CSIS Discussion Paper No. **126**

Computer-aided design of bus route maps

Yukio Sadahiro*, Takahito Tanabe**, Maxime Pierre***, & Koichi Fujii**

May 2014

***** Center for Spatial Information Science, The University of Tokyo
** NTT DATA Mathematical Systems Inc.
*** National School of Geographical Sciences, The Université Paris Diderot

Center for Spatial Information Science, The University of Tokyo
5-1-5, Kashiwanoha, Kashiwa-shi, Chiba 277-8568, Japan
E-mail: sada@csis.u-tokyo.ac.jp

Computer-aided design of bus route maps

Abstract

The bus route map is a diagram that aims to convey necessary information for map readers to find an appropriate way of moving from an origin to a destination. Design of bus route map is a complicated and time-consuming task that requires careful consideration of visibility and aesthetics. This paper proposes a new computational method for designing bus route maps. The method helps us to reduce the six types of undesirable elements in bus route maps, i.e., intersection, gap, shift, overlap, misalignment, and acute bend. The method consists of two phases: the topological phase determines the relative order of bus routes on each road segment and the cartographic phase calculates the actual location of bus routes drawn on a map. This paper applies the method to the design of bus route maps of Chiba City, Japan. The result supports the effectiveness of the method as well as reveals open topics for future research.

Keywords: bus route maps, computational procedure, mathematical optimization, cartographic design

Introduction

The bus route map is a diagram that visualizes the bus route network as a schematic map. Figure 1a illustrates a part of hypothetical bus route map. Similar to the metro map, it aims to convey necessary information for map readers to find an appropriate way of moving from an origin to a destination. The bus route map emphasizes the network structure of bus routes, i.e., the topological relationship between bus routes, terminals, and junctions, without drastically changing their geometrical structure. Design of bus route map is a complicated task that requires careful consideration of visibility and aesthetics. Assistance of computational method is highly desirable and useful for improving the efficiency of this time-consuming manual process.

Few methods are available for computational design of bus route maps at present. However, efficient algorithms of metro map design have been developed in computer science and operations research. The metro map visualizes metro lines as a diagram that provides an effective way of navigating the metro network (Ovenden 2003). Existing algorithms can be broadly classified into cartographic methods and topological methods, each of which focuses on the cartographic and topological aspect of metro map, respectively.

Cartographic methods focus on the geometrical properties of metro network, i.e., angle, length, and arrangement of metro lines and stations. The earliest example is Beck's Underground Map of London published in 1931 (Garland 1994). This schematic metro map utilizes only horizontal, vertical, and diagonal lines to indicate metro lines and locates metro stations evenly on the lines. Following the rules of Beck’s map, Hong, Merrick, and Nascimento (2005) and Hong, Merrick, and do Nascimento (2006) develop five automated methods for generating metro maps. They formulate map design as a mathematical programming problem to find a compromise between conflicting rules. Stott and Rodgers (2004) and Stott et al. (2005) also take a similar approach though they consider some additional conditions. Nöllenburg and Wolff (2006) utilizes a different optimization problem that strictly ensures the octilinearity of metro lines.

Topological methods stress the topological structure of metro network. The topology of metro map primarily depends on the topology of actual metro network. However, when more than a single line share the same section between stations, the arrangement of metro lines in the section can be determined flexibly to some extent on a map. Inappropriate arrangement causes unnecessary intersections of metro lines, which reduces the aesthetics and readability of the map. To avoid this problem, Benkert et al. (2007) formulates the metro map design as an optimization problem that minimizes the intersections of metro lines. The paper proposes a computational algorithm that solves the problem in $O(|L|^2)$ time by using dynamic programming. Bekos et al. (2008) calls the problem the *Metro-Line Crossing Minimization problem* (*MLCM*) and extends it to the *Metro-Line Crossing Minimization problem with terminals at fixed station ends* (*MLCM-Fixed SE*) where the location of some lines is fixed at their terminals. Asquith, Gudmundsson, and Merrick (2008) proposes a faster algorithm that solves the MLCM-Fixed SE in polynomial time. The algorithm developed by

Argyriou et al. (2009) and Argyriou et al. (2010) runs in $O((|E|+|L|^2)|E|)$ time for the MLCM problem in a limited case. Nöllenburg (2010) considers three variants of the MLCM (*MLCM-P*, *MLCM-PA*, and *MLCM-T1*) that can be computed in a shorter time.

