    =394.62277233728537715.0
NODE: label=Mangeuira : x=1022.0775620267698 : y
    =398.3939739549012715.0
NODE: label=Barro : x=869.5541634144201 : y=385.5416453993967715.0
NODE: label=Tejipio : x=785.998555970139 : y=364.23382066360034715.0
NODE: label=Coqueiral : x=721.8848031982609 : y
    =362.64772465616414715.0
NODE: label=Alto Do C u : x=683.8844470722377 : y
    =392.6426644274835715.0
NODE: label=Curado : x=667.8353214211054 : y=450.8290238135853715.0

NODE: label=R o d o v i ria : x=656.4145203444778:y
    =505.135489117506715.0
NODE: label=Cavaleiro : x=683.699967753158:y
    =338.25259478495923715.0
NODE: label=Floriano : x=591.8788373675591:y
    =273.3332173663129715.0
NODE: label=Engenho Velho : x=541.2395964624005:y
    =264.6159032341998715.0
NODE: label=Jaboatao : x=487.1167013355696:y
    =256.50345863537973715.0
NODE: label=Timbi : x=563.1661821388152:y=699.40619713756715.0
EDGE: label=Red Line : color =255,0,0: adjlist=Recife , Joana Bezerra
EDGE: label=Red Line : color =255,0,0: adjlist=Joana Bezerra ,
    Afogadas
EDGE: label=Red Line : color =255,0,0: adjlist=Afogadas , Ipiranga
EDGE: label=Red Line : color =255,0,0: adjlist=Ipiranga , Mangeuira
EDGE: label=Red Line : color =255,0,0: adjlist=Mangeuira , Santa
    Luzia
EDGE: label=Red Line : color =255,0,0: adjlist=Santa Luzia , Werneck
EDGE: label=Red Line : color =255,0,0: adjlist=Werneck , Barro
EDGE: label=Red Line : color =255,0,0: adjlist=Barro , Tejipio
EDGE: label=Red Line : color =255,0,0: adjlist=Tejipio , Coqueiral
EDGE: label=Red Line : color =255,0,0: adjlist=Coqueiral , Cavaleiro
EDGE: label=Red Line : color =255,0,0: adjlist=Cavaleiro , Floriano
EDGE: label=Red Line : color =255,0,0: adjlist=Floriano , Engenho
    Velho
EDGE: label=Red Line : color =255,0,0: adjlist=Engenho Velho ,
    Jaboatao
EDGE: label=Red Line : color =255,0,0: adjlist=Coqueiral , Alto Do
    C u
EDGE: label=Red Line : color =255,0,0: adjlist=Alto Do C u , Curado
EDGE: label=Red Line : color =255,0,0: adjlist=Curado , R o d o v i ria
EDGE: label=Red Line : color =255,0,0: adjlist=R o d o v i ria , Timbi
EDGE: label=Blue Line : color =0,0,255: adjlist=Recife , Joana
    Bezerra
EDGE: label=Blue Line : color =0,0,255: adjlist=Joana Bezerra , Largo
    Da Paz
EDGE: label=Blue Line : color =0,0,255: adjlist=Largo Da Paz ,
    Imbiribeira
EDGE: label=Blue Line : color =0,0,255: adjlist=Imbiribeira , Boa
    Viagem
EDGE: label=Blue Line : color =0,0,255: adjlist=Boa Viagem , Shopping
EDGE: label=Blue Line : color =0,0,255: adjlist=Shopping , Tancredo
    Neves
EDGE: label=Blue Line : color =0,0,255: adjlist=Tancredo Neves ,
    Aeroporto
EDGE: label=Blue Line : color =0,0,255: adjlist=Aeroporto , Porta
    Larga

EDGE:label=Blue Line:color=0,0,255:adjlist=Porta Larga, Monte
    Dos Guararpes
EDGE:label=Blue Line:color=0,0,255:adjlist=Monte Dos Guararpes,
     Prazeres
EDGE:label=Blue Line:color=0,0,255:adjlist=Prazeres, Cajueiro
    Seco

B.8 San Francisco

NODE:label=Oakland City Center/12th St:x=−122.2715:y=37.8031
NODE:label=19th St/Oakland:x=−122.2687:y=37.8075
NODE:label=MacArthur:x=−122.2673:y=37.8284
NODE:label=Ashby:x=−122.2698:y=37.8530
NODE:label=Downtown Berkeley:x=−122.2680:y=37.8699
NODE:label=North Berkeley:x=−122.2839:y=37.8739
NODE:label=El Cerrito Plaza:x=−122.2992:y=37.9028
NODE:label=El Cerrito del Norte:x=−122.3176:y=37.9258
NODE:label=Richmond:x=−122.3538:y=37.9373
NODE:label=Rockridge:x=−122.2527:y=37.8442
NODE:label=Orinda:x=−122.1833:y=37.8785
NODE:label=Lafayette:x=−122.1236:y=37.8934
NODE:label=Walnut Creek:x=−122.0681:y=37.9046
NODE:label=Pleansant Hill:x=−122.0567:y=37.9276
NODE:label=Concord:x=−122.0297:y=37.9721
NODE:label=Morth Concord/Martinez:x=−122.0251:y=38.0027
NODE:label=Pittsburgh/Bay Point:x=−121.9418:y=38.0190
NODE:label=Lake Merritt:x=−122.2655:y=37.7975
NODE:label=West Oakland:x=−122.2944:y=37.8047
NODE:label=Embarcadero:x=−122.3970:y=37.7929
NODE:label=Fruitvale:x=−122.2241:y=37.7746
NODE:label=Fremont:x=−121.9763:y=37.5572
NODE:label=Dublin/Pleasanton:x=−121.9004:y=37.7015
NODE:label=Union City:x=−122.0178:y=37.5913
NODE:label=South Hayward:x=−122.0578:y=37.6350
NODE:label=Hayward:x=−122.0878:y=37.6705
NODE:label=Bay Fair:x=−122.1278:y=37.6978
NODE:label=San Leandro:x=−122.1613:y=37.7228
NODE:label=Coliseum/Oakland Airport:x=−122.1980:y=37.7543
NODE:label=Castro Valley:x=−122.0776:y=37.6909
NODE:label=Montgomery St:x=−122.4020:y=37.7892
NODE:label=Powell St:x=−122.4068:y=37.7850
NODE:label=Civic Center:x=−122.4136:y=37.7795
NODE:label=16th St Mission:x=−122.4197:y=37.7650
NODE:label=24th St Mission:x=−122.4183:y=37.7523
NODE:label=Millbrae:x=−122.3868:y=37.6002
NODE:label=San Francisco International Airport (SFO):x
    =−122.3925:y=37.6157
NODE:label=San Bruno:x=−122.4157:y=37.6371

NODE: label=Glen Park:x=−122.4339:y=37.7329
NODE: label=Balboa Park:x=−122.4475:y=37.7217
NODE: label=Daly City:x=−122.4690:y=37.7062
NODE: label=Colma:x=−122.4676:y=37.6847
NODE: label=South San Francisco:x=−122.4439:y=37.6642
EDGE: label=BART Richmond−Daly City:color=255,0,0:adjlist=
    Richmond, El Cerrito del Norte
EDGE: label=BART Richmond−Daly City:color=255,0,0:adjlist=El
    Cerrito del Norte, El Cerrito Plaza
EDGE: label=BART Richmond−Daly City:color=255,0,0:adjlist=El
    Cerrito Plaza, North Berkeley
EDGE: label=BART Richmond−Daly City:color=255,0,0:adjlist=North
    Berkeley, Downtown Berkeley
EDGE: label=BART Richmond−Daly City:color=255,0,0:adjlist=
    Downtown Berkeley, Ashby
EDGE: label=BART Richmond−Daly City:color=255,0,0:adjlist=Ashby,
     MacArthur
EDGE: label=BART Richmond−Daly City:color=255,0,0:adjlist=
    MacArthur, 19th St/Oakland
EDGE: label=BART Richmond−Daly City:color=255,0,0:adjlist=19th
    St/Oakland, Oakland City Center/12th St
EDGE: label=BART Richmond−Daly City:color=255,0,0:adjlist=
    Oakland City Center/12th St, West Oakland
EDGE: label=BART Richmond−Daly City:color=255,0,0:adjlist=West
    Oakland, Embarcadero
EDGE: label=BART Richmond−Daly City:color=255,0,0:adjlist=
    Embarcadero, Montgomery St
EDGE: label=BART Richmond−Daly City:color=255,0,0:adjlist=
    Montgomery St, Powell St
EDGE: label=BART Richmond−Daly City:color=255,0,0:adjlist=Powell
     St, Civic Center
EDGE: label=BART Richmond−Daly City:color=255,0,0:adjlist=Civic
    Center, 16th St Mission
EDGE: label=BART Richmond−Daly City:color=255,0,0:adjlist=16th
    St Mission, 24th St Mission
EDGE: label=BART Richmond−Daly City:color=255,0,0:adjlist=24th
    St Mission, Glen Park
EDGE: label=BART Richmond−Daly City:color=255,0,0:adjlist=Glen
    Park, Balboa Park
EDGE: label=BART Richmond−Daly City:color=255,0,0:adjlist=Balboa
     Park, Daly City
EDGE: label=BART Fremont−Daly City:color=0,123,49:adjlist=
    Fremont, Union City
EDGE: label=BART Fremont−Daly City:color=0,123,49:adjlist=Union
    City, South Hayward
EDGE: label=BART Fremont−Daly City:color=0,123,49:adjlist=South
    Hayward, Hayward

EDGE: label=BART Fremont−Daly City : color =0,123,49: adjlist=
    Hayward, Bay Fair
EDGE: label=BART Fremont−Daly City : color =0,123,49: adjlist=Bay
    Fair, San Leandro
EDGE: label=BART Fremont−Daly City : color =0,123,49: adjlist=San
    Leandro, Coliseum/Oakland Airport
EDGE: label=BART Fremont−Daly City : color =0,123,49: adjlist=
    Coliseum/Oakland Airport, Fruitvale
EDGE: label=BART Fremont−Daly City : color =0,123,49: adjlist=
    Fruitvale, Lake Merritt
EDGE: label=BART Fremont−Daly City : color =0,123,49: adjlist=Lake
    Merritt, West Oakland
EDGE: label=BART Fremont−Daly City : color =0,123,49: adjlist=West
    Oakland, Embarcadero
EDGE: label=BART Fremont−Daly City : color =0,123,49: adjlist=
    Embarcadero, Montgomery St
EDGE: label=BART Fremont−Daly City : color =0,123,49: adjlist=
    Montgomery St, Powell St
EDGE: label=BART Fremont−Daly City : color =0,123,49: adjlist=Powell
    St, Civic Center
EDGE: label=BART Fremont−Daly City : color =0,123,49: adjlist=Civic
    Center, 16th St Mission
EDGE: label=BART Fremont−Daly City : color =0,123,49: adjlist=16th
    St Mission, 24th St Mission
EDGE: label=BART Fremont−Daly City : color =0,123,49: adjlist=24th
    St Mission, Glen Park
EDGE: label=BART Fremont−Daly City : color =0,123,49: adjlist=Glen
    Park, Balboa Park
EDGE: label=BART Fremont−Daly City : color =0,123,49: adjlist=Balboa
    Park, Daly City
EDGE: label=BART Richmond−Fremont : color =255,200,0: adjlist=
    Richmond, El Cerrito del Norte
EDGE: label=BART Richmond−Fremont : color =255,200,0: adjlist=El
    Cerrito del Norte, El Cerrito Plaza
EDGE: label=BART Richmond−Fremont : color =255,200,0: adjlist=El
    Cerrito Plaza, North Berkeley
EDGE: label=BART Richmond−Fremont : color =255,200,0: adjlist=North
    Berkeley, Downtown Berkeley
EDGE: label=BART Richmond−Fremont : color =255,200,0: adjlist=
    Downtown Berkeley, Ashby
EDGE: label=BART Richmond−Fremont : color =255,200,0: adjlist=Ashby,
    MacArthur
EDGE: label=BART Richmond−Fremont : color =255,200,0: adjlist=
    MacArthur, 19th St/Oakland
EDGE: label=BART Richmond−Fremont : color =255,200,0: adjlist=19th
    St/Oakland, Oakland City Center/12th St
EDGE: label=BART Richmond−Fremont : color =255,200,0: adjlist=
    Oakland City Center/12th St, Lake Merritt

