Automatic Metro Map Layout Using Multicriteria Optimization

Jonathan Stott, Peter Rodgers, Juan Carlos Martínez-Ovando, and Stephen G. Walker

Abstract—This paper describes an automatic mechanism for drawing metro maps. We apply multicriteria optimization to find effective placement of stations with a good line layout and to label the map unambiguously. A number of metrics are defined, which are used in a weighted sum to find a fitness value for a layout of the map. A hill climbing optimizer is used to reduce the fitness value, and find improved map layouts. To avoid local minima, we apply clustering techniques to the map—the hill climber moves both stations and clusters when finding improved layouts. We show the method applied to a number of metro maps, and describe an empirical study that provides some quantitative evidence that automatically-drawn metro maps can help users to find routes more efficiently than either published maps or undistorted maps. Moreover, we have found that, in these cases, study subjects indicate a preference for automatically-drawn maps over the alternatives.

Index Terms—Information visualization, diagram layout, graph drawing.

1 Introduction

Since Harry Beck developed the iconic map of the London Underground, first published in 1933 [1], [2], similar schematic diagrams have been used to guide travelers on public transport networks. Typically, such diagrams are produced by modifying the network layout so that unnecessary complexity is removed. For example, lines run at regular angles, stations are evenly spaced, and labels are placed in unambiguous locations. While the geometry of the map is changed, the topology is retained. The great advantage of such diagrams over undistorted maps is that they simplify the key tasks of route planning and navigation for travelers.

Currently, schematic diagrams are produced by human designers and take a considerable time to generate. While this may be sensible for the use of such maps as static views of entire networks, there are other applications for which automated layout would be a great benefit. Cheap and quickly produced computer-generated schematic maps might be used for personal travel plans, generating diagrams for networks not currently included in current maps, or for using schematic networks in other application areas, such as water or gas utility networks [3]. In addition, the metro map metaphor, which draws graph-based data in a similar form to metro maps, has been widely used in nongeographical application areas, where the lack of geographical constraints allow more freedom in diagram layout. Such applications include project plans [4], cancer pathways [5], and Web site mapping [6].

Our paper describes a mechanism for drawing usable metro maps. This is achieved with a new method for metro map layout, multicriteria optimization, which performs the difficult task of generating a good line layout with unambiguous, readable labels. A number of metrics are defined, which are used in a weighted sum to attempt to measure the esthetic quality of the diagram. Our approach also uses three new clustering mechanisms to avoid local minima. The usability of metro maps produced by our system has been tested by empirical study and we describe the experiments and statistical analysis that brings us to the conclusion that our metro map layout method can produce usable diagrams. Previous conference publications [7], [8] have described early versions of the method that appears in Section 3. However, the work in this section is extended from the early publications, with improved criteria and optimization. The clustering and empirical research is new work and has not been previously published. The work given here formed the basis of a PhD thesis [9].

The rest of this paper is organized as follows: Section 2 describes some background in the layout of metro maps and other schematic diagrams; Section 3 describes our optimization method; Section 4 describes how the optimizer is extended with clustering methods; Section 5 gives some examples of metro maps drawn with the system; Section 6 describes the empirical study and gives our interpretation of results; finally, Section 7 gives our conclusions and suggests further work.

2 Related Work

Other approaches to metro map layout have not yet been empirically tested for their effectiveness in drawing real-world examples of metro maps. These methods, in the broad, attempt to achieve similar layouts to those we attempt, with similar diagram criteria either implicitly or

- *J. Stott and P. Rodgers are with the School of Computing, University of Kent, Canterbury, Kent CT2 7NF, UK. E-mail: jonathan.stott@axonuk.com, P.J.Rodgers@kent.ac.uk.*
- *J.C. Martínez-Ovando and S.G. Walker are with the School of Mathematics, Statistics and Actuarial Science, University of Kent, Canterbury, Kent CT2 7NF, UK. E-mail: jc.martinez.ovando@gmail.com, S.G.Walker@kent.ac.uk.*

Manuscript received 12 Dec. 2008; revised 3 June 2009; accepted 27 Oct. 2009; published online 28 Jan. 2010.
Recommended for acceptance by M. Chen.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org, and reference IEEECS Log Number TVCG-2008-12-0202.
Digital Object Identifier no. 10.1109/TVCG.2010.24.

explicitly specified. However, unlike our method, labeling is typically not attempted at all or performed after the diagram layout has been completed.

The first attempt at automatic metro map layout was by Hong et al. who use a force-directed approach to laying out metro maps [10], [11]. They use combinations of different force-based algorithms applied sequentially. Labeling occurs after the diagram layout, but because this does not allow the map layout to make room for labels, occlusion and ambiguity can still occur. Other problems exist because they do not consider the geography of the map, resulting in counterintuitive layouts, for example, stations geographically to the north of others can be placed to the south. In practice, the implicit definition of criteria through the force method tends to result in irregular spacing of stations and nondiscrete line angles typically appear, which are not a feature of most published schematic metro maps.

Nöllenburg and Wolff describe a method of drawing metro maps using mixed-integer linear programming [12] which extends linear programming by introducing the notion of constraining variables to be within certain discrete integer ranges. Constraints include octilinear lines (horizontal, vertical, or 45 degree diagonal), maintaining a minimum line length, minimum line bends, and minimum total line length. The final diagrams lack labels. Note that both the work of Nöllenburg and Wolff and that of Hong, outlined in the previous paragraph, use variants of the Sydney metro map as examples, allowing for comparison with the Sydney map generated by the system described in this paper (see Fig. 22).

Merrick and Gudmundsson describe a path simplification method which restricts the number of directions that lines can take [13]. They simplify the lines in order of importance, determined using a heuristic function based on the number of interchange stations on the line. The method fails to maintain topology, does not achieve an effective overall structure, and lacks labeling; however, it produces results in a relatively quick time. Other research includes efforts by Bekos et al. to minimize line crossings for embedded metro maps [14], which they achieve for restricted types of diagrams.

In addition, there are similar generalization and schematic problems for application areas other than metro maps. Agrawala and Stolte describe a simulated annealing system for producing simplified route maps [15]. This work uses criteria of: length generalization to even out route segments; angle generalization to prevent small turning angles; and shape generalization to simplify the shape of roads. Avelar and Huber [16] show a similar method but model their route maps on the characteristics of public transport networks.

Casakin et al. [17] provide a taxonomy of various aspects of schematic route maps (particularly intersections), and use their taxonomy to provide an empirical assessment of schematic graphs. Yates and Humphries [18] 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. [19] 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 stations and not moving stations. Lauther and Stübinger [20] present a demonstration of software which is capable of laying out schematic diagrams using a force-directed approach with the aim of visualizing cable plans schematically.

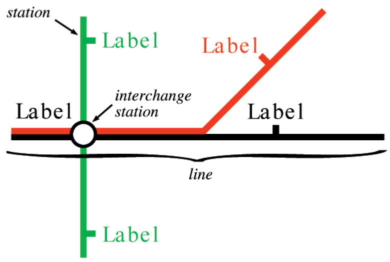

Fig. 1. Metro map features.

The methods for route map schematic layout are generally successful for the problems that they try to solve. However, the size of diagram is generally smaller and less complex than would be expected for a metro map.