The above methods are also applicable to the design of bus route maps since they share the same objective with metro maps. Bus route maps, however, are usually far more complicated than metro maps so that their design involves a wider variety of problems to be solved. Figure 1b is another bus route map indicating the same bus routes shown in Figure 1a. It contains the six types of undesirable elements in map design, i.e., intersection, gap, shift, overlap, misalignment, and acute bend. They reduce not only the aesthetics but also the readability of bus route maps. Intersections of bus routes should be kept minimal as discussed in existing studies. Gaps between bus routes increase the space required to locate all the bus routes on a single map. Shifts of bus routes on a street cause the visual complexity of the map. Overlaps of bus routes make us difficult to distinguish bus routes running on different streets. Misalignments of bus routes and road network occur when they are drawn simultaneously on a single map. Acute bends clutter the map especially when many bus routes are running in parallel.

The techniques developed for the metro map cannot fully resolve these six types of problems. Combining existing methods, we may avoid intersections, overlaps and acute bends. However, existing methods do not consider gaps, shifts, and misalignments explicitly. There is no integrated framework tailored for the design of bus route maps that treat the six types of undesirable elements simultaneously. To answer this demand, this paper proposes a new computational method for designing bus route maps. Our focus is on the bus route map that is not highly schematized as Beck's map. Bus routes basically keep their geometrical structure as shown in Figure 1a. The arrangement of bus stops is out of the scope of the paper since it is primarily determined by the location of bus routes. We describe our method in the next section, which is followed by several applications. We finally summarize the result with discussion.

Method

Our method consists of two phases: topological phase and cartographic phase. The topological phase aims to reduce the intersections, gaps, and shifts of bus routes. To this end, this phase focuses on the topological structure of bus routes, or more specifically, determines the order of bus routes on each link of road network. Using the obtained result, the cartographic phase calculates the position of bus routes on a map. This phase aims to avoid the overlaps, misalignments, and acute bends of bus routes. We describe the two phases successively in the following.

Topological phase

Table 1 summarizes the notations for spatial objects, constants, and variables used in the topological phase. Suppose a road network on which a set of K bus routes $\boldsymbol{R}=\{R_1, R_2, ..., R_K\}$ are running. Figure 2a shows a part of road network, where white and black points indicate end nodes and intermediate vertices of the network, respectively. Figure 2b shows bus routes and bus stops on the network. We define a *topological road model* based on the road network to describe the topological structure of bus routes. The model focuses only on the topological structure of road network by omitting its geometrical properties such as the angle and length of each element. The model neglects the roads without bus routes, intermediate vertices, and end nodes where bus routes do not branch. Instead, the model adds extra nodes and links to represent the terminals of bus routes explicitly. The nodes and links of the topological road model are simply called *nodes* and *links*, denoted by $\boldsymbol{N}$= and $\boldsymbol{L}=\{L_1, L_2, ..., L_M\}$, respectively. Figure 2c shows the topological road model defined based on the road network and bus routes shown in Figures 2a and 2b. Nodes N_1 and N_3 indicate the branches of bus routes, while N_2 and N_4 correspond to terminals. Extra nodes and links added at terminals such as N_5, N_6, L_{10} and L_{11} are called *terminal nodes* and *terminal links*, respectively. The set of terminal links is denoted by $\boldsymbol{L'}=\{L_{M+1}, L_{M+2}, ..., L_{M+Q}\}$. Let E_{i1} and E_{i2} be the first and second end nodes of link L_i, respectively. In Figure 2c, for instance, we may define $E_{21}=N_1$, $E_{22}=N_2$, $E_{31}=N_2$, and $E_{32}=N_3$.

Here, we introduce the concept of street. A *street* is a sequence of connected links on which bus routes should be drawn smoothly without locational shifts. One definition is that two links belong to the same street if they are connected at an angle larger than a threshold value ρ. In this definition, the value of ρ is given a priori based on a cartographer's aesthetic sense. An alternative is to adopt the definition of streets in the real world as they are. However, this choice is often problematic since streets in the real world do not always run straight.