EDGE: label=BART Richmond−Fremont: color =255,200,0: adjlist=Lake
    Merritt , Fruitvale
EDGE: label=BART Richmond−Fremont: color =255,200,0: adjlist=
    Fruitvale , Coliseum/Oakland Airport
EDGE: label=BART Richmond−Fremont: color =255,200,0: adjlist=
    Coliseum/Oakland Airport , San Leandro
EDGE: label=BART Richmond−Fremont: color =255,200,0: adjlist=San
    Leandro , Bay Fair
EDGE: label=BART Richmond−Fremont: color =255,200,0: adjlist=Bay
    Fair , Hayward
EDGE: label=BART Richmond−Fremont: color =255,200,0: adjlist=
    Hayward , South Hayward
EDGE: label=BART Richmond−Fremont: color =255,200,0: adjlist=South
    Hayward , Union City
EDGE: label=BART Richmond−Fremont: color =255,200,0: adjlist=Union
    City , Fremont
EDGE: label=BART Pittsburgh/Bay Point−Millbrae : color =255,255,0:
    adjlist=Pittsburgh/Bay Point , Morth Concord/Martinez
EDGE: label=BART Pittsburgh/Bay Point−Millbrae : color =255,255,0:
    adjlist=Morth Concord/Martinez , Concord
EDGE: label=BART Pittsburgh/Bay Point−Millbrae : color =255,255,0:
    adjlist=Concord, Pleansant Hill
EDGE: label=BART Pittsburgh/Bay Point−Millbrae : color =255,255,0:
    adjlist=Pleansant Hill , Walnut Creek
EDGE: label=BART Pittsburgh/Bay Point−Millbrae : color =255,255,0:
    adjlist=Walnut Creek , Lafayette
EDGE: label=BART Pittsburgh/Bay Point−Millbrae : color =255,255,0:
    adjlist=Lafayette , Orinda
EDGE: label=BART Pittsburgh/Bay Point−Millbrae : color =255,255,0:
    adjlist=Orinda , Rockridge
EDGE: label=BART Pittsburgh/Bay Point−Millbrae : color =255,255,0:
    adjlist=Rockridge , MacArthur
EDGE: label=BART Pittsburgh/Bay Point−Millbrae : color =255,255,0:
    adjlist=MacArthur , 19th St/Oakland
EDGE: label=BART Pittsburgh/Bay Point−Millbrae : color =255,255,0:
    adjlist=19th St/Oakland , Oakland City Center/12th St
EDGE: label=BART Pittsburgh/Bay Point−Millbrae : color =255,255,0:
    adjlist=Oakland City Center/12th St , West Oakland
EDGE: label=BART Pittsburgh/Bay Point−Millbrae : color =255,255,0:
    adjlist=West Oakland , Embarcadero
EDGE: label=BART Pittsburgh/Bay Point−Millbrae : color =255,255,0:
    adjlist=Embarcadero , Montgomery St
EDGE: label=BART Pittsburgh/Bay Point−Millbrae : color =255,255,0:
    adjlist=Montgomery St , Powell St
EDGE: label=BART Pittsburgh/Bay Point−Millbrae : color =255,255,0:
    adjlist=Powell St , Civic Center
EDGE: label=BART Pittsburgh/Bay Point−Millbrae : color =255,255,0:
    adjlist=Civic Center , 16th St Mission

EDGE: label=BART Pittsburgh/Bay Point−Millbrae: color=255,255,0:
    adjlist=16th St Mission, 24th St Mission
EDGE: label=BART Pittsburgh/Bay Point−Millbrae: color=255,255,0:
    adjlist=24th St Mission, Glen Park
EDGE: label=BART Pittsburgh/Bay Point−Millbrae: color=255,255,0:
    adjlist=Glen Park, Balboa Park
EDGE: label=BART Pittsburgh/Bay Point−Millbrae: color=255,255,0:
    adjlist=Balboa Park, Daly City
EDGE: label=BART Pittsburgh/Bay Point−Millbrae: color=255,255,0:
    adjlist=Daly City, Colma
EDGE: label=BART Pittsburgh/Bay Point−Millbrae: color=255,255,0:
    adjlist=Colma, South San Francisco
EDGE: label=BART Pittsburgh/Bay Point−Millbrae: color=255,255,0:
    adjlist=South San Francisco, San Bruno
EDGE: label=BART Pittsburgh/Bay Point−Millbrae: color=255,255,0:
    adjlist=San Bruno, Millbrae
EDGE: label=BART Dublin/Pleasanton−SF Airport (SFO): color
    =0,56,255: adjlist=Dublin/Pleasanton, Castro Valley
EDGE: label=BART Dublin/Pleasanton−SF Airport (SFO): color
    =0,56,255: adjlist=Castro Valley, Bay Fair
EDGE: label=BART Dublin/Pleasanton−SF Airport (SFO): color
    =0,56,255: adjlist=Bay Fair, San Leandro
EDGE: label=BART Dublin/Pleasanton−SF Airport (SFO): color
    =0,56,255: adjlist=San Leandro, Coliseum/Oakland Airport
EDGE: label=BART Dublin/Pleasanton−SF Airport (SFO): color
    =0,56,255: adjlist=Coliseum/Oakland Airport, Fruitvale
EDGE: label=BART Dublin/Pleasanton−SF Airport (SFO): color
    =0,56,255: adjlist=Fruitvale, Lake Merritt
EDGE: label=BART Dublin/Pleasanton−SF Airport (SFO): color
    =0,56,255: adjlist=Lake Merritt, West Oakland
EDGE: label=BART Dublin/Pleasanton−SF Airport (SFO): color
    =0,56,255: adjlist=West Oakland, Embarcadero
EDGE: label=BART Dublin/Pleasanton−SF Airport (SFO): color
    =0,56,255: adjlist=Embarcadero, Montgomery St
EDGE: label=BART Dublin/Pleasanton−SF Airport (SFO): color
    =0,56,255: adjlist=Montgomery St, Powell St
EDGE: label=BART Dublin/Pleasanton−SF Airport (SFO): color
    =0,56,255: adjlist=Powell St, Civic Center
EDGE: label=BART Dublin/Pleasanton−SF Airport (SFO): color
    =0,56,255: adjlist=Civic Center, 16th St Mission
EDGE: label=BART Dublin/Pleasanton−SF Airport (SFO): color
    =0,56,255: adjlist=16th St Mission, 24th St Mission
EDGE: label=BART Dublin/Pleasanton−SF Airport (SFO): color
    =0,56,255: adjlist=24th St Mission, Glen Park
EDGE: label=BART Dublin/Pleasanton−SF Airport (SFO): color
    =0,56,255: adjlist=Glen Park, Balboa Park
EDGE: label=BART Dublin/Pleasanton−SF Airport (SFO): color
    =0,56,255: adjlist=Balboa Park, Daly City

EDGE: label=BART Dublin/Pleasanton−SF Airport (SFO): color
    =0,56,255: adjlist=Daly City, Colma
EDGE: label=BART Dublin/Pleasanton−SF Airport (SFO): color
    =0,56,255: adjlist=Colma, South San Francisco
EDGE: label=BART Dublin/Pleasanton−SF Airport (SFO): color
    =0,56,255: adjlist=South San Francisco, San Bruno
EDGE: label=BART Dublin/Pleasanton−SF Airport (SFO): color
    =0,56,255: adjlist=San Bruno, San Francisco International
    Airport (SFO)
EDGE: label=BART Millbrae−SF Airport (SFO): color=123,9,90:
    adjlist=Millbrae, San Francisco International Airport (SFO)

B.9   Stockholm

NODE: label=Norsborg: x=17.8145: y=59.2438
NODE: label=Slussen: x=18.0724: y=59.3195
NODE: label=M\"orby Centrum: x=18.0361: y=59.3987
NODE: label=Mariatorget: x=18.0633: y=59.3169
NODE: label=Zinkensdamm: x=18.0502: y=59.3178
NODE: label=Hornstull: x=18.0341: y=59.3158
NODE: label=Liljeholmen: x=18.0229: y=59.3107
NODE: label=Medborgarplatsen: x=18.0735: y=59.3143
NODE: label=Skanstull: x=18.0762: y=59.3079
NODE: label=Gullmarsplan: x=18.0805: y=59.2990
NODE: label=Gamla Stan: x=18.0671: y=59.3233
NODE: label=T−Centralen: x=18.0593: y=59.3310
NODE: label=Kungst\"adgarden: x=18.0735: y=59.3307
NODE: label=H\"ortorget: x=18.0637: y=59.3354
NODE: label=\"Ostermalmstorg: x=18.0741: y=59.3350
NODE: label=Odenplan: x=18.0497: y=59.3430
NODE: label=Sank Eriksplan: x=18.0365: y=59.3398
NODE: label=Fridhemsplan: x=18.0324: y=59.3343
NODE: label=Radhuset: x=18.0421: y=59.3303
NODE: label=Thorildsplan: x=18.0155: y=59.3319
NODE: label=Kristineberg: x=18.0035: y=59.3329
NODE: label=Alvik: x=17.9801: y=59.3337
NODE: label=Stora Mossen: x=17.9662: y=59.3345
NODE: label=Stadshagen: x=18.0175: y=59.3370
NODE: label=V\"astra Skogen: x=18.0041: y=59.3475
NODE: label=Huvudsta: x=17.9857: y=59.3495
NODE: label=Vreten: x=17.9740: y=59.3542
NODE: label=Sundbybergs Centrum: x=17.9722: y=59.3609
NODE: label=Duvbo: x=17.9647: y=59.3678
NODE: label=Rissne: x=17.9400: y=59.3758
NODE: label=Brommaplan: x=17.9391: y=59.3383
NODE: label=Abrahamsberg: x=17.9528: y=59.3366
NODE: label=\"Akeshov: x=17.9249: y=59.3421
NODE: label=\"Angbyplan: x=17.9070: y=59.3418