3 Optimization Method

This section describes the basic concepts of our metro map layout method. We detail the hill climbing optimizer, the criteria measured, and the method for combining criteria to produce a fitness value. This section only discusses individual station movements; Section 4 discusses the movement of clusters of stations.

A *diagram*, G, is a set of *stations*, V, with connections between pairs of stations represented by a set of edges, E. When drawing metro maps, we use an *edge* to represent a single connection between two stations. In some cases, there may be several edges connecting two stations 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 color that identifies which line they are part of. These features are illustrated in Fig. 1.

The diagram is embedded on an integer grid, meaning that stations must be centered on grid intersections; however, edges do not have to follow grid lines. The spacing between adjacent intersections in the grid is denoted by g and is always large enough to allow parallel edges between stations to be placed without ambiguity. Making the search space discrete in this manner allows us to dramatically reduce the number of potential locations for stations. We also define a preferred multiple of the grid spacing for station separation, l, making the ideal edge length, lg.

The method has been tested on nine real world maps to date, which can be seen in [9], in addition to a number of diagrams constructed to test particular issues. See also the Appendices of this paper for examples. The number of real world test maps is restricted by the difficulty of getting data, as each undistorted map must be encoded by hand. The criteria we selected includes criteria that empirical research suggests are effective in the field of graph layout. Not all such criteria are appropriate for metro map layout (such as symmetry). In addition, the line straightness and balanced edge length criteria have been added for the particular requirements of metro map layout. The clustering mechanisms were based on informal examination of the output while the system was being developed.

3.1 Hill Climbing

Multicriteria optimization has been used previously in graph drawing [21], [22]. These previous methods use genetic algorithms or simulated annealing to optimize a fitness function; however, we found that a simpler method using hill climbing was more appropriate for this application. Simulated annealing adds an element of nondeterminism in order to escape from local minima in the search space, but a larger number of iterations would be necessary to reach a minimum in the search space. Moreover, the local minima that typically occur in schematic networks are better dealt with by clustering. Genetic algorithms are also nondeterministic, and converge more slowly than hill climbers or simulated annealers.

In outline, our method operates in this manner: first, we find an initial layout for the stations, which is the undistorted layout, but with the stations snapped to the grid and only one station at any point. For each station in the diagram, we calculate the fitness of the diagram. We then search the points around a rectangle centered on the station at a given distance. We then move the station to the location that most improves the fitness. If none do, then the station is not moved. This is performed for all stations in the diagram. We then see if the label layout in the diagram can be improved in a similar manner. Once all station positions and label positions have been tested, the process is repeated until no more improvement can be made. On each of these repetitions, a cooling factor reduces the search rectangle to allow fine tuning of layout as the search progresses. The process in detail, including the clustering step, discussed in Section 4, is given in Algorithm 1.

Algorithm 1. Metro Map Layout

```
G ⇐ (V, E, L)
snapStations(V)
m_T0 ⇐ calcStationCriteria(V) + calcLabelCriteria(L)
running ⇐ true
while running do
    for v ∈ V do
        m_N0 ⇐ calcStationCriteria(V)
        m_N ⇐ findLowestStationCriteria(V)
        if m_N < m_N0 then
            moveStation(v)
        end if
    end for
    P ⇐ clusterOverlengthEdges(V, E) ∪
    clusterBends(V, E) ∪ clusterPartitions(V, E)
    for p ∈ P do
        m_N0 ⇐ calcStationCriteria(V)
        m_N ⇐ findLowestStationCriteria(V)
        if m_N < m_N0 then
            moveCluster(p)
        end if
    end for
    for l ∈ L do
        m_L0 ⇐ calcLabelCriteria(L)
        m_L ⇐ findLowestLabelCriteria(L)
        if m_L < m_L0 then
            moveLabel(l)
        end if
    end for
    m_T ⇐ calcStationCriteria(V) + calcLabelCriteria(L)
    if m_T ≮ m_T0 then
        running ⇐ false
    else
        m_T0 ⇐ m_T
    end if
end while
```

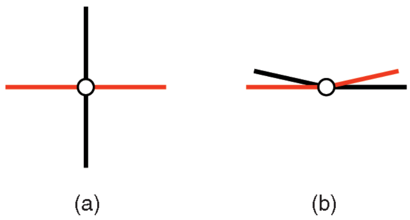

Fig. 2. Examples of (a) optimal angular resolution and (b) poor angular resolution.

3.2 Station Criteria

Movement of stations depends on the calculation of the weighted sum of several criteria which are judged to affect the esthetic quality of the map. Our basis for the selection of criteria comes from existing research that evaluates esthetic criteria in relation to graph drawing [23] as well as criteria considered specific to the esthetics of schematic diagrams and metro maps [10], [17], [24], [25]. The criteria evaluate to a lower value when improved. The station criteria are:

- **Angular Resolution Criterion,** c_{N1}**.** The angles of incident edges at each station should be maximized, because if there is only a small angle between any two adjacent edges, then it can become difficult to distinguish between them. See Fig. 2. It is calculated by

$$c_{N1} = \sum_{v \in V} \sum_{\{e_1, e_2\} \in E_v} \left| \frac{2\pi}{\rho(v)} - \theta(e_1, e_2) \right|, \tag{1}$$

  where $\rho(v)$ is the degree of the station v (the degree of a station is the count of its incident edges) and $\theta(e_1, e_2)$ is the angle in radians between two adjacent edges e_1 and e_2 incident to v.
- **Edge Length Criterion,** c_{N2}**.** The edge lengths across the whole map should be approximately equal to ensure regular spacing between stations. It is based on the preferred multiple, l, of the grid spacing, g. The purpose of the criterion is to penalize edges that are longer than or shorter than lg. It is calculated by

$$c_{N2} = \sum_{e \in E} \left| \frac{|e|}{lg} - 1 \right|, \tag{2}$$

  where $|e|$ is the length of edge e.

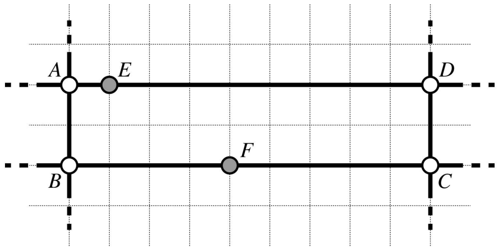

Fig. 3. Balanced edge lengths.

- **Balanced Edge Length Criterion,** c_{N3}**.** The length of edges incident to a particular station should be similar. One of the characteristics of metro maps is that there are many stations with two incident edges (degree two). Fig. 3 shows an example whereby there are two stations, E and F, with degree two. If we are only considering the edge length criterion for these two stations, it evaluates to the same value for both stations. However, we want to ensure that the edge lengths are similar. In these cases, the balanced edge length criterion can help by penalizing stations with degree two that have incident edges with unbalanced lengths. It is calculated as the sum of the absolute difference between the lengths of the two incident edges of every degree two station in the diagram

$$c_{N3} = \sum_{v \in V, \rho(v)=2} ||e_1| - |e_2||, \tag{3}$$

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

- **Line Straightness Criterion,** c_{N4}**.** Edges that form part of a line should, where possible, be collinear either side of each station that the line passes through. One of the important features of metro maps is that metro lines appear to pass through stations so that the entry edge is more-or-less directly opposite the exit edge. This is particularly important if there are two or more lines passing through a station, see Fig. 4. It is calculated by

$$c_{N4} = \sum_{(v \in V)} \left(\sum_{e_1, e_2 \in E} \theta(e_1, e_2) \right), \tag{4}$$

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 edges of the same line that are incident to the station v.