Bus routes are represented as sets of links in $\boldsymbol{L}+\boldsymbol{L'}$. Let p_{ri} be a binary function that indicates whether route R_r is running on link L_i:

$$p_{ri} = \begin{cases} 1 & \text{if } R_r \text{ is running on } L_i \\ 0 & \text{otherwise} \end{cases} . \tag{1}$$

Using this function, we represent route R_r as $R_r=\{p_{r1}, p_{r2}, ..., p_{r, M+Q}\}$ (r=1, ..., K). In Figure 2, for instance, routes R_1 and R_2 are represented as

$$\begin{aligned} R_1 &= \{ p_{11}, p_{12}, p_{13}, p_{14}, p_{15}, p_{16}, ...\} \\ &= \{1, 0, 0, 0, 1, 0, ...\} \end{aligned} \tag{2}$$

and

$$\begin{aligned} R_2 &= \{ p_{21}, p_{22}, p_{23}, p_{24}, p_{25}, p_{26}, ...\} \\ &= \{1, 1, 1, 1, 0, 0, ...\} \end{aligned} , \tag{3}$$

respectively.

Using the topological road model, we describe the arrangement of bus routes on a map. Our focus is on the relative order of bus routes on each link. We define *tracks* to represent the candidate location of bus routes on each link. Each track can be assigned at most a single route. Link L_i has a set of n_i tracks denoted by $\boldsymbol{T_i}=\{T_{i1}, T_{i2}, ..., T_{in_i}\}$. Figure 3a illustrates tracks defined for bus routes shown in Figure 2. We prepare enough number of tracks on each link except terminal links to keep the flexibility of bus route arrangement. On terminal links, the number of tracks is equal to that of bus routes terminate at the node.

To derive the arrangement of bus routes, we define the following variables (they also appear in Table 1).

1) s_{ij} ($1 \le i, j \le M$)

It is a binary function that becomes one if links L_i and L_j belong to the same street:

$$s_{ij} = \begin{cases} 1 & \text{if } L_i \text{ and } L_j \text{ belong to the same street} \\ 0 & \text{otherwise} \end{cases}. \tag{4}$$

2) c_{iejl} ($1 \le r \le K$, $1 \le i, j \le M$, $1 \le e \le 2$, $1 \le l \le n_j$)

This variable indicates the relative location of track T_{jl} around end node E_{ie} counted clockwise from T_{i1} except the tracks of terminal links. Figure 4a illustrates the numbering of tracks around node N_3 counted from T_{61}, where $N_3=E_{32}=E_{41}=E_{61}=E_{72}$. In this case, for instance, $c_{6161}=1$, $c_{6141}=5$, and $c_{6173}=10$. If we evaluate the location of tracks in relation to L_3, we have $c_{3261}=5$, $c_{3241}=9$, and $c_{3273}=14$. If L_i is not connected with L_j at E_{ie}, $c_{iejl}=0$ for any l.

The above two variables act as constants in the derivation of bus route map since they are given a priori. We then introduce variables to be determined through the arrangement of bus routes.

1) x_{rikjl} ($1 \le r \le K$, $1 \le i, j \le M+Q$, $1 \le k \le n_i$, $1 \le l \le n_j$)

This is a binary variable that becomes one if route R_r is assigned to tracks T_{ik} and T_{jl}, where links L_i and L_j are directly connected at a node. Figure 3b presents an assignment of routes to tracks. The assignment of route R_1, for instance, is represented as

$$x_{1ikjl} = \begin{cases} 1 & (i, j, k, l) = (1,5,2,1), (5,1,1,2) \\ 0 & (i, j, k, l) \ne (1,5,2,1), (5,1,1,2) \end{cases}. \tag{5}$$

Similarly, the assignment of route R_3 is represented as

$$x_{3ikjl} = \begin{cases} 1 & (i, j, k, l) = (10,3,1,2), (3,10,2,1), (3,6,2,2), (6,3,2,2) \\ 0 & (i, j, k, l) \ne (10,3,1,2), (3,10,2,1), (3,6,2,2), (6,3,2,2) \end{cases}. \tag{6}$$

2) y_{ike} ($1 \le i \le M$, $1 \le k \le n_i$, $1 \le e \le 2$)

This variable indicates the relative location of the track with which a route assigned to track T_{ik} at end node E_{ie} is connected. It is defined by

$$y_{ike} = \sum_{r} \sum_{j} \sum_{l} c_{iejl} x_{rikjl} \ . \tag{7}$$

Figure 4b illustrates an assignment of bus routes. Route R_3, for instance, is assigned to tracks T_{32} and T_{62}. Consequently, we obtain $y_{322}=c_{3262}=6$ and $y_{621}=c_{6232}=15$. Similarly, since R_2 is assigned to tracks T_{33} and T_{43}, $y_{332}=c_{3343}=11$ and $y_{431}=c_{4333}=10$.