NODE: label=Islandstorget:x=17.8939:y=59.3459
NODE: label=Blackeberg:x=17.8827:y=59.3482
NODE: label=Racksta:x=17.8818:y=59.3547
NODE: label=V\"allingby:x=17.8721:y=59.3633
NODE: label=Johannelund:x=17.8575:y=59.3679
NODE: label=H\"asselby Gard:x=17.8438:y=59.3669
NODE: label=H\"asselby Strand:x=17.8326:y=59.3612
NODE: label=Solna Centrum:x=17.9989:y=59.3588
NODE: label=N\"ackrosen:x=17.9834:y=59.3667
NODE: label=Hallonbergen:x=17.9693:y=59.3754
NODE: label=Kista:x=17.9424:y=59.4030
NODE: label=Husby:x=17.9255:y=59.4102
NODE: label=Akalla:x=17.9128:y=59.4148
NODE: label=Stadion:x=18.0817:y=59.3430
NODE: label=Tekniska H\"ogskolan:x=18.0718:y=59.3458
NODE: label=Universitet:x=18.0550:y=59.3655
NODE: label=Bergshamra:x=18.0367:y=59.3815
NODE: label=Danderyds Sjukhus:x=18.0414:y=59.3918
NODE: label=Karlaplan:x=18.0908:y=59.3388
NODE: label=G\"ardet:x=18.0996:y=59.3466
NODE: label=Ropsten:x=18.1023:y=59.3573
NODE: label=Blasut:x=18.0915:y=59.2904
NODE: label=Midsommarkransen:x=18.0120:y=59.3018
NODE: label=Enskede Gard:x=18.0703:y=59.2894
NODE: label=Skarmarbrink:x=18.0905:y=59.2954
NODE: label=Fru\"angen:x=17.9650:y=59.2859
NODE: label=V\"astertorp:x=17.9667:y=59.2914
NODE: label=Telefonplan:x=17.9972:y=59.2983
NODE: label=H\"agerstens\"asen:x=17.9792:y=59.2956
NODE: label=Aspudden:x=18.0014:y=59.3064
NODE: label=Satra:x=17.9214:y=59.2850
NODE: label=Bred\"ang:x=17.9338:y=59.2948
NODE: label=M\"alarh\"olden:x=17.9573:y=59.3010
NODE: label=\"Ornsberg:x=17.9892:y=59.3055
NODE: label=Axelsberg:x=17.9755:y=59.3044
NODE: label=Sk\"arholmen:x=17.9071:y=59.2771
NODE: label=Varberg:x=17.8902:y=59.2759
NODE: label=Varby Gard:x=17.8845:y=59.2646
NODE: label=Masmo:x=17.8802:y=59.2497
NODE: label=Fittja:x=17.8610:y=59.2475
NODE: label=Hallunda:x=17.8255:y=59.2433
NODE: label=Alby:x=17.8457:y=59.2395
NODE: label=Radmansgatan:x=18.0588:y=59.3406
NODE: label=Rinkeby:x=17.9288:y=59.3881
NODE: label=Tensta:x=17.9012:y=59.3944
NODE: label=Hjulsta:x=17.8886:y=59.3968
NODE: label=Ragsved:x=18.0283:y=59.2566
NODE: label=H\"ogdalen:x=18.0429:y=59.2638

NODE: label=Bandhagen : x=18.0495:y=59.2704
NODE: label=Globen : x=18.0779:y=59.2942
NODE: label=Sockenplan : x=18.0701:y=59.2835
NODE: label=Stureby : x=18.0556:y=59.2746
NODE: label=Svedmyra : x=18.0673:y=59.2777
NODE: label=Hags\"atra : x=18.0124:y=59.2628
NODE: label=Sandsborg : x=18.0923:y=59.2848
NODE: label=Hammarbyh\"ojden : x=18.1045:y=59.2947
NODE: label=Bj\"orkhagen : x=18.1156:y=59.2912
NODE: label=K\"arrtorp : x=18.1144:y=59.2845
NODE: label=Bargarmossen : x=18.1315:y=59.2762
NODE: label=Skarpn\"ack : x=18.1334:y=59.2668
NODE: label=Skogskyrkogarden : x=18.0955:y=59.2792
NODE: label=Farsta : x=18.0932:y=59.2435
NODE: label=Farsta Strand : x=18.1019:y=59.2350
NODE: label=H\"okar\"angen : x=18.0826:y=59.2579
NODE: label=Gubb\"angen : x=18.0820:y=59.2629
NODE: label=Tallkrogen : x=18.0853:y=59.2711
EDGE: label=Red : color =234,80,0: adjlist=M\"orby Centrum ,
    Danderyds Sjukhus
EDGE: label=Red : color =234,80,0: adjlist=Danderyds Sjukhus ,
    Bergshamra
EDGE: label=Red : color =234,80,0: adjlist=Bergshamra , Universitet
EDGE: label=Red : color =234,80,0: adjlist=Universitet , Tekniska H\"
    ogskolan
EDGE: label=Red : color =234,80,0: adjlist=Tekniska H\"ogskolan ,
    Stadion
EDGE: label=Red : color =234,80,0: adjlist=Stadion , \"Ostermalmstorg
EDGE: label=Red : color =234,80,0: adjlist=\"Ostermalmstorg , T–
    Centralen
EDGE: label=Red : color =234,80,0: adjlist=T–Centralen , Gamla Stan
EDGE: label=Red : color =234,80,0: adjlist=Gamla Stan , Slussen
EDGE: label=Red : color =234,80,0: adjlist=Slussen , Mariatorget
EDGE: label=Red : color =234,80,0: adjlist=Mariatorget , Zinkensdamm
EDGE: label=Red : color =234,80,0: adjlist=Zinkensdamm , Hornstull
EDGE: label=Red : color =234,80,0: adjlist=Hornstull , Liljeholmen
EDGE: label=Red : color =234,80,0: adjlist=Liljeholmen , Aspudden
EDGE: label=Red : color =234,80,0: adjlist=Aspudden , \"Ornsberg
EDGE: label=Red : color =234,80,0: adjlist=\"Ornsberg , Axelsberg
EDGE: label=Red : color =234,80,0: adjlist=Axelsberg , M\"alarh\"
    olden
EDGE: label=Red : color =234,80,0: adjlist=M\"alarh\"olden , Bred\"
    ang
EDGE: label=Red : color =234,80,0: adjlist=Bred\"ang , Satra
EDGE: label=Red : color =234,80,0: adjlist=Satra , Sk\"arholmen
EDGE: label=Red : color =234,80,0: adjlist=Sk\"arholmen , Varberg
EDGE: label=Red : color =234,80,0: adjlist=Varberg , Varby Gard
EDGE: label=Red : color =234,80,0: adjlist=Varby Gard , Masmo

```
EDGE: label=Red: color =234,80,0: adjlist=Masmo, Fittja
EDGE: label=Red: color =234,80,0: adjlist=Fittja , Alby
EDGE: label=Red: color =234,80,0: adjlist=Alby , Hallunda
EDGE: label=Red: color =234,80,0: adjlist=Hallunda , Norsborg
EDGE: label=Red: color =234,80,0: adjlist=Liljeholmen ,
    Midsommarkransen
EDGE: label=Red: color =234,80,0: adjlist=Midsommarkransen ,
    Telefonplan
EDGE: label=Red: color =234,80,0: adjlist=Telefonplan , H\" agerstens
    \" asen
EDGE: label=Red: color =234,80,0: adjlist=H\" agerstens \" asen , V\"
    astertorp
EDGE: label=Red: color =234,80,0: adjlist=V\" astertorp , Fru\" angen
EDGE: label=Red: color =234,80,0: adjlist=Ropsten , G\" ardet
EDGE: label=Red: color =234,80,0: adjlist=G\" ardet , Karlaplan
EDGE: label=Red: color =234,80,0: adjlist=Karlaplan , \"
    Ostermalmstorg
EDGE: label=Blue: color =32,42,132: adjlist=Akalla , Husby
EDGE: label=Blue: color =32,42,132: adjlist=Husby , Kista
EDGE: label=Blue: color =32,42,132: adjlist=Kista , Hallonbergen
EDGE: label=Blue: color =32,42,132: adjlist=Hallonbergen , N\"
    ackrosen
EDGE: label=Blue: color =32,42,132: adjlist=N\" ackrosen , Solna
    Centrum
EDGE: label=Blue: color =32,42,132: adjlist=Solna Centrum, V\" astra
    Skogen
EDGE: label=Blue: color =32,42,132: adjlist=V\" astra Skogen ,
    Stadshagen
EDGE: label=Blue: color =32,42,132: adjlist=Stadshagen ,
    Fridhemsplan
EDGE: label=Blue: color =32,42,132: adjlist=Fridhemsplan , Radhuset
EDGE: label=Blue: color =32,42,132: adjlist=Radhuset , T–Centralen
EDGE: label=Blue: color =32,42,132: adjlist=T–Centralen , Kungst\"
    adgarden
EDGE: label=Blue: color =32,42,132: adjlist=Hjulsta , Tensta
EDGE: label=Blue: color =32,42,132: adjlist=Tensta , Rinkeby
EDGE: label=Blue: color =32,42,132: adjlist=Rinkeby , Rissne
EDGE: label=Blue: color =32,42,132: adjlist=Rissne , Duvbo
EDGE: label=Blue: color =32,42,132: adjlist=Duvbo, Sundbybergs
    Centrum
EDGE: label=Blue: color =32,42,132: adjlist=Sundbybergs Centrum,
    Vreten
EDGE: label=Blue: color =32,42,132: adjlist=Vreten , Huvudsta
EDGE: label=Blue: color =32,42,132: adjlist=Huvudsta , V\" astra
    Skogen
EDGE: label=Green: color =133,186,13: adjlist=H\" asselby Strand , H
    \" asselby Gard
```

EDGE: label=Green : color =133 ,186 ,13: adjlist=H\" asselby Gard ,
    Johannelund
EDGE: label=Green : color =133 ,186 ,13: adjlist=Johannelund , V\"
    allingby
EDGE: label=Green : color =133 ,186 ,13: adjlist=V\" allingby , Racksta
EDGE: label=Green : color =133 ,186 ,13: adjlist=Racksta , Blackeberg
EDGE: label=Green : color =133 ,186 ,13: adjlist=Blackeberg ,
    Islandstorget
EDGE: label=Green : color =133 ,186 ,13: adjlist=Islandstorget , \"
    Angbyplan
EDGE: label=Green : color =133 ,186 ,13: adjlist =\" Angbyplan , \"
    Akeshov
EDGE: label=Green : color =133 ,186 ,13: adjlist =\" Akeshov , Brommaplan
EDGE: label=Green : color =133 ,186 ,13: adjlist=Brommaplan ,
    Abrahamsberg
EDGE: label=Green : color =133 ,186 ,13: adjlist=Abrahamsberg , Stora
    Mossen
EDGE: label=Green : color =133 ,186 ,13: adjlist=Stora Mossen , Alvik
EDGE: label=Green : color =133 ,186 ,13: adjlist=Alvik , Kristineberg
EDGE: label=Green : color =133 ,186 ,13: adjlist=Kristineberg ,
    Thorildsplan
EDGE: label=Green : color =133 ,186 ,13: adjlist=Thorildsplan ,
    Fridhemsplan
EDGE: label=Green : color =133 ,186 ,13: adjlist=Fridhemsplan , Sank
    Eriksplan
EDGE: label=Green : color =133 ,186 ,13: adjlist=Sank Eriksplan ,
    Odenplan
EDGE: label=Green : color =133 ,186 ,13: adjlist=Odenplan ,
    Radmansgatan
EDGE: label=Green : color =133 ,186 ,13: adjlist=Radmansgatan , H\"
    ortorget
EDGE: label=Green : color =133 ,186 ,13: adjlist=H\" ortorget , T–
    Centralen
EDGE: label=Green : color =133 ,186 ,13: adjlist=T–Centralen , Gamla
    Stan
EDGE: label=Green : color =133 ,186 ,13: adjlist=Gamla Stan , Slussen
EDGE: label=Green : color =133 ,186 ,13: adjlist=Slussen ,
    Medborgarplatsen
EDGE: label=Green : color =133 ,186 ,13: adjlist=Medborgarplatsen ,
    Skanstull
EDGE: label=Green : color =133 ,186 ,13: adjlist=Skanstull ,
    Gullmarsplan
EDGE: label=Green : color =133 ,186 ,13: adjlist=Gullmarsplan , Globen
EDGE: label=Green : color =133 ,186 ,13: adjlist=Globen , Enskede Gard
EDGE: label=Green : color =133 ,186 ,13: adjlist=Enskede Gard ,
    Sockenplan
EDGE: label=Green : color =133 ,186 ,13: adjlist=Sockenplan , Svedmyra
EDGE: label=Green : color =133 ,186 ,13: adjlist=Svedmyra , Stureby