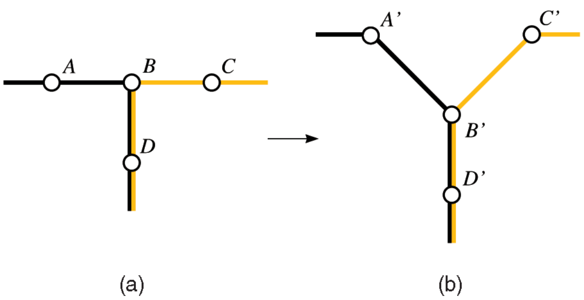

Fig. 4. Examples of (a) poor line straightness and (b) improved line straightness.

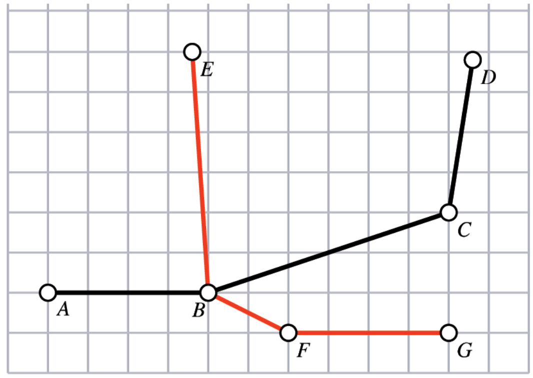

Fig. 5. Octilinearity example.

- **Octilinearity Criterion,** c_{N5}**.** Each edge should be drawn horizontally, vertically, or diagonally at 45 degree, so we penalize edges that are not at a desired angle. It is calculated by

$$c_{N5} = \sum_{\{u,v\} \in E} \left| \sin 4 \left(\tan^{-1} \frac{|y(u) - y(v)|}{|x(u) - x(v)|} \right) \right|, \tag{5}$$

where $\{u, v\}$ is an edge between stations u and v, and $y(v)$ and $x(v)$ are the y- and x-coordinates of station v, respectively.

Fig. 5 shows an example that illustrates the octilinearity criterion. The result of calculating the criterion for each edge in the example graph is shown in Table 1. As is expected, edges which are already at an angle of some multiple of 45 degree (AB and FG) evaluate to zero, whereas edges which are at angles furthest from multiples of 45 degree evaluate to the highest values. Edges BC and BF evaluate to the same value because they are both 18.43 degree away from the nearest multiple of 45 degree.

3.3 Station Rules

As well as the above five criteria, we have implemented four station movement rules which are strictly enforced during the layout process. We apply rules in addition to criteria to enforce particularly important features of a schematic diagram. The four station movement rules are:

- **Bounding Area Restriction Rule.** Restrict the movement of stations to be within a certain bounding area so that the final diagram will fit on the target

TABLE 1
Examples of Octilinearity Criterion Calculations

Edge, $e = \{u, v\}$	c^e_{N5}
$\{A, B\}$	$\left|\sin 4 \left(\tan^{-1} \frac{0}{4}\right)\right| = 0$
$\{B, C\}$	$\left|\sin 4 \left(\tan^{-1} \frac{2}{6}\right)\right| = 0.96$
$\{C, D\}$	$\left|\sin 4 \left(\tan^{-1} \frac{3.8}{0.6}\right)\right| = 0.586$
$\{B, E\}$	$\left|\sin 4 \left(\tan^{-1} \frac{6}{0.6}\right)\right| = 0.388$
$\{B, F\}$	$\left|\sin 4 \left(\tan^{-1} \frac{1}{2}\right)\right| = 0.96$
$\{F, G\}$	$\left|\sin 4 \left(\tan^{-1} \frac{0}{4}\right)\right| = 0$

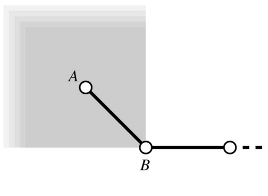

Fig. 6. Example of the enforcement of the relative positions when moving a station. The gray-shaded area shows the degree of freedom afforded to (a) station A and (b) station C.

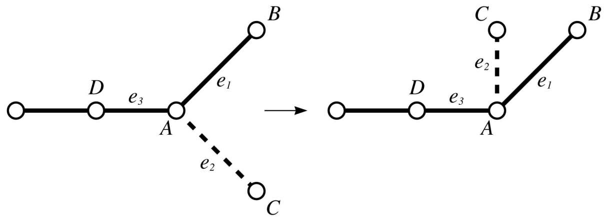

Fig. 7. Preservation of edge ordering. Without preserving the ordering of edges, station C would be able to move as shown, changing the topology of the map.

display. When multiple stations are moved, no movement may place any of the stations beyond the bounding area.

- **Relative Position Rule.** Enforce the relative position between adjacent stations. Although metro maps are a generalization of the undistorted network, relationships such as one station being north of another station are still important to the usability of the drawn map. This rule ensures that the relative positions between neighboring stations do not change. Fig. 6 illustrates the effect of enforcing relative positions. Stations may move only within the quadrant in which they start. There are four possible quadrants, defined by the division of the plane by two orthogonal axes centered on the relevant neighboring station. If the station to be moved starts on the border of two quadrants (because it is horizontally or vertically aligned with the neighbor), then it may move in either quadrant.
- **Occlusions Rule.** Avoid the introduction of occlusions of other edges and stations to ensure that a station is not moved so that it is not lying on top of any other station or edge, and that edges do not cross other edges or lie on top of any other station.
- **Edge Ordering Rule.** Preserve the ordering of edges incident to a station. The relative positions rule allows us to restrict the relative positions between two stations. However, there are limitations to this rule that mean that the topology of the diagram could be changed by the movement of a station (see Fig. 7). To implement this rule, we need to find the clockwise ordering of edges around the station being moved and any neighboring station in the diagram. At each potential new location for a station, the edge ordering is checked and the location disregarded if the orderings change.

3.4 Station Movement Criteria Weightings

Each criterion has 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 (normalized). It is not possible to bound all criteria to upper and lower values, then scale to between 0 and 1, because many of the station criteria are unbounded. However, it is still 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, down to zero, if it is not appropriate for that particular case.

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

$$m_N = \sum_{i=1}^{5} w_{Ni} c_{Ni}, \tag{6}$$

where w_{Ni} is the weighting for criterion c_{Ni}. 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 normalized (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. The process was driven by the subjective judgement of the investigators, as a more formal approach would have taken more time than was available.

Appendix A, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TVCG.2010.24, shows the effect of removing each station movement criterion in turn, indicating that all criteria have some effect on the final result.

3.5 Label Criteria

Labeling is an integral part of metro maps—in an informal discussion with a professional metro map designer, we discovered that labeling was considered the major issue in the layout of the London Underground map. Hence, it should form an integral part of any schematics layout method. Similarly, with station and line optimization, we have designed a number of criteria for label placement. These criteria are based on cartographic point labeling considerations [26]. However, it should be noted that some principles differ in metro map layout, in particular, positions directly to the left and right of the station are acceptable in metro map layout because the line prevents the station being misinterpreted as a type character in the label. The advantage of calculating a fitness function for labeling, rather than applying alternative, more widely used methods, is that the labeling can be integrated with the station layout in the hill climber.