3) z_{ike} ($1 \le i \le M+Q$, $1 \le k \le n_i$, $1 \le e \le 2$)

This is a binary variable indicating the termination of the route assigned to track T_{ik} at end node E_{ie}. It is defined by

$$z_{ike} = \sum_{r} \sum_{j \in L', \text{ connected to } E_{ie}} \sum_{l} x_{rikjl} \ . \tag{8}$$

This variable becomes one only if the route assigned to track T_{ik} terminates at end E_{ie}. In Figure 3b, route R_3 terminates at node N_2 ($=E_{22}=E_{31}=E_{10,1}$), and consequently, $z_{321}=1$.

4) δ_{ike} ($1 \le i \le M+Q$, $1 \le k \le n_i$, $1 \le e \le 2$)

This is a binary variable indicating the connection of track T_{ik} with another track at E_{ie} by sharing the same route. It is defined by

$$\delta_{ike} = 1 - \sum_{r} \sum_{j, \text{ connected to } E_{ie}} \sum_{l} x_{rikjl} \ . \tag{9}$$

In Figure 3b, tracks T_{12} and T_{51} share route R_1 at node N_1 ($=E_{12}=E_{21}=E_{51}$). We thus obtain $\delta_{122}=\delta_{511}=1$. Similarly, route R_3 yields $\delta_{10,1,1}=\delta_{321}=\delta_{322}=\delta_{621}=1$.

5) φ_{ikme} ($1 \le i \le M$, $1 \le k$, $m \le n_i$, $1 \le e \le 2$)

This variable evaluates the undesirable arrangement of two bus routes, each of which is assigned to track T_{ik} and T_{im} at E_{ie}, respectively. It is based on the number of gaps and intersections in the arrangement of bus routes. In general, a tradeoff relationship exists between the two undesirable elements, i.e., reduction of gaps inevitably increases intersections. Since they are not completely avoidable, we need to seek a compromise. Variable φ_{ikme} is defined by the following inequality in such a way that cancels the gaps and intersections at E_{ie}:

$$y_{ime} \le y_{ike} + C\left\{\varphi_{ikme} + \alpha\left(z_{ike} + z_{ime}\right) + \left(1-\alpha\right)\left(\delta_{ike} + \delta_{ime}\right)\right\} \quad \left(\forall k, m, e, c_{ieik} < c_{ieim}\right), \tag{10}$$

where C is a constant larger enough than the number of tracks of L_i, and α is the weight of gaps compared with intersections ranging from zero to one. If gaps are more undesirable than intersections, we adopt a large α to reduce gaps. If gaps are relatively acceptable, a small α reduces intersections though it may increase gaps.

Using the above variables, we formulate the arrangement of bus routes as a mathematical optimization problem. The problem aims to minimize the undesirable arrangement of bus routes measured by the summation of φ_{ikme}.

Problem RA (Route Assignment):

$$\underset{x_{rikjl}}{\text{minimize}} \sum_{i}\sum_{e}\sum_{k}\sum_{m, c_{STiik} < c_{STiim}} \varphi_{ikme} \quad .$$

subject to

1: $x_{rikjl} = x_{rjlik} \quad (\forall r, i, j, k, l)$

2: $x_{rikjl} = \sum_{j', l' \neq j, l} x_{rj'l'ik} \quad (\forall r, i, j, k, l)$

3: $s_{ij} x_{rikjl} = 0 \quad (\forall r, i, j, \beta < |k - l|)$

4: $p_{ri} = \sum_{k}\sum_{j \neq i}\sum_{l} x_{rikjl} = \sum_{k}\sum_{j \neq i}\sum_{l} x_{rjlik} \quad (\forall r, i)$

5: $\sum_{j\text{, connected to link } i}\sum_{l} x_{rikjl} = 1 \quad (\forall r, i \in L', k)$

6: $\sum_{i\text{, connected to link } j}\sum_{l} x_{rikjl} = 1 \quad (\forall r, j \in L', l)$

7: $\sum_{r}\sum_{j\text{, connected to link } i}\sum_{l} x_{rijkl} \leq 1 \quad (\forall i, k)$

8: $z_{ike} \leq \sum_{r}\left\{ \sum_{j \in L'\text{, connected to } E_{ie}}\sum_{l} x_{rijkl} + \sum_{j \in L'\text{, connected to } E_{ie}}\sum_{l} x_{rjilk} \right\} \quad (\forall i, k)$

9: $\delta_{ike} = 1 - \sum_{r}\sum_{j\text{, connected to } E_{ie}}\sum_{l} x_{rikjl}$