EDGE: label=Green : color =133,186,13: adjlist=Stureby , Bandhagen
EDGE: label=Green : color =133,186,13: adjlist=Bandhagen , H\" ogdalen
EDGE: label=Green : color =133,186,13: adjlist=H\" ogdalen , Ragsved
EDGE: label=Green : color =133,186,13: adjlist=Ragsved , Hags\" atra
EDGE: label=Green : color =133,186,13: adjlist=Gullmarsplan ,
    Skarmarbrink
EDGE: label=Green : color =133,186,13: adjlist=Skarmarbrink , Blasut
EDGE: label=Green : color =133,186,13: adjlist=Blasut , Sandsborg
EDGE: label=Green : color =133,186,13: adjlist=Sandsborg ,
    Skogskyrkogarden
EDGE: label=Green : color =133,186,13: adjlist=Skogskyrkogarden ,
    Tallkrogen
EDGE: label=Green : color =133,186,13: adjlist=Tallkrogen , Gubb\"
    angen
EDGE: label=Green : color =133,186,13: adjlist=Gubb\" angen , H\" okar
    \" angen
EDGE: label=Green : color =133,186,13: adjlist=H\" okar\" angen ,
    Farsta
EDGE: label=Green : color =133,186,13: adjlist=Farsta , Farsta Strand
EDGE: label=Green : color =133,186,13: adjlist=Skarmarbrink ,
    Hammarbyh\" ojden
EDGE: label=Green : color =133,186,13: adjlist=Hammarbyh\" ojden , Bj
    \" orkhagen
EDGE: label=Green : color =133,186,13: adjlist=Bj\" orkhagen , K\"
    arrtorp
EDGE: label=Green : color =133,186,13: adjlist=K\" arrtorp ,
    Bargarmossen
EDGE: label=Green : color =133,186,13: adjlist=Bargarmossen , Skarpn
    \" ack

B.10 Sydney

NODE: label=Berowra : x=151.1526:y=−33.6238
NODE: label=Mount Kuring−gai : x=151.1367:y=−33.6533
NODE: label=Mount Colah : x=151.1150:y=−33.6717
NODE: label=Asquith : x=151.1081:y=−33.6886
NODE: label=Hornsby : x=151.0984:y=−33.7027
NODE: label=Waitara : x=151.1045:y=−33.7100
NODE: label=Wahroonga : x=151.1170:y=−33.7177
NODE: label=Warrawee : x=151.1217:y=−33.7243
NODE: label=Turramurra : x=151.1288:y=−33.7324
NODE: label=Pymble : x=151.1422:y=−33.7449
NODE: label=Gordon : x=151.1543:y=−33.7560
NODE: label=Killara : x=151.1619:y=−33.7656
NODE: label=Lindfield : x=151.1692:y=−33.7752
NODE: label=Roseville : x=151.1765:y=−33.7837
NODE: label=Chatswood : x=151.1810:y=−33.7974
NODE: label=Artarmon : x=151.1857:y=−33.8091

NODE: label=St Leonards:x=151.1944:y=−33.8233
NODE: label=Wollstonecraft:x=151.1918:y=−33.8319
NODE: label=Waverton:x=151.1977:y=−33.8379
NODE: label=North Sydney:x=151.2073:y=−33.8411
NODE: label=Milsons Point:x=151.2119:y=−33.8461
NODE: label=Wynyard:x=151.2056:y=−33.8660
NODE: label=Circular Quay:x=151.2110:y=−33.8613
NODE: label=Martin Place:x=151.2118:y=−33.8679
NODE: label=St James:x=151.2121:y=−33.8702
NODE: label=Kings Cross:x=151.2225:y=−33.8749
NODE: label=Edgecliff:x=151.2371:y=−33.8799
NODE: label=Bondi Junction:x=151.2474:y=−33.8910
NODE: label=Museum:x=151.2098:y=−33.8763
NODE: label=Town Hall:x=151.2069:y=−33.8739
NODE: label=Central:x=151.2064:y=−33.8832
NODE: label=Redfern:x=151.1988:y=−33.8919
NODE: label=Macdonaldtown:x=151.1863:y=−33.8967
NODE: label=Newtown:x=151.1796:y=−33.8977
NODE: label=Stanmore:x=151.1639:y=−33.8928
NODE: label=Petersham:x=151.1556:y=−33.8938
NODE: label=Lewisham:x=151.1473:y=−33.8932
NODE: label=Summer Hill:x=151.1390:y=−33.8904
NODE: label=Ashfield:x=151.1254:y=−33.8874
NODE: label=Croydon:x=151.1159:y=−33.8836
NODE: label=Burwood:x=151.1039:y=−33.8771
NODE: label=Strathfield:x=151.0941:y=−33.8715
NODE: label=North Strathfield:x=151.0883:y=−33.8590
NODE: label=Concord West:x=151.0856:y=−33.8489
NODE: label=Rhodes:x=151.0872:y=−33.8300
NODE: label=Meadowbank:x=151.0900:y=−33.8168
NODE: label=West Ryde:x=151.0903:y=−33.8071
NODE: label=Denistone:x=151.0872:y=−33.7998
NODE: label=Eastwood:x=151.0821:y=−33.7901
NODE: label=Epping:x=151.0819:y=−33.7727
NODE: label=Cheltenham:x=151.0787:y=−33.7557
NODE: label=Beecroft:x=151.0662:y=−33.7503
NODE: label=Pennant Hills:x=151.0718:y=−33.7387
NODE: label=Thornleigh:x=151.0781:y=−33.7320
NODE: label=Normanhurst:x=151.0965:y=−33.7212
NODE: label=Carlingford:x=151.0472:y=−33.7820
NODE: label=Telopea:x=151.0414:y=−33.7940
NODE: label=Dundas:x=151.0344:y=−33.8033
NODE: label=Rydalmere:x=151.0290:y=−33.8103
NODE: label=Camellia:x=151.0243:y=−33.8188
NODE: label=Rosehill:x=151.0219:y=−33.8236
NODE: label=Clyde:x=151.0169:y=−33.8359
NODE: label=Lidcombe:x=151.0453:y=−33.8638
NODE: label=Auburn:x=151.0328:y=−33.8494

NODE: label=Olympic Park:x=151.0694:y=−33.8466
NODE: label=Flemington:x=151.0698:y=−33.8649
NODE: label=Homebush:x=151.0869:y=−33.8669
NODE: label=Berala:x=151.0318:y=−33.8721
NODE: label=Regents Park:x=151.0239:y=−33.8830
NODE: label=Granville:x=151.0126:y=−33.8331
NODE: label=Harris Park:x=151.0077:y=−33.8235
NODE: label=Parramatta:x=151.0056:y=−33.8177
NODE: label=Westmead:x=150.9876:y=−33.8085
NODE: label=Wentworthville:x=150.9727:y=−33.8072
NODE: label=Pendle Hill:x=150.9564:y=−33.8014
NODE: label=Toongabbie:x=150.9516:y=−33.7875
NODE: label=Seven Hills:x=150.9357:y=−33.7743
NODE: label=Blacktown:x=150.9096:y=−33.7690
NODE: label=Marayong:x=150.9002:y=−33.7467
NODE: label=Quakers Hill:x=150.8870:y=−33.7282
NODE: label=Schofields:x=150.8698:y=−33.6968
NODE: label=Riverstone:x=150.8605:y=−33.6796
NODE: label=Vineyard:x=150.8513:y=−33.6508
NODE: label=Mulgrave:x=150.8304:y=−33.6266
NODE: label=Windsor:x=150.8112:y=−33.6138
NODE: label=Clarendon:x=150.7880:y=−33.6088
NODE: label=East Richmond:x=150.7590:y=−33.6016
NODE: label=Richmond:x=150.7525:y=−33.5989
NODE: label=Emu Plains:x=150.6720:y=−33.7455
NODE: label=Penrith:x=150.6960:y=−33.7501
NODE: label=Kingswood:x=150.7195:y=−33.7582
NODE: label=Werrington:x=150.7574:y=−33.7591
NODE: label=St Marys:x=150.7753:y=−33.7621
NODE: label=Mount Druitt:x=150.8202:y=−33.7697
NODE: label=Rooty Hill:x=150.8443:y=−33.7716
NODE: label=Doonside:x=150.8692:y=−33.7638
NODE: label=Merrylands:x=150.9925:y=−33.8366
NODE: label=Guildford:x=150.9845:y=−33.8543
NODE: label=Yennora:x=150.9706:y=−33.8649
NODE: label=Fairfield:x=150.9575:y=−33.8721
NODE: label=Canley Vale:x=150.9440:y=−33.8868
NODE: label=Cabramatta:x=150.9387:y=−33.8949
NODE: label=Warwick Farm:x=150.9351:y=−33.9135
NODE: label=Liverpool:x=150.9268:y=−33.9253
NODE: label=Casula:x=150.9118:y=−33.9501
NODE: label=Glenfield:x=150.8932:y=−33.9722
NODE: label=Macquarie Fields:x=150.8787:y=−33.9852
NODE: label=Ingleburn:x=150.8648:y=−33.9974
NODE: label=Minto:x=150.8424:y=−34.0275
NODE: label=Leumeah:x=150.8302:y=−34.0509
NODE: label=Campbelltown:x=150.8143:y=−34.0637
NODE: label=Macarthur:x=150.7969:y=−34.0720