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 labeling space. Fig. 8 shows our chosen labeling space, which allows eight different label positions. The values for the

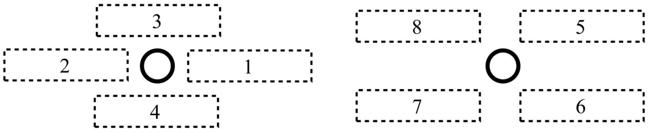

Fig. 8. Search space for labeling the metro map.

positions are currently independent of line orientation; however, they could be adjusted for particular line orientations for greater flexibility.

Occasionally, a label might contain a large amount of text with several words so we split a long label length of the label if it exceeds $0.75lg$. This is a relatively simplistic strategy, as while this usually ensures the label fits in between stations, often a split is made when there is still plenty of room for it in the diagram.

The seven labeling criteria are:

- **Label Occlusion Criteria,** c_{L1}, c_{L2}, c_{L3}**.** These three criteria count the number of stations, edges, and other labels that intersect/occlude labels, respectively. As 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 diagram would be improved if a label were allowed to occlude an edge, as in dense areas of the diagram, it may not be possible to find any improvements to the position of labels to resolve all label occlusions.
- **Label Position Criterion,** c_{L4}**.** Places a preference on label positions in the labeling space by putting a value on each position. A label can occupy any one of the eight locations in the labeling space shown in Fig. 8. Some label positions are more preferential than others, so each different position in the labeling space is assigned a value relating to the preference for that position. Table 2 shows the set of values for each position in the labeling space. The label position criterion is then defined as the sum of the position values for each label in the diagram.
- **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. This improves readability because the labels appear as a list which can be read easily rather than having to switch attention from one side of the line to the other. The criterion is only calculated for labels with one or two neighboring stations, as stations with more than two neighbors have to be consistent with more than one line. The calculation is fairly simple: for each station in the diagram with degree equal to two, a count is kept of the number of times the position of the label of an adjacent station (if that station has degree less than three) differs to the position of the current station, counting each pair of stations once. Fig. 9 shows an example of poor label position consistency where the unweighted value of the label position criterion would be three.
- **Station Proximity Criterion,** c_{L6}**.** Penalizes labels that come into close proximity to unrelated stations, so discouraging labels from being positioned too close to other unrelated stations, which causes ambiguity when deciding to which station the label relates (see Fig. 10). It is calculated by

$$c_{L6} = \sum_{k \in L} \sum_{v \in V, k_v \neq k} \frac{1}{d(k,v)^2}, \tag{7}$$

where $d(k,v)$ is a function giving the distance from the closest point on the bounding box of label k to station v. Notice that we are interested in $n \in V, k_v \neq k$, that is, all stations in the diagram except the one for which the label (k_v) is the label we are considering (k). In other words, we do not take into account the distance between a label and station that label belongs to. In practice, because most stations in the diagram will be some way from the label in question, they will contribute very little to c_{L6}. We can, therefore, approximate the contribution of stations with $d(k,v) > x$ to zero. We use a value of g for x.
- **Perpendicular Tick Criterion,** c_{L7}**.** Encourages the tick (and therefore, the position of the label) for a particular station to be perpendicular to the line.

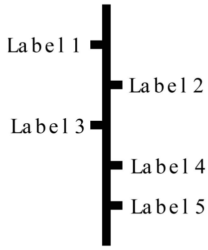

Fig. 9. Label position consistency.

TABLE 2
Label Position Values

	position	value
1	east	1.0 (best)
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 (worst)

The positions refer to the positions shown in Fig. 8.

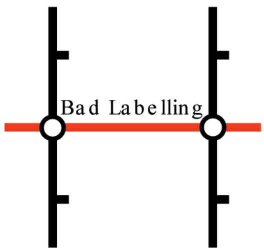

Fig. 10. An example of ambiguous labeling.

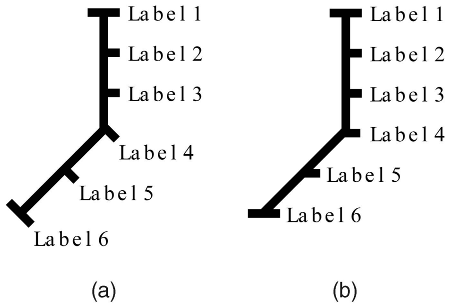

Fig. 11. Examples of (a) perpendicular tick labels and (b) nonperpendicular tick labels.

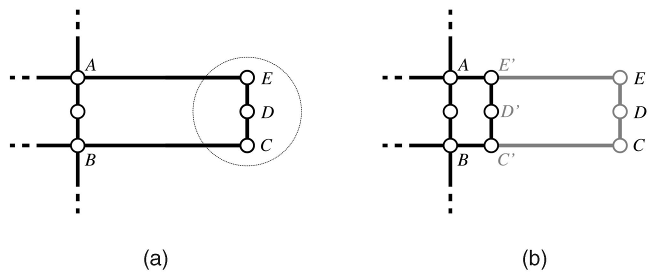

Fig. 12. Clustering multiple overlength edges. The edges AE and BC are (a) too long and it is only possible to reduce the length of these edges by moving stations C, D, and E (b) at the same time.

Fig. 11 illustrates the need for this criterion. 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). While the labels and ticks for the vertical part of the line remain the same, the perpendicular ticks on the diagonal part of the line are more prominent. The minimum distance between the line and the labels on the diagonal part is also greater when the labels are drawn diagonally, but the association with the relevant tick is not lost. For a station v, $e1_v$ and $e2_v$ are the connecting edges and $\theta1_v$ and $\theta2_v$ are the angles between the tick and $e1_v$ and $e2_v$, respectively. The unweighted value of this criterion for a single station is the absolute difference between the two angles. The total value for all stations, V, in the graph is, therefore,

$$c_{L7} = \sum_{v \in V} |\theta1_v - \theta2_v|. \tag{8}$$

3.6 Labeling Criteria Weightings

The values for label weightings were determined through trial and error with various examples in a similar manner to the way that we determined station movement criteria weightings (see Section 3.4), and as with the station criteria, the label criteria are not restricted to between 0 and 1, for consistency. However, unlike the station criteria, the label criteria can be bound by dividing by the number of stations in the diagram.

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

$$m_L = \sum_{i=1}^{7} w_{Li} c_{Li}, \tag{9}$$

where w_{Li} is the weighting for criterion c_{Li}. As with the station movement criteria weightings, the values for the label weightings can be modified by the user depending on the characteristics of the particular metro map being drawn.

4 Clustering

Section 3.1 introduced a method for laying out metro maps using multicriteria optimization that improved the fitness in the diagram by moving individual stations. This often results in easily identifiable cases of local minima that seem improvable if clustered groups of stations were moved together.

We have three methods for clustering:

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

Once clusters have been identified, they are moved in exactly the same way that individual stations are moved with the only difference being that all the stations in the cluster are moved and the relative position of stations in a cluster is maintained.

As with station movement criteria, the effect of removing each clustering method in turn is shown in Appendix A, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TVCG.2010.24, indicating that all the clustering methods have some effect on the final result.

4.1 Clustering Overlength Edges

A frequent problem we encountered when experimenting with our layout system was that of long edges that do not reduce in length. We define overlength edges as being edges which are longer than lg (the ideal edge length). Fig. 12 shows such an example with two overlength edges. If we only allow one station to move at a time, the overlength edges connecting the two groups of stations cannot reduce in length.