10: $y_{ike} = \sum_{r}\sum_{j}\sum_{l} c_{iejl} x_{rikjl}$

11: $y_{ime} \leq y_{ike} + C\left\{ \varphi_{ikme} + \alpha\left(z_{ike} + z_{ime}\right) + \left(1 - \alpha\right)\left(\delta_{ike} + \delta_{ime}\right) \right\} \quad (\forall k, m, e, c_{ieik} < c_{ieim})$

12: $\varphi_{ikme} \geq 0 \quad (\forall i, k, m, e)$

Table 2 briefly describes the above constraints. Among the three problems in the topological phase, gaps and intersections are evaluated in the objective function while shifts are considered in the third constraint. The problems can be either strictly prohibited or permitted to a certain extent. We can avoid gaps or intersections by setting α to 1or 0, respectively. A value between 0 and 1 represents a compromise between gaps and intersections. If no shift is permissible, we set β to zero. If a shift of one degree is acceptable, we adopt β=1. One may even remove the third constraint if shifts are not problematic at all.

Problem RA is a variant of the MLCM that aims to minimize the intersections of metro lines. It is a binary integer programming problem whose global optimum cannot be obtained in a practical time. We thus employ a heuristic approach to solve Problem RA.

Cartographic phase

The topological phase yields the set of x_{rikjl}'s that represents the topological arrangement of bus routes on a map, i.e., the relative order of bus routes on each link. Using the result, the cartographic phase calculates the actual position of bus routes drawn on a map.

We define a *cartographic road model* that indicates the location of road segments on a map. The model is based on the original road network data such as shown in Figure 2a. The cartographic road model neglects the roads without bus routes while adding extra nodes to represent the terminals of bus routes. It involves both topological and locational information of road network, the latter of which indicates the location of road segments on a map. Figure 5 displays the cartographic road model of the road network shown in Figure 2. As seen in the figure, an extra node is added to represent a terminal of bus routes. To distinguish the links of this model from those of topological road model, we call the links of cartographic road model *road sections* hereafter. When roads are represented as polygons in spatial data, road sections are given by the center lines of road polygons as shown in Figure 5. If roads are already represented as a network, the cartographic road model is defined based on the network data.

Each bus route is drawn as a collection of line segments, which we simply call *segments*. Each road section is assigned a set of segments of the same length running in parallel at the same interval v. The number of segments of each section is equal to that of the tracks between the most outer tracks that are assigned bus routes. On a street, the most outer tracks are defined based on all the tracks of the sequence of sections. We initially locate segments in such a way that the center of the most outer segments is exactly located on each road section. If a track is not assigned a bus route, the place of its corresponding segment is kept empty. On a street, we place segments in such a way that the summation of the distance between each road section and the center of the most outer segments is minimized.

Figure 6a shows a part of a cartographic road model, where streets are indicated by bold lines. Figure 6b shows the initial location of segments on the network. The center of the most outer segments basically fits road sections, though the segments are not centered on streets.

Given a set of segments, we connect them at each end. When two neighboring segments of a bus route intersect, we simply shorten the segments to connect them. If segments do not intersect, we extend the segments until they connect. As a result, we obtain a map such as shown in Figure 6c. This simple procedure, however, does not always work successfully. Problems often occur including acute bends, overlaps, and misalignments.

Figure 6c shows an example of acute bends. Acute bends are acceptable when only a single bus route is running. However, acute bends of many bus routes running in parallel degrade the visual quality of bus route maps. To avoid the problem, we calculate the angle between neighboring segments and add extra segments if the angle is smaller than a predetermined threshold $\theta_{\min}$. We locate extra segments at intervals v at the same angle with the neighboring segments except for the most inner bus route. Figure 6d shows the result of this revision applied to the bus routes shown in Figure 6c.

Overlaps of bus routes occur in dense road networks. Figure 7a shows the road sections on which bus routes are running. Figure 7b is the initial location of segments of bus routes, which causes overlaps as shown in Figure 7c. We resolve the overlaps of bus routes by translating the segments of one of the sections until the outer segments are clearly separated. In Figure 7b, for instance, we translate segments of road section RS_1 until segments S_{15} and S_{21} are clearly separated as shown in Figure 7d.