NODE: label=Carramar:x=150.9610:y=−33.8845
NODE: label=Villawood:x=150.9760:y=−33.8809
NODE: label=Leightonfield:x=150.9849:y=−33.8815
NODE: label=Chester Hill:x=150.9996:y=−33.8835
NODE: label=Sefton:x=151.0116:y=−33.8851
NODE: label=Birrong:x=151.0239:y=−33.8932
NODE: label=Yagoona:x=151.0242:y=−33.9068
NODE: label=Bankstown:x=151.0342:y=−33.9179
NODE: label=Punchbowl:x=151.0553:y=−33.9255
NODE: label=Wiley Park:x=151.0672:y=−33.9230
NODE: label=Lakemba:x=151.0757:y=−33.9201
NODE: label=Belmore:x=151.0886:y=−33.9172
NODE: label=Campsie:x=151.1026:y=−33.9105
NODE: label=Canterbury:x=151.1187:y=−33.9121
NODE: label=Hurlstone Park:x=151.1318:y=−33.9104
NODE: label=Dulwich Hill:x=151.1410:y=−33.9111
NODE: label=Marrickville:x=151.1547:y=−33.9144
NODE: label=Erskineville:x=151.1855:y=−33.9001
NODE: label=St Peters:x=151.1810:y=−33.9071
NODE: label=Sydenham:x=151.1663:y=−33.9146
NODE: label=Tempe:x=151.1563:y=−33.9244
NODE: label=Wolli Creek:x=151.1535:y=−33.9282
NODE: label=Green Square:x=151.2026:y=−33.9061
NODE: label=Mascot:x=151.1875:y=−33.9228
NODE: label=Domestic Airport:x=151.1834:y=−33.9341
NODE: label=International Airport:x=151.1647:y=−33.9339
NODE: label=Holsworthy:x=150.9574:y=−33.9632
NODE: label=East Hills:x=150.9850:y=−33.9616
NODE: label=Panania:x=150.9982:y=−33.9544
NODE: label=Revesby:x=151.0148:y=−33.9524
NODE: label=Padstow:x=151.0320:y=−33.9519
NODE: label=Riverwood:x=151.0521:y=−33.9515
NODE: label=Narwee:x=151.0701:y=−33.9475
NODE: label=Beverly Hills:x=151.0809:y=−33.9491
NODE: label=Kingsgrove:x=151.1007:y=−33.9406
NODE: label=Bexley North:x=151.1138:y=−33.9375
NODE: label=Bardwell Park:x=151.1252:y=−33.9315
NODE: label=Turrella:x=151.1403:y=−33.9299
NODE: label=Arncliffe:x=151.1473:y=−33.9365
NODE: label=Banksia:x=151.1403:y=−33.9454
NODE: label=Rockdale:x=151.1368:y=−33.9521
NODE: label=Kogarah:x=151.1328:y=−33.9617
NODE: label=Carlton:x=151.1235:y=−33.9683
NODE: label=Allawah:x=151.1145:y=−33.9697
NODE: label=Hurstville:x=151.1024:y=−33.9675
NODE: label=Penshurst:x=151.0887:y=−33.9661
NODE: label=Mortdale:x=151.0812:y=−33.9708
NODE: label=Oatley:x=151.0790:y=−33.9808

NODE: label=Como:x=151.0680:y=−34.0046
NODE: label=Jannali:x=151.0647:y=−34.0171
NODE: label=Sutherland:x=151.0572:y=−34.0319
NODE: label=Loftus:x=151.0513:y=−34.0455
NODE: label=Engadine:x=151.0150:y=−34.0679
NODE: label=Heathcote:x=151.0081:y=−34.0881
NODE: label=Waterfall:x=150.9947:y=−34.1348
NODE: label=Kirrawee:x=151.0712:y=−34.0349
NODE: label=Gymea:x=151.0853:y=−34.0349
NODE: label=Miranda:x=151.1022:y=−34.0363
NODE: label=Caringbah:x=151.1221:y=−34.0415
NODE: label=Woolooware:x=151.1443:y=−34.0477
NODE: label=Cronulla:x=151.1514:y=−34.056
EDGE: label=Eastern Suburbs and Illawarra Line:color=7,148,255:
   adjlist=Bondi Junction, Edgecliff, Kings Cross, Martin Place
   , Town Hall, Central, Redfern, Erskineville, St Peters,
   Sydenham, Tempe, Wolli Creek, Arncliffe, Banksia, Rockdale,
   Kogarah, Carlton, Allawah, Hurstville, Penshurst, Mortdale,
   Oatley, Como, Jannali, Sutherland, Loftus, Engadine,
   Heathcote, Waterfall
EDGE: label=Eastern Suburbs and Illawarra Line:color=7,148,255:
   adjlist=Sutherland, Kirrawee, Gymea, Miranda, Caringbah,
   Woolooware, Cronulla
EDGE: label=Bankstown Line:color=255,82,0:adjlist=Town Hall,
   Wynyard, Circular Quay, St James, Museum, Central, Redfern,
   Erskineville, St Peters, Marrickville, Dulwich Hill,
   Hurlstone Park, Canterbury, Campsie, Belmore, Lakemba, Wiley
   Park, Punchbowl, Bankstown, Yagoona, Birrong, Sefton,
   Chester Hill, Leightonfield, Villawood, Carramar, Cabramatta
   , Warwick Farm, Liverpool
EDGE: label=Bankstown Line:color=255,82,0:adjlist=Birrong,
   Regents Park, Berala, Lidcombe
EDGE: label=Inner West Line:color=122,115,255:adjlist=Liverpool,
   Warwick Farm, Cabramatta, Carramar, Villawood,
   Leightonfield, Chester Hill, Sefton, Regents Park, Berala,
   Lidcombe, Flemington, Homebush, Strathfield, Burwood,
   Croydon, Ashfield, Summer Hill, Lewisham, Petersham,
   Stanmore, Newtown, Macdonaldtown, Redfern, Central, Town
   Hall, Wynyard, Circular Quay, St James, Museum
EDGE: label=Cumberland Line:color=168,27,255:adjlist=
   Campbelltown, Leumeah, Minto, Ingleburn, Macquarie Fields,
   Glenfield, Casula, Liverpool, Warwick Farm, Cabramatta,
   Canley Vale, Fairfield, Yennora, Guildford, Merrylands,
   Harris Park, Parramatta, Westmead, Wentworthville, Pendle
   Hill, Toongabbie, Seven Hills, Blacktown

EDGE: label=Airport and East Hills Line: color=35,227,0: adjlist=
Macarthur, Campbelltown, Leumeah, Minto, Ingleburn,
Macquarie Fields, Glenfield, Holsworthy, East Hills, Panania
, Revesby, Padstow, Riverwood, Narwee, Beverly Hills,
Kingsgrove, Bexley North, Bardwell Park, Turrella, Wolli
Creek, International Airport, Domestic Airport, Mascot,
Green Square, Central, Museum, St James, Circular Quay,
Wynyard, Town Hall
EDGE: label=South Line: color=121,255,255: adjlist=Campbelltown,
Leumeah, Minto, Ingleburn, Macquarie Fields, Glenfield,
Casula, Liverpool, Warwick Farm, Cabramatta, Canley Vale,
Fairfield, Yennora, Guildford, Merrylands, Granville, Clyde,
Auburn, Lidcombe, Flemington, Homebush, Strathfield,
Burwood, Croydon, Ashfield, Summer Hill, Lewisham, Petersham
, Stanmore, Newtown, Macdonaldtown, Redfern, Central, Town
Hall, Wynyard, Circular Quay, St James, Museum
EDGE: label=North Shore and Western Lines: color=254,179,0:
adjlist=Emu Plains, Penrith, Kingswood, Werrington, St Marys
, Mount Druitt, Rooty Hill, Doonside, Blacktown, Seven Hills
, Toongabbie, Pendle Hill, Wentworthville, Westmead,
Parramatta, Harris Park, Granville, Clyde, Auburn, Lidcombe,
Flemington, Homebush, Strathfield, Burwood, Croydon,
Ashfield, Summer Hill, Lewisham, Petersham, Stanmore,
Newtown, Macdonaldtown, Redfern, Central, Town Hall, Wynyard
, Milsons Point, North Sydney, Waverton, Wollstonecraft, St
Leonards, Artarmon, Chatswood, Roseville, Lindfield, Killara
, Gordon, Pymble, Turramurra, Warrawee, Wahroonga, Waitara,
Hornsby, Asquith, Mount Colah, Mount Kuring−gai, Berowra
EDGE: label=North Shore and Western Lines: color=254,179,0:
adjlist=Richmond, East Richmond, Clarendon, Windsor,
Mulgrave, Vineyard, Riverstone, Schofields, Quakers Hill,
Marayong, Blacktown
EDGE: label=Northern Line: color=247,0,55: adjlist=Hornsby,
Normanhurst, Thornleigh, Pennant Hills, Beecroft, Cheltenham
, Epping, Eastwood, Denistone, West Ryde, Meadowbank, Rhodes
, Concord West, North Strathfield, Strathfield, Burwood,
Croydon, Ashfield, Summer Hill, Lewisham, Petersham,
Stanmore, Newtown, Macdonaldtown, Redfern, Central, Town
Hall, Wynyard, Milsons Point, North Sydney
EDGE: label=Carlingford Line: color=0,11,198: adjlist=Clyde,
Rosehill, Camellia, Rydalmere, Dundas, Telopea, Carlingford
EDGE: label=Olympic Park Sprint: color=0,0,0: adjlist=Lidcombe,
Olympic Park

B.11 Toronto

NODE: label=North York Centre: x=548.3656148413377: y
=637.6924963871645692.0

NODE: label=Union:x=534.9854062654489:y=149.41603630634563692.0
NODE: label=St Andrew:x=512.9701663477811:y
    =162.1193095696016 92.0
NODE: label=Osgoode:x=513.6985360369094:y
    =190.22721650690892692.0
NODE: label=St Patrick:x=513.6101189821484:y
    =215.288297837154692.0
NODE: label=Wellesley:x=564.2825154560209:y
    =268.4576291498545692.0
NODE: label=Rosedale:x=549.4386125984867:y=323.007699818422692.0
NODE: label=Summerhill:x=551.3083169666285:y
    =359.92153347510896692.0
NODE: label=St Clair:x=552.0416506817132:y
    =395.0265476351312692.0
NODE: label=Davisville:x=548.2876809114198:y
    =442.26962390056184692.0
NODE: label=Spadina:x=480.8517476767438:y
    =291.37480302812986692.0
NODE: label=Eglington:x=547.8382397801869:y
    =486.54438806941386 92.0
NODE: label=Lawrence:x=549.3916631335366:y
    =525.6352244419058692.0
NODE: label=York Mills:x=549.367640778923:y
    =566.48042522553016 92.0
NODE: label=Sheppard−Yonge:x=551.284816355123:y
    =605.1201681694863692.0
NODE: label=Finch:x=550.8371768151151:y=671.0981218621517692.0
NODE: label=Bessarion:x=673.6638726496225:y
    =637.2966514837284692.0
NODE: label=Bayview:x=619.2563527217508:y=627.9832570106405692.0
NODE: label=Leslie:x=729.1709737664762:y=654.8582831767869692.0
NODE: label=Don Mills:x=779.6605791855211:y
    =658.7592630811425692.0
NODE: label=Dupont:x=475.10963401099434:y=326.82204897579692.0
NODE: label=St Clair West:x=420.7467101798818:y
    =393.31365872394656692.0
NODE: label=Bathurst:x=459.433874576835:y=288.4817577354929692.0
NODE: label=Christie:x=425.7317277352348:y
    =288.24399723160866692.0
NODE: label=Ossington:x=390.4904490149398:y
    =290.6641745650783692.0
NODE: label=Dufferin:x=363.5208404208381:y
    =289.70859128315396692.0
NODE: label=Lansdowne:x=337.05411555862634:y
    =295.5685832707462692.0
NODE: label=Dundas West:x=308.63675433094295:y
    =298.63264130070627692.0
NODE: label=Keele:x=278.1608021637177:y=297.4022257844125692.0