Our first attempt at solving this problem, given in [8], attempted to cluster groups of stations at the ends of lines. However, this does not always deal with multiple overlength edges, as shown in Fig. 12. Instead, we cluster the diagram into groups of stations connected by ideal length edges.

4.2 Clustering Nonstraight Lines

Many lines contain short deviations or kinks. This occurs when fitting a slightly off-straight line to the grid or where three stations are too close together to fit onto the grid without the middle station being offset relative to the rest of the line.

To improve nonstraight lines, we identify clusters of stations by looking at stations which have exactly one or two neighbors. Fig. 13 shows an example of cluster found with this method. This means that stations A and F are discounted from forming part of a cluster from the outset (and could

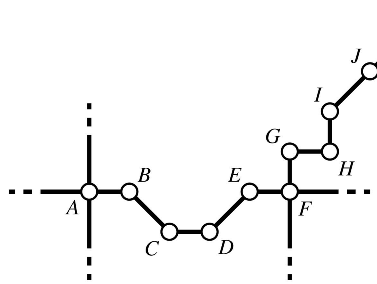

Fig. 13. Clustering stations to find nonstraight lines. Ultimately, six clusters will be identified in this graph: $\{BC\}$, $\{CD\}$, $\{DE\}$, $\{GH\}$, $\{HI\}$, and $\{IJK\}$.

even be removed from the graph while we are searching for clusters). Clusters are then identified by finding the minimum set of connected stations which are collinear.

4.3 Partitioning

The results of experiments on test maps also identified local minima that occur because overlength edges cannot always be reduced by the clustering as described in Section 4.1. An example is shown in Fig. 14. However, improvements can be made by partitioning the diagram into two along overlength edges and treating these partitions as clusters.

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

1. Find a plane graph from the diagram by replacing edge crossings with dummy stations.
2. Derive the dual graph, that is, the graph found from the diagram by placing a vertex in each face of the diagram. The dual graph also has an edge that cuts each edge in the diagram, and which connects the vertices in the two faces of a diagram edge. The dual graph for a diagram is unique as the diagram embedding is known.
3. Diminish the dual graph by removing unnecessary edges. These are dual graph edges that cut diagram edges of ideal edge length and dangling edges (which are dealt with using the method of Section 4.1). This leaves only the dual graph edges that cut overlength edges.
4. Partition the plane graph by finding a route through the dual graph from the vertex in the outer face that passes through at least one other vertex and returns to the vertex in the outer face.

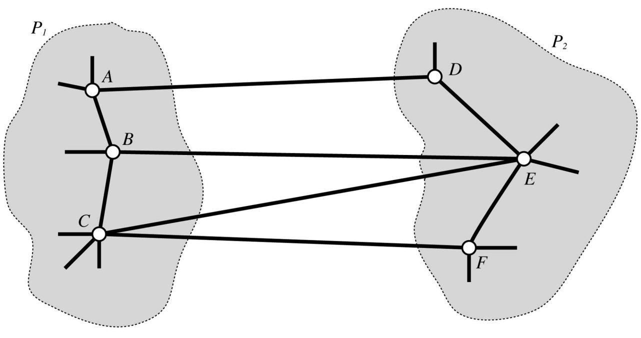

Fig. 14. Partitioning a diagram into two. The edges AD, BE, CE, and CF are all too long but the only way of shortening them is to move either the left-hand partition P_1 or the right-hand partition P_2. Both P_1 and P_2 contain other overlength edges that would stop the method for clustering based on overlength edges from finding these partitions.

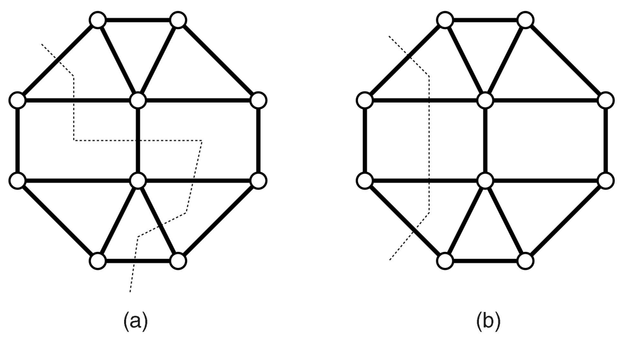

Fig. 15. Example of cuts which are likely to lead to poor and good partitions. The example in (a) is likely to lead to poor partition selection while the example in (b) is likely to lead to a better partition selection.

There are a number of possible partitions. We use a heuristic that finds a path that cuts diagram edges which are *most opposite* each other. Cuts through the diagram that consist of nearly parallel edges are more likely to result in good clusters (see Fig. 15). For each new face, we take the current edge and find an overlength edge that is both opposite the current one in the face and as close to parallel to the starting edge as possible.

The overlength edges clustering method of Section 4.1 is still required, as the partitioning in this section only operates on edges that are in cycles formed from faces in the diagram. It is not applied to "dangling" lines, which are common in the outer sections of the diagram, and which often have overlength edges.

5 Examples

In this section, we give some example metro maps produced by our system, showing for comparison the undistorted and published maps. The time taken to generate the automatically generated maps discussed in this paper is given in Table 3. These timings were performed in Java 1.6, on a computer with a 1.4 GHz Celeron M processor, 1.5 GB RAM, and running Windows XP. The values are the average of three runs. All maps (automatically generated, published, and undistorted) can be seen in Appendix B, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TVCG.2010.24.

5.1 Mexico City

The Mexico City metro map [27] is a complex, decentralized map. It has a relatively high number of lines and faces and a

TABLE 3
Time Taken to Generate Maps

Map	Time (In Seconds)
Atlanta	125
Bucharest	175
Mexico City	3559
Stockholm	695
Sydney	7189
Toronto	339
Washington	1237

Fig. 16. Mexico City map: official layout.

Fig. 17. Mexico City map: official layout, normalized to the layout software style.

total of 175 stations. The officially published map is shown in Fig. 16, with a version drawn in the style of the diagram layout software shown in Fig. 17. The undistorted map is shown in Fig. 18.

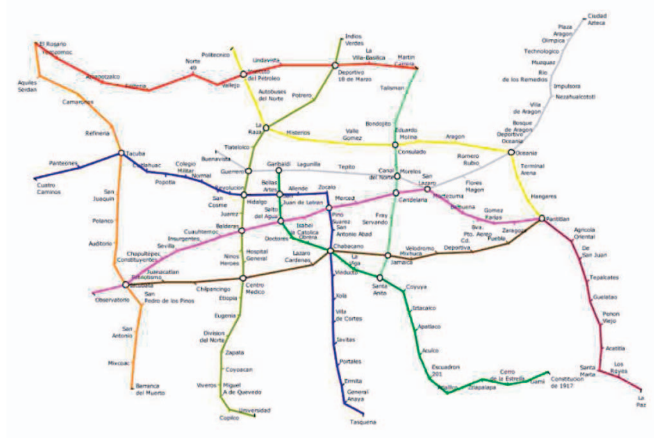

Fig. 18. Mexico City map: undistorted layout.

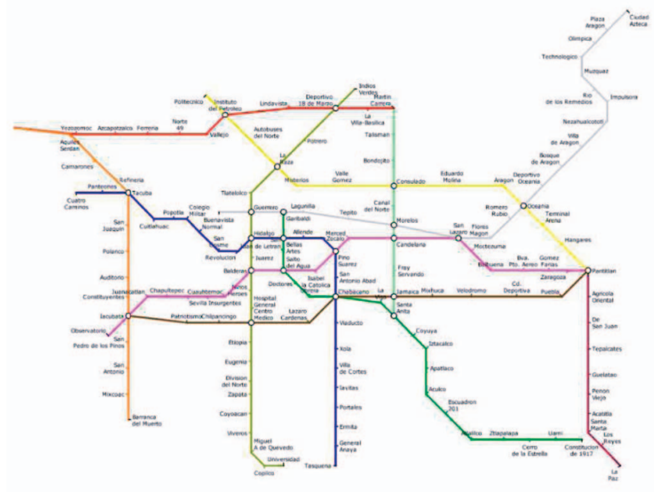

Fig. 19. Mexico City map: our layout.

TABLE 4
Weightings and Parameters for the Mexico City and Sydney Maps

Station 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}	220.0	w_{L4}	15.0	Min. Cluster Distance	3
w_{N5}	9250.0	w_{L5}	3.0	Max. Station Movement	8
		w_{L6}	1.0		
		w_{L7}	30.0		

The map produced using our method is shown in Fig. 19. We used the criteria weightings shown in Table 4 to produce this map. We believe our finished map shows a significant enhancement over both the undistorted and official maps. The official map has irregular station spacing and no attempt to achieve octilinear angles which is a feature of the automatically-drawn map. The labeling in the automatically-drawn map is also of good quality, particularly along long lines. The gray line to the top-right of the map shows a 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

Fig. 20. Sydney CityRail map: official layout.

overlength lines. A few examples of local minima are notable in our map, particularly where several lines pass through a station (where the blue, green, and brown lines meet), where a triangular face exists (top middle of the diagram), or where the red and orange lines are drawn very close together (top right of the diagram). 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.

5.2 Sydney

The Sydney CityRail [28] is a very large network covering an area of approximately 3,600 km^2 of metropolitan Sydney. The use of enlarged scale is very prominent in the central Sydney area where most of the lines converge in a tight loop around the city center. 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 Fig. 20 and the undistorted map is shown in Fig. 21. We have constrained our area of interest to the main metropolitan area of Sydney. The version of the Sydney CityRail map drawn using our method is shown in Fig. 22 and uses the criteria weightings given in Table 4.

The finished version of our map has succeeded in evening out station spacing and nearly all the edges are drawn octilinearly. Labeling is also of good quality. One particular area posing a problem for our method is the central area at the right-hand side of the map. This section has up to 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 enlarged scale seen in published metro maps), but our method does not explicitly handle scale enlargement for such a small area of the map. A few edges are not drawn octilinearly, most notably the bottommost horizontal line in the map. In this case, a local minimum has been reached where none of the clustering algorithms will find the right cluster of stations as the length of some of the edges is greater than the minimum cluster distance.

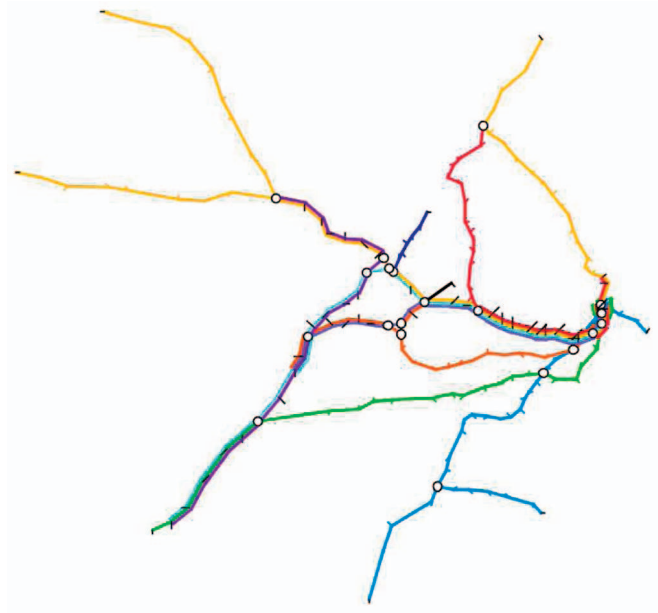

Fig. 21. Sydney CityRail map: undistorted layout.

6 Empirical Study

In this section, we report an empirical study conducted to evaluate maps drawn using the method layout described in Sections 3 and 4. We compared them with the official published map and undistorted map. We aimed to evaluate the following four hypotheses:

1. A map of a metro system drawn with our automated software is better for finding an optimal route than a undistorted map of the system.
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.
3. A map of a metro system drawn with our automated software is preferred over an undistorted map of the system.

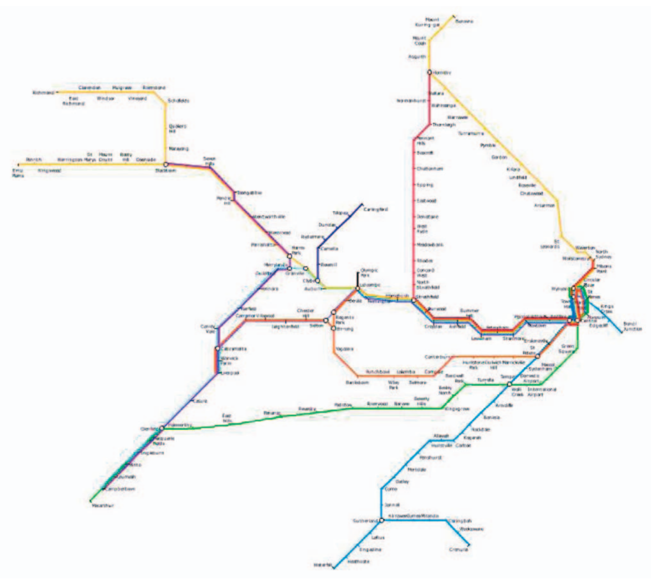

Fig. 22. Sydney CityRail map: our layout.

TABLE 5
Metro Maps Used in the Empirical Study

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

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

The empirical experiment involves a sample of human subjects performing route planning tasks using different map versions of the metro systems given in Table 5. These maps differ in characteristics and complexity from fairly simple two-line, centralized network, in the case of Atlanta, through to complex, highly interconnected, decentralized network, in the case of Mexico City.

All the maps were rendered in a similar way, to avoid discrepancies due to different label fonts or line thickness. For example, the difference between the original published Mexico City map and the normalized published map used in the empirical study can be seen in Figs. 16 and 17.

The study comprises the recording and analysis of objective and subjective measurements. Specifically, we evaluated Hypotheses A and B using objective data corresponding to the time taken by a subject to complete the task of finding a correct specific route. We evaluated Hypotheses C and D using subjective opinions obtained from the same subjects about their preferences for the maps.

6.1 Experiment Design

A total of 43 subjects participated in our study, nearly all of whom were Computer Science undergraduates from the University of Kent. We split the set of subjects into three balanced groups: I, II, and III.

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 I used the normalized published map, group II used the undistorted map, and group III used the automatically-drawn map. The map variants were distributed evenly among the groups.

A software application was written which ensured a controlled environment when showing the maps. A screenshot of the software application is shown in Fig. 23. The following procedure was used for each experiment:

1. An introductory script was read aloud and the test supervisor worked through two example questions.
2. The subjects were told to begin and were presented with problems for 20 minutes. For each problem there was a map, a question, and a list of five possible answers. The subject selected their answer and then continued on to the next problem. The subjects were able to rest after the completion of each problem.