NODE: label=High Park : x=250.89186053236813 : y
    =299.3530140074223692.0
NODE: label=Runnymede : x=218.42970985990746 : y
    =299.77668153357126692.0
NODE: label=Jane : x=182.5962614468292 : y=300.66328230433544692.0
NODE: label=Old Mill : x=147.30234879174463 : y
    =299.6652910497543692.0
NODE: label=Royal York : x=110.4733633922744 : y
    =298.13705994154395692.0
NODE: label=Islington : x=77.15119181226339 : y
    =298.21038567972073692.0
NODE: label=Kipling : x=42.33248533112442 : y=300.8690008314476692.0
NODE: label=Bloor−Yonge : x=554.2434471922602 : y
    =296.0001374120451692.0
NODE: label=Bay : x=535.7131858086171 : y=295.579957021216055692.0
NODE: label=St George : x=509.18001734150425 : y
    =294.37532458834033692.0
NODE: label=Museum : x=513.5099140132093 : y=266.78771450949955692.0
NODE: label=Queen's Park : x=513.6889089330482 : y
    =238.70549958688895692.0
NODE: label=College : x=559.2061257777948 : y
    =238.90679448436816692.0
NODE: label=Dundas : x=561.1316717043912 : y=213.42064344042603692.0
NODE: label=Queen : x=562.1244442947623 : y=188.80487496273702692.0
NODE: label=King : x=559.1061008130134 : y=164.22731135568415692.0
NODE: label=Sherbourne : x=583.1827965800443 : y
    =297.069553286791692.0
NODE: label=Castle Frank : x=615.0495542265041 : y
    =293.6085436965329692.0
NODE: label=Broadview : x=650.101774858699 : y
    =293.66215574191847692.0
NODE: label=Chester : x=680.0663347230288 : y
    =292.81886037740753692.0
NODE: label=Pape : x=703.5411023844736 : y=295.4480271603864692.0
NODE: label=Donlands : x=729.7289981176939 : y
    =297.93355121967716692.0
NODE: label=Greenwood : x=753.2786480302722 : y
    =297.8691433725385692.0
NODE: label=Coxwell : x=780.1063759895965 : y
    =297.20369839133514692.0
NODE: label=Woodbine : x=812.1914533588848 : y
    =305.2910061438438692.0
NODE: label=Main Street : x=838.3992982377013 : y
    =320.83416566832886692.0
NODE: label=Victoria Park : x=853.3786830714891 : y
    =339.77212999281164692.0
NODE: label=Warden : x=869.5270225475806 : y=364.1712868241165692.0
NODE: label=Kennedy : x=875.9828865340006 : y=393.1506888781412692.0

NODE: label=Lawrence East : x=876.6193897026717 : y
=424.8275297596737692.0
NODE: label=Ellesmere : x=874.053448555486 : y
=464.2684808438154692.0
NODE: label=Midland : x=908.6953883467754 : y
=488.48023901669956692.0
NODE: label=Scarborough Centre : x=943.8338711405282 : y
=488.5509340626954692.0
NODE: label=McCowan : x=989.7821300689836 : y=491.5011334766272692.0
NODE: label=Eglington West : x=375.2912193114021 : y
=477.0711599507095692.0
NODE: label=Glencairn : x=341.0749312294223 : y=515.79228944121692.0
NODE: label=Eginton West : x=312.1464905532554 : y
=563.3174345442742692.0
NODE: label=Yorkdale : x=308.43619446331184 : y
=589.8486207689948692.0
NODE: label=Wilson : x=306.82276942709404 : y=624.1184864518065692.0
NODE: label=Downsview : x=305.6772253459253 : y
=655.9688768792022692.0
EDGE: label=Bloor−Darnforth : color =0,150,35: adjlist=Kipling ,
Islington
EDGE: label=Bloor−Darnforth : color =0,150,35: adjlist=Islington ,
Royal York
EDGE: label=Bloor−Darnforth : color =0,150,35: adjlist=Royal York,
Old Mill
EDGE: label=Bloor−Darnforth : color =0,150,35: adjlist=Old Mill ,
Jane
EDGE: label=Bloor−Darnforth : color =0,150,35: adjlist=Jane ,
Runnymede
EDGE: label=Bloor−Darnforth : color =0,150,35: adjlist=Runnymede,
High Park
EDGE: label=Bloor−Darnforth : color =0,150,35: adjlist=High Park ,
Keele
EDGE: label=Bloor−Darnforth : color =0,150,35: adjlist=Keele , Dundas
West
EDGE: label=Bloor−Darnforth : color =0,150,35: adjlist=Dundas West,
Lansdowne
EDGE: label=Bloor−Darnforth : color =0,150,35: adjlist=Lansdowne,
Dufferin
EDGE: label=Bloor−Darnforth : color =0,150,35: adjlist=Dufferin ,
Ossington
EDGE: label=Bloor−Darnforth : color =0,150,35: adjlist=Ossington ,
Christie
EDGE: label=Bloor−Darnforth : color =0,150,35: adjlist=Christie ,
Bathurst
EDGE: label=Bloor−Darnforth : color =0,150,35: adjlist=Bathurst ,
Spadina

EDGE: label=Bloor−Darnforth : color =0,150,35: adjlist=Spadina , St
    George
EDGE: label=Bloor−Darnforth : color =0,150,35: adjlist=St George ,
    Bay
EDGE: label=Bloor−Darnforth : color =0,150,35: adjlist=Bay, Bloor−
    Yonge
EDGE: label=Bloor−Darnforth : color =0,150,35: adjlist=Bloor−Yonge ,
    Sherbourne
EDGE: label=Bloor−Darnforth : color =0,150,35: adjlist=Sherbourne ,
    Castle Frank
EDGE: label=Bloor−Darnforth : color =0,150,35: adjlist=Castle Frank ,
    Broadview
EDGE: label=Bloor−Darnforth : color =0,150,35: adjlist=Broadview ,
    Chester
EDGE: label=Bloor−Darnforth : color =0,150,35: adjlist=Chester , Pape
EDGE: label=Bloor−Darnforth : color =0,150,35: adjlist=Pape ,
    Donlands
EDGE: label=Bloor−Darnforth : color =0,150,35: adjlist=Donlands ,
    Greenwood
EDGE: label=Bloor−Darnforth : color =0,150,35: adjlist=Greenwood ,
    Coxwell
EDGE: label=Bloor−Darnforth : color =0,150,35: adjlist=Coxwell ,
    Woodbine
EDGE: label=Bloor−Darnforth : color =0,150,35: adjlist=Woodbine ,
    Main Street
EDGE: label=Bloor−Darnforth : color =0,150,35: adjlist=Main Street ,
    Victoria Park
EDGE: label=Bloor−Darnforth : color =0,150,35: adjlist=Victoria Park
    , Warden
EDGE: label=Bloor−Darnforth : color =0,150,35: adjlist=Warden ,
    Kennedy
EDGE: label=Yonge−University −Spadina : color =255,255,0: adjlist=
    Downsview , Wilson
EDGE: label=Yonge−University −Spadina : color =255,255,0: adjlist=
    Wilson , Yorkdale
EDGE: label=Yonge−University −Spadina : color =255,255,0: adjlist=
    Yorkdale , Eginton West
EDGE: label=Yonge−University −Spadina : color =255,255,0: adjlist=
    Eginton West, Glencairn
EDGE: label=Yonge−University −Spadina : color =255,255,0: adjlist=
    Glencairn , Eglington West
EDGE: label=Yonge−University −Spadina : color =255,255,0: adjlist=
    Eglington West, St Clair West
EDGE: label=Yonge−University −Spadina : color =255,255,0: adjlist=St
    Clair West, Dupont
EDGE: label=Yonge−University −Spadina : color =255,255,0: adjlist=
    Dupont , Spadina

EDGE: label=Yonge−University−Spadina: color=255,255,0: adjlist=
    Spadina, St George
EDGE: label=Yonge−University−Spadina: color=255,255,0: adjlist=St
    George, Museum
EDGE: label=Yonge−University−Spadina: color=255,255,0: adjlist=
    Museum, Queen's Park
EDGE: label=Yonge−University−Spadina: color=255,255,0: adjlist=
    Queen's Park, St Patrick
EDGE: label=Yonge−University−Spadina: color=255,255,0: adjlist=St
    Patrick, Osgoode
EDGE: label=Yonge−University−Spadina: color=255,255,0: adjlist=
    Osgoode, St Andrew
EDGE: label=Yonge−University−Spadina: color=255,255,0: adjlist=St
    Andrew, Union
EDGE: label=Yonge−University−Spadina: color=255,255,0: adjlist=
    Union, King
EDGE: label=Yonge−University−Spadina: color=255,255,0: adjlist=
    King, Queen
EDGE: label=Yonge−University−Spadina: color=255,255,0: adjlist=
    Queen, Dundas
EDGE: label=Yonge−University−Spadina: color=255,255,0: adjlist=
    Dundas, College
EDGE: label=Yonge−University−Spadina: color=255,255,0: adjlist=
    College, Wellesley
EDGE: label=Yonge−University−Spadina: color=255,255,0: adjlist=
    Wellesley, Bloor−Yonge
EDGE: label=Yonge−University−Spadina: color=255,255,0: adjlist=
    Bloor−Yonge, Rosedale
EDGE: label=Yonge−University−Spadina: color=255,255,0: adjlist=
    Rosedale, Summerhill
EDGE: label=Yonge−University−Spadina: color=255,255,0: adjlist=
    Summerhill, St Clair
EDGE: label=Yonge−University−Spadina: color=255,255,0: adjlist=St
    Clair, Davisville
EDGE: label=Yonge−University−Spadina: color=255,255,0: adjlist=
    Davisville, Eglington
EDGE: label=Yonge−University−Spadina: color=255,255,0: adjlist=
    Eglington, Lawrence
EDGE: label=Yonge−University−Spadina: color=255,255,0: adjlist=
    Lawrence, York Mills
EDGE: label=Yonge−University−Spadina: color=255,255,0: adjlist=
    York Mills, Sheppard−Yonge
EDGE: label=Yonge−University−Spadina: color=255,255,0: adjlist=
    Sheppard−Yonge, North York Centre
EDGE: label=Yonge−University−Spadina: color=255,255,0: adjlist=
    North York Centre, Finch
EDGE: label=Scarborough RT: color=0,131,212: adjlist=Kennedy,
    Lawrence East

EDGE: label=Scarborough RT: color=0,131,212: adjlist=Lawrence East
, Ellesmere
EDGE: label=Scarborough RT: color=0,131,212: adjlist=Ellesmere,
Midland
EDGE: label=Scarborough RT: color=0,131,212: adjlist=Midland,
Scarborough Centre
EDGE: label=Scarborough RT: color=0,131,212: adjlist=Scarborough
Centre, McCowan
EDGE: label=Sheppard: color=123,0,95: adjlist=Sheppard−Yonge,
Bayview
EDGE: label=Sheppard: color=123,0,95: adjlist=Bayview, Bessarion
EDGE: label=Sheppard: color=123,0,95: adjlist=Bessarion, Leslie
EDGE: label=Sheppard: color=123,0,95: adjlist=Leslie, Don Mills