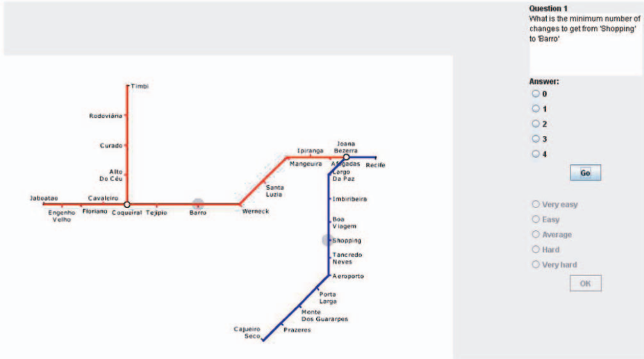

Fig. 23. Screenshot of the software application used to conduct the empirical evaluation.

3. After 20 minutes had elapsed, the subjects were shown how many of their answers were correct. For incorrect answers, they were shown the right answer.
4. The questionnaire script was then read aloud. The subjects were then shown the three variants of each map and were asked to write down their preference from "most preferable" to "least preferable."
5. The subjects were then rewarded with £5 for their time and were allowed to leave.

The full list of questions and answers for the experiments is given in Appendix B, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TVCG.2010.24. Prior to the real experiment sessions, a pilot study was used to determine any problems in our methodology. During this pilot, 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. The results from the pilot were discarded.

6.2 Statistical Analysis

6.2.1 Duration Data

Each individual in the study contributes toward learning about the duration time taken to accurately find a specific route (step iii of the experiment). With the data, the aim is to evaluate the time-effectiveness of each metro map using the time elapsed in completing route planning tasks. During the study, we recorded "exact" measurements corresponding to accurate routes. However, in some cases, the task was performed incorrectly. However, it is natural to assume those tasks will be correctly answered with more time. The key here is that we are not assessing correctness as an outcome; rather, the time taken to find the correct route. So, if an incorrect route is given at a particular time, the time taken to provide a correct route would be greater than this time. Therefore, we can and do use this as information, being compatible with the aims of the experiment without losing any information.

We used statistical methods developed in time-to-event data analysis. The idea is that we model the times when a route is given, and within the model, we include key parameters which allow us to assess the hypotheses A and B highlighted previously. See Appendix C, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TVCG.2010.24, for a

TABLE 6
Estimated Probabilities for Hypotheses A and B

Hypothesis	Probability	(Std. Err.)
A	0.986	(0.113)
B	0.994	(0.076)
A & B	0.994	(0.105)

detailed description of the model and more general statistical results.

6.2.2 Time-Effectiveness Assessment.

Hypotheses A and B can be assessed (independently and jointly) using probability statements by using Bayesian methods. Effectively, this allows us to evaluate Probability(A) which obviously records the estimate of the probability of hypothesis A being true. Table 6 shows the estimates of these probabilities. As we can see, the three hypotheses are highly likely since the estimated probabilities are considerably greater than 0.5.

6.2.3 Preference Data

Table 7 displays the contingency table of individuals' preferences for map variants across metro systems (part iv of the experiment). The preference for the automatically-drawn map is clear, with the exception of Stockholm. Note that there are six possible rankings an individual can assign within the three map versions, which can be enumerated as follows, starting with the most preferable:

1. Auto-Un-Pub,
2. Auto-Pub-Un,
3. Un-Pub-Auto,
4. Un-Auto-Pub,
5. Pub-Auto-Un, and
6. Pub-Un-Auto.

Hypotheses C and D can be summarized by events 1 and 2. Therefore, assessing hypotheses C and D simultaneously (C & D) would be equivalent to evaluating how likely events 1 or 2 are. Assessing hypotheses C & D is accomplished by collapsing the six possible rankings into a dichotomous variable, defined to be 1 if the automatically-drawn metro map is the most preferable version and 0 otherwise.

We use statistical methods to model the probability of selecting the automatically-drawn version as the most preferable. See also Appendix C, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TVCG.2010.24, for details and statistical results.

TABLE 7
Contingency Table of Map Preferences for the Three Map Versions

System	Preferences: Most			Medium			Least		
	Auto	Un	Pub	Auto	Un	Pub	Auto	Un	Pub
Atlanta	21	10	12	12	10	21	10	23	10
Bucharest	29	9	5	6	22	15	8	12	23
Mexico City	41	2	0	1	35	7	1	6	36
Stockholm	5	2	36	28	9	6	10	32	1
Toronto	24	5	14	17	20	6	2	18	23
Washington	33	0	10	9	3	31	1	40	2

TABLE 8
Estimated Probabilities for Hypotheses C and D by Metro Map System

System	Probability	(Std. Err.)
Atlanta	0.439	(0.496)
Bucharest	0.990	(0.011)
Mexico City	0.993	(0.007)
Stockholm	0.008	(0.003)
Toronto	0.778	(0.414)
Washington	0.995	(0.009)

6.2.4 Preference Assessment.

Hypotheses C and D can be statistically assessed by comparing estimated probabilities. Estimated probabilities are displayed in Table 8. Here, we can see that the automatic map is the more preferable to the alternatives in Mexico City and Washington. However, we see that for Stockholm, the published map remains preferable. Overall, the automatically-drawn metro map is the most preferable, since the overall estimated probability to choose this version as the most preferable is equal to 0.597. See Appendix C, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TVCG.2010.24, for further details.

6.3 Study Summary

In this study, we found some quantitative evidence to assert that the automatically-drawn metro map version helps users to find routes more efficiently than the alternatives. Moreover, we found that preferences tend to favor the automatically-drawn map version, particularly in highly complex metro systems, such as Mexico City and Washington. However, the evidence is not conclusive due to the Stockholm metro system. Further analysis would be required to establish individuals' preferences in terms of metro map complexity. However, the automatically-drawn metro maps have been shown to be the best maps in terms of effectiveness of finding correct routes.

7 Conclusions

We have described an automatic metro map layout system based on multicriteria optimization. The method includes labeling and station clustering. Our empirical study shows that, for some of the maps tested, the layouts produced by the method can be considered better for route planning than both published and undistorted layouts.

While improving on some less effective published maps, it is unlikely that this system will generate maps that are better than the best hand drawn maps. The complexity of the maps, combined with the sophisticated decisions made by map designers, mean that many published maps have features that are not adequately covered by our optimizer. These include: avoiding label overlap, multiple parallel lines, and local enlargement. While esthetic criteria could be developed to deal with these circumstances, there are always likely to be further issues of a more subtle nature.

In terms of future work, various aspects of the layout research could be taken forward, as the current layout mechanism does not fully capture all aspects of published metro maps. Line bends in between stations could be added, as currently changes in line direction are only allowed at stations. In addition, other geographic features such as rivers, shoreline, and parkland could be shown on the diagram. Esthetic criteria could be added to integrate these features in the layout.

The optimizer has great potential for improvement. The criteria weighting is performed in an ad-hoc manner, and while a more systematic method for deciding weighting is difficult to design, further empirical study of diagrams drawn with different characteristics would provide evidence for improving weightings. Also, if particular weightings lead to an automatic drawing similar to a published map (perhaps adapting a layout by example approach [29]), it might be possible to characterize the map in terms of weighting, enabling other diagrams to be drawn in the style of that map.