B.12   Washington

NODE: label=Gallery Pl−Chinatown: x=−77.0214:y=38.8983
NODE: label=White Flint: x=−77.1130:y=39.0482
NODE: label=Arlington Cemetery: x=−77.0627:y=38.8845
NODE: label=Pentagon: x=−77.0536:y=38.8694
NODE: label=Virginia Sq−GMU: x=−77.1037:y=38.8830
NODE: label=Shady Grove: x=−77.1647:y=39.1199
NODE: label=Court House: x=−77.0835:y=38.8915
NODE: label=Clarendon: x=−77.0964:y=38.8861
NODE: label=Vienna/Fairfax−GMU: x=−77.2713:y=38.8778
NODE: label=Dunn Loring−Merrifield: x=−77.2287:y=38.8832
NODE: label=West Falls Church−VT/UVA: x=−77.1888:y=38.9008
NODE: label=East Falls Church: x=−77.1565:y=38.8861
NODE: label=Baliston−MU: x=−77.1111:y=38.8825
NODE: label=Cleveland Park: x=−77.0578:y=38.9348
NODE: label=Van Ness−UDC: x=−77.0631:y=38.9436
NODE: label=Tenleytown−AU: x=−77.0793:y=38.9478
NODE: label=Medical Center: x=−77.0969:y=38.9999
NODE: label=Grosvenor−Strathmore: x=−77.1039:y=39.0293
NODE: label=Twinbrook: x=−77.1208:y=39.0624
NODE: label=Rockville: x=−77.1460:y=39.0846
NODE: label=Friendship Heights: x=−77.0853:y=38.9598
NODE: label=Bethesda: x=−77.0942:y=38.9845
NODE: label=Metro Center: x=−77.0272:y=38.8987
NODE: label=Judiciary Sq: x=−77.0163:y=38.8962
NODE: label=Union Station: x=−77.0063:y=38.8981
NODE: label=Dupont Circle: x=−77.0429:y=38.9098
NODE: label=Woodley Park−Zoo/Adams Morgan: x=−77.0524:y=38.9249
NODE: label=McPherson Sq: x=−77.0332:y=38.9014
NODE: label=Farragut North: x=−77.0395:y=38.9033
NODE: label=Farragut West: x=−77.0394:y=38.9013
NODE: label=Foggy Bottom−GWU: x=−77.0500:y=38.9007
NODE: label=Rosslyn: x=−77.0718:y=38.8953

NODE: label=Rhode Island Ave:x=−76.9957:y=38.9208
NODE: label=Brookland−CUA:x=−76.9945:y=38.9331
NODE: label=Fort Totten:x=−77.0018:y=38.9518
NODE: label=Mt Vernon Sq/7th St−Convention Center:x=−77.0218:y
    =38.9056
NODE: label=Shaw−Howard U:x=−77.0219:y=38.9129
NODE: label=U St/African−American Civil War Memorial/Cardozo:x
    =−77.0284:y=38.9165
NODE: label=Columbia Heights:x=−77.0325:y=38.9287
NODE: label=Georgia Ave−Petworth:x=−77.0245:y=38.9361
NODE: label=Takoma:x=−77.0175:y=38.9753
NODE: label=Silver Spring:x=−77.0267:y=38.9895
NODE: label=Forest Glen:x=−77.0427:y=39.0155
NODE: label=Wheaton:x=−77.0508:y=39.0386
NODE: label=Glenmont:x=−77.0530:y=39.0616
NODE: label=West Hyattsville:x=−76.9694:y=38.9550
NODE: label=Prince George's Plaza:x=−76.9561:y=38.9652
NODE: label=College Park−U of Md:x=−76.9282:y=38.9785
NODE: label=Greenbelt:x=−76.9111:y=39.0111
NODE: label=Federal Triangle:x=−77.0272:y=38.8987
NODE: label=Smithsonian:x=−77.0279:y=38.8880
NODE: label=L'Enfant Plaza:x=−77.0216:y=38.8849
NODE: label=Archives−Navy Memorial:x=−77.0215:y=38.8939
NODE: label=Federal Center SW:x=−77.0156:y=38.8849
NODE: label=Capitol South:x=−77.0048:y=38.8850
NODE: label=Eastern Market:x=−76.9952:y=38.8841
NODE: label=Potomac Ave:x=−76.9853:y=38.8808
NODE: label=Stadium−Armory:x=−76.9771:y=38.8860
NODE: label=Benning Road:x=−76.9379:y=38.8903
NODE: label=Capitol Heights:x=−76.9132:y=38.8895
NODE: label=Minnesota Ave:x=−76.9476:y=38.8980
NODE: label=Deanwood:x=−76.9355:y=38.9080
NODE: label=Cheverly:x=−76.9151:y=38.9165
NODE: label=Landover:x=−76.8900:y=38.9339
NODE: label=New Carroliton:x=−76.8718:y=38.9483
NODE: label=Addison Road−Seat Pleasant:x=−76.8932:y=38.8866
NODE: label=Waterfront−SEU:x=−77.0170:y=38.8764
NODE: label=Navy Yard:x=−77.0047:y=38.8765
NODE: label=Anacostia:x=−76.9954:y=38.8620
NODE: label=Branch Ave:x=−76.9117:y=38.8270
NODE: label=Congress Heights:x=−76.9879:y=38.8453
NODE: label=Suitland:x=−76.9315:y=38.8437
NODE: label=Naylor Road:x=−76.9564:y=38.8511
NODE: label=Southern Ave:x=−76.9749:y=38.8411
NODE: label=Pentagon City:x=−77.0590:y=38.8629
NODE: label=Crystal City:x=−77.0503:y=38.8577
NODE: label=Ronald Reagan Washington National Airport:x
    =−77.0436:y=38.8529

NODE: label=Braddock Road:x=−77.0533:y=38.8142
NODE: label=Van Dorn Street:x=−77.1293:y=38.7993
NODE: label=Franconia−Springfield:x=−77.1685:y=38.7660
NODE: label=Eisenhower Ave:x=−77.0708:y=38.8001
NODE: label=Huntington:x=−77.0750:y=38.7940
NODE: label=King Street:x=−77.0609:y=38.8064
EDGE: label=Red: color=255,0,0: adjlist=Rockville, Shady Grove
EDGE: label=Red: color=255,0,0: adjlist=Rockville, Twinbrook
EDGE: label=Red: color=255,0,0: adjlist=Twinbrook, White Flint
EDGE: label=Red: color=255,0,0: adjlist=White Flint, Grosvenor−
    Strathmore
EDGE: label=Red: color=255,0,0: adjlist=Grosvenor−Strathmore,
    Medical Center
EDGE: label=Red: color=255,0,0: adjlist=Medical Center, Bethesda
EDGE: label=Red: color=255,0,0: adjlist=Bethesda, Friendship
    Heights
EDGE: label=Red: color=255,0,0: adjlist=Friendship Heights,
    Tenleytown−AU
EDGE: label=Red: color=255,0,0: adjlist=Tenleytown−AU, Van Ness−
    UDC
EDGE: label=Red: color=255,0,0: adjlist=Van Ness−UDC, Cleveland
    Park
EDGE: label=Red: color=255,0,0: adjlist=Cleveland Park, Woodley
    Park−Zoo/Adams Morgan
EDGE: label=Red: color=255,0,0: adjlist=Woodley Park−Zoo/Adams
    Morgan, Dupont Circle
EDGE: label=Red: color=255,0,0: adjlist=Farragut North, Dupont
    Circle
EDGE: label=Red: color=255,0,0: adjlist=Farragut North, Metro
    Center
EDGE: label=Red: color=255,0,0: adjlist=Metro Center, Gallery Pl−
    Chinatown
EDGE: label=Red: color=255,0,0: adjlist=Gallery Pl−Chinatown,
    Judiciary Sq
EDGE: label=Red: color=255,0,0: adjlist=Union Station, Judiciary
    Sq
EDGE: label=Red: color=255,0,0: adjlist=Rhode Island Ave, Union
    Station
EDGE: label=Red: color=255,0,0: adjlist=Rhode Island Ave,
    Brookland−CUA
EDGE: label=Red: color=255,0,0: adjlist=Brookland−CUA, Fort Totten
EDGE: label=Red: color=255,0,0: adjlist=Takoma, Fort Totten
EDGE: label=Red: color=255,0,0: adjlist=Silver Spring, Takoma
EDGE: label=Red: color=255,0,0: adjlist=Forest Glen, Silver Spring
EDGE: label=Red: color=255,0,0: adjlist=Wheaton, Forest Glen
EDGE: label=Red: color=255,0,0: adjlist=Wheaton, Glenmont
EDGE: label=Orange: color=255,190,40: adjlist=Dunn Loring−
    Merrifield, Vienna/Fairfax−GMU

EDGE: label=Orange: color =255,190,40: adjlist=Dunn Loring−
    Merrifield , West Falls Church−VT/UVA
EDGE: label=Orange: color =255,190,40: adjlist=West Falls Church−VT
    /UVA, East Falls Church
EDGE: label=Orange: color =255,190,40: adjlist=East Falls Church ,
    Baliston −MU
EDGE: label=Orange: color =255,190,40: adjlist=Baliston −MU,
    Virginia Sq−GMU
EDGE: label=Orange: color =255,190,40: adjlist=Virginia Sq−GMU,
    Clarendon
EDGE: label=Orange: color =255,190,40: adjlist=Court House ,
    Clarendon
EDGE: label=Orange: color =255,190,40: adjlist=Court House , Rosslyn
EDGE: label=Orange: color =255,190,40: adjlist=Rosslyn , Foggy
    Bottom−GWU
EDGE: label=Orange: color =255,190,40: adjlist=Farragut West , Foggy
    Bottom−GWU
EDGE: label=Orange: color =255,190,40: adjlist=Farragut West ,
    McPherson Sq
EDGE: label=Orange: color =255,190,40: adjlist=McPherson Sq , Metro
    Center
EDGE: label=Orange: color =255,190,40: adjlist=Federal Triangle ,
    Metro Center
EDGE: label=Orange: color =255,190,40: adjlist=Smithsonian , Federal
    Triangle
EDGE: label=Orange: color =255,190,40: adjlist=L'Enfant Plaza ,
    Smithsonian
EDGE: label=Orange: color =255,190,40: adjlist=Federal Center SW, L
    'Enfant Plaza
EDGE: label=Orange: color =255,190,40: adjlist=Federal Center SW,
    Capitol South
EDGE: label=Orange: color =255,190,40: adjlist=Capitol South ,
    Eastern Market
EDGE: label=Orange: color =255,190,40: adjlist=Potomac Ave, Eastern
    Market
EDGE: label=Orange: color =255,190,40: adjlist=Stadium−Armory,
    Potomac Ave
EDGE: label=Orange: color =255,190,40: adjlist=Minnesota Ave,
    Stadium−Armory
EDGE: label=Orange: color =255,190,40: adjlist=Deanwood , Minnesota
    Ave
EDGE: label=Orange: color =255,190,40: adjlist=Cheverly , Deanwood
EDGE: label=Orange: color =255,190,40: adjlist=Landover , Cheverly
EDGE: label=Orange: color =255,190,40: adjlist=Landover , New
    Carroliton
EDGE: label=Blue: color =0,0,255: adjlist=Van Dorn Street ,
    Franconia−Springfield