In addition, the performance of the optimizer is slow, and little effort has been put into improving the computation time. Large speedups are possible by: integrating the calculation of the metrics (which often perform very similar item-item comparisons, and so repeated iterations might be avoided); avoiding the comparison of items that are far away from each other in the diagram; and reuse of calculations from previous iterations where items have not moved. Some criteria might also be removed, for example, if the edge length criterion were measured as the square of difference between the current edge length and desired edge length, then this might remove the need for the balanced edge length criterion.

Finally, applying this work beyond the layout of complete maps is feasible. The frequent use of the metro map metaphor in laying out non-transport-based information means that there is opportunity to provide an automated layout mechanism for such areas. In addition, the widespread use of small devices connected to the Internet, such as mobile phones, means that the provision of personal travel maps would seem to be a promising application for automatic metro map layout.

Acknowledgments

The first author (Jonathan Stott) was supported by an ESPRC PhD studentship. The third author (Juan Carlos Martínez-Ovando) was supported by CONACyT (Mexico), scholarship 159977/229515.

References

[1] "Transport for London Website," http://www.tfl.gov.uk/tube/maps/, Apr. 2005.

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

[3] S. Anand, S. Avelar, J. Ware, and J. Jackson, "Automated Schematic Map Production Using Simulated Annealing and Gradient Descent Approaches," *Proc. 15th Ann. GIS Research UK Conf. (GISRUK),* 2007.

[4] R.A. Burkhard and M. Meier, "Tube Map: Evaluation of a Visual Metaphor for Interfunctional Communication of Complex Projects," *Proc. Fourth Int'l Conf. Knowledge Management (I-KNOW '04),* pp. 449-456, 2004.

[5] W.C. Hahn and R.A. Weinberg, "A Subway Map of Cancer Pathways," *Nature Reviews Cancer,* vol. 2, no. 5, May 2002.

[6] T. Bang, K. Grønbæk, and P.S. Hansen, "Using a Metro Map Metaphor for Organizing Web-Based Learning Resources," *Proc. World Conf. Educational Media, Hypermedia and Telecomm. (ED-MEDIA '02),* pp. 647-652, 2002.

[7] J.M. Stott and P. Rodgers, "Metro Map Layout Using Multicriteria Optimization," *Proc. Information Visualization Conf. (IV '04),* pp. 355-362, 2004.

[8] J.M. Stott and P. Rodgers, "Automatic Metro Map Design Techniques," *Proc. 22nd Int'l Cartographic Conf.,* 2005.

[9] J.M. Stott, "Automatic Layout of Metro Maps Using Multicriteria Optimisation," PhD dissertation, Computing Laboratory, Univ. of Kent, 2008.

[10] S.-H. Hong, D. Merrick, and H.A.D. do Noscimento, "The Metro Map Layout Problem," *Proc. Asia Pacific Symp. Information Vizualization (APVIS '04),* pp. 91-100, 2004.

[11] S.-H. Hong, D. Merrick, and H.A.D. do Noscimento, "Automatic Visualisation of Metro Maps," *J. Visual Languages and Computing,* vol. 17, pp. 203-224, 2006.

[12] M. Nöllenburg and A. Wolff, "A Mixed Integer Program for Drawing High-Quality Metro Maps," *Proc. 13th Int'l Symp. Graph Drawing (GD '05),* pp. 321-333, 2005.

[13] D. Merrick and J. Gudmundsson, "Path Simplification for Metro Map Layout," *Proc. 14th Int'l Symp. Graph Drawing (GD '07),* pp. 258-269, 2007.

[14] M.A. Bekos, M. Kaufmann, K. Potika, and A. Symvonis, "Line Crossing Minimization on Metro Maps," *Proc. 15th Int'l Symp. Graph Drawing (GD '07),* pp. 231-242, 2007.

[15] M. Agrawala and C. Stolte, "Rendering Effective Route Maps: Improving Usability through Generalization," *Proc. ACM SIGGRAPH '01,* pp. 241-249, 2001.

[16] S. Avelar and R. Huber, "Modeling a Public Transport Network for Generation of Schematic Maps and Location Queries," *Proc. 20th Int'l Cartographic Conf.,* pp. 1472-1480, 2001.

[17] H. Casakin, T. Barkowsky, A. Klippel, and C. Freska, "Schematic Maps as Wayfinding Aids," *Proc. Int'l Conf. Spatial Cognition II,* pp. 54-71, 2001.

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

[19] S. Cabello, M. de Berg, S. van Dijk, and M.J. van Kreveld, "Schematization of Road Networks," *Proc. 17th Symp. Computational Geometry,* pp. 33-39, 2001.

[20] U. Lauther and A. Stübinger, "Generating Schematic Cable Plans Using Spring Embedder Methods," *Proc. Symp. Graph Drawing,* pp. 465-466, 2001.

[21] R. Davidson and D. Harel, "Drawing Graphs Nicely Using Simulated Annealing," *ACM Trans. Graphics,* vol. 15, no. 4, pp. 301-331, 1996.

[22] C. Kosak, J. Marks, and S. Shieber, "A Parallel Genetic Algorithm for Network-Diagram Layout," *Proc. Fourth Int'l Conf. Genetic Algorithms,* pp. 458-465, 1991.

[23] H.C. Purchase, R.F. Cohen, and M. Jones, "Validating Graph Drawing Aesthetics," *Proc. Symp. Graph Drawing,* F.J. Brandenburg, ed., pp. 435-446, 1995.

[24] T. Barkowsky, L.J. Latecki, and K.-F. Richter, "Schematizing Maps: Simplification of Geographic Shapes by Discrete Curve Elimination," *Proc. Int'l Conf. Spatial Cognition II,* pp. 41-53, 2000.

[25] M. Nöllenburg, "Automated Drawing of Metro Maps," Master's thesis, Inst. für Theoretische Informatik, Univ. Karlsruhe, Aug. 2005.

[26] T. Slocum, R. McMaster, F. Kessler, and H. Howard, *Thematic Cartography and Geographic Visualization.* Prentice Hall, 2005.

[27] Mexico City "Sistema de Transporte Colectivo" Website, http://www.metro.df.gob.mx/, Feb. 2008.

[28] "Sydney CityRail Website," http://www.cityrail.info/, Sept. 2007.

[29] I. Vollick, D. Vogel, M. Agrawala, and A. Herzmann, "Specifying Label Layout Styles by Example," *Proc. 20th ACM Symp. User Interface Software and Technology (UIST '07),* pp. 221-230, Oct. 2007.

Jonathan Stott graduated with a first class degree with honors in computer science in 2003 from the University of Kent, where, he received the PhD degree in 2008 with a thesis titled "Automatic Layout of Metro Maps Using Multicriteria Optimization."

Peter Rodgers is a senior lecturer at the University of Kent. His main research interests include diagrammatic visualization, including graph and Euler diagram layout techniques. He has led several research projects supported by national and international funding bodies.

Juan Carlos Martínez-Ovando is a postgraduate student in statistics at the University of Kent. His research focuses on Bayesian nonparametric methods.

Stephen G. Walker is a professor in statistics at the University of Kent. His main area of research is Bayesian nonparametric methods. He has served as a program committee member for several conferences in Bayesian statistics.