EDGE: label=Blue: color=0,0,255: adjlist=Van Dorn Street, King
    Street
EDGE: label=Blue: color=0,0,255: adjlist=King Street, Braddock
    Road
EDGE: label=Blue: color=0,0,255: adjlist=Ronald Reagan Washington
    National Airport, Braddock Road
EDGE: label=Blue: color=0,0,255: adjlist=Ronald Reagan Washington
    National Airport, Crystal City
EDGE: label=Blue: color=0,0,255: adjlist=Crystal City, Pentagon
    City
EDGE: label=Blue: color=0,0,255: adjlist=Pentagon City, Pentagon
EDGE: label=Blue: color=0,0,255: adjlist=Pentagon, Arlington
    Cemetery
EDGE: label=Blue: color=0,0,255: adjlist=Arlington Cemetery,
    Rosslyn
EDGE: label=Blue: color=0,0,255: adjlist=Foggy Bottom–GWU, Rosslyn
EDGE: label=Blue: color=0,0,255: adjlist=Foggy Bottom–GWU,
    Farragut West
EDGE: label=Blue: color=0,0,255: adjlist=McPherson Sq, Farragut
    West
EDGE: label=Blue: color=0,0,255: adjlist=Metro Center, McPherson
    Sq
EDGE: label=Blue: color=0,0,255: adjlist=Metro Center, Federal
    Triangle
EDGE: label=Blue: color=0,0,255: adjlist=Federal Triangle,
    Smithsonian
EDGE: label=Blue: color=0,0,255: adjlist=Smithsonian, L'Enfant
    Plaza
EDGE: label=Blue: color=0,0,255: adjlist=L'Enfant Plaza, Federal
    Center SW
EDGE: label=Blue: color=0,0,255: adjlist=Capitol South, Federal
    Center SW
EDGE: label=Blue: color=0,0,255: adjlist=Eastern Market, Capitol
    South
EDGE: label=Blue: color=0,0,255: adjlist=Eastern Market, Potomac
    Ave
EDGE: label=Blue: color=0,0,255: adjlist=Potomac Ave, Stadium−
    Armory
EDGE: label=Blue: color=0,0,255: adjlist=Benning Road, Stadium−
    Armory
EDGE: label=Blue: color=0,0,255: adjlist=Capitol Heights, Benning
    Road
EDGE: label=Blue: color=0,0,255: adjlist=Addison Road−Seat
    Pleasant, Capitol Heights
EDGE: label=Yellow: color=255,255,0: adjlist=Eisenhower Ave,
    Huntington
EDGE: label=Yellow: color=255,255,0: adjlist=Eisenhower Ave, King
    Street

EDGE: label=Yellow: color =255,255,0: adjlist=Braddock Road, King
    Street
EDGE: label=Yellow: color =255,255,0: adjlist=Braddock Road, Ronald
    Reagan Washington National Airport
EDGE: label=Yellow: color =255,255,0: adjlist=Crystal City, Ronald
    Reagan Washington National Airport
EDGE: label=Yellow: color =255,255,0: adjlist=Pentagon City,
    Crystal City
EDGE: label=Yellow: color =255,255,0: adjlist=Pentagon, Pentagon
    City
EDGE: label=Yellow: color =255,255,0: adjlist=Pentagon, L'Enfant
    Plaza
EDGE: label=Yellow: color =255,255,0: adjlist=Archives−Navy
    Memorial, L'Enfant Plaza
EDGE: label=Yellow: color =255,255,0: adjlist=Gallery Pl−Chinatown,
    Archives−Navy Memorial
EDGE: label=Yellow: color =255,255,0: adjlist=Mt Vernon Sq/7th St−
    Convention Center, Gallery Pl−Chinatown
EDGE: label=Green: color =20,150,0: adjlist=Suitland, Branch Ave
EDGE: label=Green: color =20,150,0: adjlist=Suitland, Naylor Road
EDGE: label=Green: color =20,150,0: adjlist=Naylor Road, Southern
    Ave
EDGE: label=Green: color =20,150,0: adjlist=Southern Ave, Congress
    Heights
EDGE: label=Green: color =20,150,0: adjlist=Congress Heights,
    Anacostia
EDGE: label=Green: color =20,150,0: adjlist=Anacostia, Navy Yard
EDGE: label=Green: color =20,150,0: adjlist=Navy Yard, Waterfront−
    SEU
EDGE: label=Green: color =20,150,0: adjlist=Waterfront−SEU, L'
    Enfant Plaza
EDGE: label=Green: color =20,150,0: adjlist=L'Enfant Plaza,
    Archives−Navy Memorial
EDGE: label=Green: color =20,150,0: adjlist=Archives−Navy Memorial,
    Gallery Pl−Chinatown
EDGE: label=Green: color =20,150,0: adjlist=Gallery Pl−Chinatown,
    Mt Vernon Sq/7th St−Convention Center
EDGE: label=Green: color =20,150,0: adjlist=Shaw−Howard U, Mt
    Vernon Sq/7th St−Convention Center
EDGE: label=Green: color =20,150,0: adjlist=U St/African−American
    Civil War Memorial/Cardozo, Shaw−Howard U
EDGE: label=Green: color =20,150,0: adjlist=Columbia Heights, U St/
    African−American Civil War Memorial/Cardozo
EDGE: label=Green: color =20,150,0: adjlist=Georgia Ave−Petworth,
    Columbia Heights
EDGE: label=Green: color =20,150,0: adjlist=Georgia Ave−Petworth,
    Fort Totten

EDGE: label=Green : color =20 ,150 ,0: adjlist=Fort Totten , West
    Hyattsville
EDGE: label=Green : color =20 ,150 ,0: adjlist=Prince George's Plaza ,
    West Hyattsville
EDGE: label=Green : color =20 ,150 ,0: adjlist=Prince George's Plaza ,
    College Park–U of Md
EDGE: label=Green : color =20 ,150 ,0: adjlist=College Park–U of Md,
    Greenbelt

Appendix C

R Code for Empirical Experiment Analysis

This Appendix contains the R code that was used to carry out the statistical analysis as part of our empirical experiment. The analysis is described in Section 7.5.

```
library(survBayes)
library(session)
library(foreign)


###############################################################
#         Reading the data
###############################################################
MapTypeP <- read.csv("MapType_P.CSV", header = TRUE, sep = ",",
    quote="\"", dec=".", fill = TRUE)
MapTypeG <- read.csv("MapType_G.CSV", header = TRUE, sep = ",",
    quote="\"", dec=".", fill = TRUE)
MapTypeA <- read.csv("MapType_A.CSV", header = TRUE, sep = ",",
    quote="\"", dec=".", fill = TRUE)

# mergging the data in a single object and defining factors
  attach(MapTypeG)        # Geo
  attach(MapTypeP)        # Pub
  attach(MapTypeA)        # Aut
  MetroMap_GPA <- rbind(MapTypeP, MapTypeA)
  MetroMap_GPA <- rbind(MetroMap_GPA, MapTypeG)

MetroMap_GPA[,"TypeG"] <- 0
MetroMap_GPA[,"TypeP"] <- 0
MetroMap_GPA[,"TypeA"] <- 0
MetroMap_GPA[MetroMap_GPA["MapType"] == "G","TypeG"] <- 1
MetroMap_GPA[MetroMap_GPA["MapType"] == "P","TypeP"] <- 1
MetroMap_GPA[MetroMap_GPA["MapType"] == "A","TypeA"] <- 1

write.csv(MetroMap_GPA, file = "MetroMap_GPA.csv", row.names =
    FALSE)
```

```
###################################################################
# COVARIATES and FRAILTIES
###################################################################
# Covariate:              1.  Map system (City)
#                         2.  MapType (releant)
#
# Clusters (frailties):   1.  Login (individuals)
###################################################################
library(survBayes)
library(session)
library(foreign)

MetroMap <- read.csv("MetroMap_GPA.csv")
head(MetroMap)
summary(MetroMap)

control <- survBayes.control(sigma.lbh.1 = 0.01, rate.sigma.lbh
    .1 = 0.001, shape.sigma.lbh.1 = 0.001)
control.frailty <- survBayes.control.gamma.frailty(mu.cl = 1,
    prec.tau.cl = 1e-04)

MetroMap.res <- survBayes(Surv(Time.s./10, CorrectCod, type = "
    right")
        ~ as.factor(TypeP) + as.factor(TypeA) + as.factor(Map) +
            frailty(Login, dist = "gamma") ,
        data = MetroMap,
        burn.in = 20000,
        number.sample = 150000)

str(MetroMap.res)

save.session("MetroMap_Out.Rda")

###################################################################
#           reading the data
###################################################################
library(survBayes)
library(session)
library(foreign)
library(psych)

MetroMap <- read.csv("MetroMap_GPA.csv")
head(MetroMap)

# Statistics by group (MapType)
describe.by(MetroMap["Time.s."], MetroMap$MapType)
```

```
# Statistics by group (MapType) with "Correct" answers
describe.by(MetroMap[MetroMap["Correct"]==TRUE,"Time.s."],
    MetroMap[MetroMap["Correct"]==TRUE,]$MapType)

# Statistics by group (MapType) with "Incorrect" answers
describe.by(MetroMap[MetroMap["Correct"]==FALSE,"Time.s."],
    MetroMap[MetroMap["Correct"]==FALSE,]$MapType)


###################################################################
#           Loading the output saved
###################################################################
library(survBayes)
library(session)
library(foreign)
library(psych)


load("MetroMap_Out.Rda")
# Summary results
str(MetroMap.res)

summary(MetroMap.res$beta)

# Bayesian estimates of proportional hazard coefficients
beta.est <- apply(MetroMap.res$beta, 2, mean)
is.table(MetroMap.res$beta)
Betas.Path <- as.table(MetroMap.res$beta)
write.csv(Betas.Path, "Betas.Path.csv")
Betas.Path <- read.csv("Betas.Path.csv", row.names = 1)
names(Betas.Path) <- c("TypeP","TypeA","Map2","Map3","Map4","
    Map5","Map6")
head(Betas.Path)
Betas.Path["TypeAvsP"] <- Betas.Path["TypeA"] - Betas.Path["
    TypeP"]
Betas.Path[Betas.Path["TypeAvsP"] < 0,"ContrastAvsP"] <- 0
Betas.Path[Betas.Path["TypeAvsP"] >= 0,"ContrastAvsP"] <- 1
Betas.Path <- as.matrix(Betas.Path)

summary(Betas.Path[,c("TypeP","TypeA","TypeAvsP","ContrastAvsP
    ")])
hist(Betas.Path[,c("TypeAvsP")], breaks = 50, col="lightblue",
    border="pink",xlab="Contrast of beta_A vs beta_P", main="")
abline(v=0,lty=2)


write.csv(Betas.Path, "Betas.Path.csv")

# Bayesian estimates of the survival function
```

```
baseline.hazard <- survBayes.baseline.hazard(MetroMap.res,
    start = 1, type = "cum", n.inter = 0)
time <- baseline.hazard$time
Lambda0 <- baseline.hazard$cum.base.haz
surv.base <- exp(-Lambda0)
plot(time, surv.base, type = "l", xlab = "Time [seconds/10]",
    ylab = "Survival function", xlim = c(0,10), lty = 1, ylim =
    c(0, 1))

save.session("MetroMap_Out.Rda")
```