Figure 8 shows the misalignments of road sections and bus routes. Misalignments occur only on streets when bus routes and road network are displayed simultaneously on a single map. Figures 8a and 8b shows road sections and the initial arrangement of bus routes, respectively. Two bus routes running from east to west are located away from road sections in Figure 8b. Since extreme misalignments are clearly misleading, we calculate the distance between each road section and the closest segment. If the distance is larger than a predetermined value d, we translate the ends of segments so that their center is exactly located on the road section. Figure 8c shows the result of this revision applied to the segments in Figure 8b.

The above three revisions can be implemented as computational procedures. The computational cost of simple algorithms is $O(M'_c)$, $O(M_c^2)$, and $O(M_c)$, respectively, where M'_c and M_c are the number of intermediate vertices and road sections. The complexity of the second step can be improved to $O(M_c \log M_c + I_c)$ by the algorithms proposed by and Chazelle and Edelsbrunner (1992), Mulmuley (1990), and Chazelle and Edelsbrunner (1992) where I_c is the number of intersections. Since $M'_c < M_c$, the complexity of the overall procedure is $O(M_c \log M_c + I_c)$.

The cartographic phase contains three parameters to be given, i.e., v, $\theta_{\min}$, and d. The interval of segments v depends on the line width of bus routes w. In metro maps in Ovenden (2003), metro lines running between the same stations are drawn as either $v=w$, i.e., contiguous to each other, or $1.2w \leq v \leq 2.0w$ in many cases. This also holds for bus route maps provided on the Internet. The other parameters $\theta_{\min}$ and d are not considered explicitly in existing metro maps. Acute bends and misalignments are not serious problems in metro maps since only a few lines are running in parallel.

In bus route maps, on the other hand, $\theta_{\min}$ is usually between π/4 and π/2. Concerning d, there does not seem to exist a common standard.

Manual adjustment of parameters and editing of bus route map

The above computational procedure does not always yield satisfactory bus route maps. Adjustment of parameter values and manual editing of generated map are usually necessary to obtain a final product.

Both topological and cartographic phases contain parameters to be given by a cartographer, i.e., α and β in the topological phase, and v, $\theta_{\min}$, and d in the cartographic phase. Though they need to be determined before the computational procedure, the initial setting does not assure the best result. We recommend trying various values interactively with checking the obtained result.

Manual editing is also necessary in various situations. We illustrate two cases in the following.

Figure 9 shows a case where two overlaps of bus routes interact with each other. Figures 9a and 9b are road sections and the initial arrangement of bus routes obtained in the cartographic phase. Since the segments on sections S_1 and S_2 overlap, the computational procedure translates those on S_2 downward, which results in the overlap of segments on S_2 and S_3 as shown in Figure 9c. In such a case, we need to translate segments on S_1 and S_3 simultaneously by hand to resolve all the overlaps of segments as shown in Figure 9d.

Figure 10 shows another case where human decision is helpful. Figures 10a and 10b show road sections and the initial locations of segments, respectively. Since the segments on S_1 and S_3 overlap, the computational procedure translates the segments on S_3 to obtain Figure 10c. In this case, however, several other options are also available. We may translate the segments of all the sections S_1, S_2, and S_3 (Figure 10d), those of only S_2 (Figure 10d), or omit several segments of S_2 to connect directly the segments of S_1 and S_3 (Figure 10f). Figures 10c, 10e, and 10f are safer than Figure 10d because they translate the segments of only a single section so that the effect is locally limited. Figure 10e looks well-balanced since the segments of longer sections S_1 and S_3 are both centered with respect to road sections. We can also choose Figure 10f if the angle between S_1 and S_3 is not too acute. Final decision clearly requires the evaluation of a professional cartographer.

Applications

This section tests the effectiveness of the proposed method by generating bus route maps of Chiba City, Japan. Source data are provided in GML format from the Ministry of Land, Infrastructure, Transport and Tourism. We converted the data into text format to use in the topological phase. We employed a mathematical programming software NUOPT ver.16 (NTT DATA Mathematical Systems Inc.) for solving Problem RA. NUOPT employs the tabu search proposed in Nonobe and Ibaraki (1998) and Nonobe and Ibaraki (2001) to solve integer programming problems. Using the outcome of the topological phase, the cartographic phase generates bus route maps by using Java language. We set α and β to 1 and 0, respectively, in all the cases except that shown in Figure 12a. This implies that we strictly prohibit shifts and gaps in our applications. The interval of bus routes v is $1.5w$. Acute bends are defined by $\theta=\pi/2$. We set parameter d to w, the maximum gap between bus routes and road network.

Figure 11 illustrates a part of bus route map obtained by the proposed method. The dotted lines indicate the initial result generated in the cartographic phase, where misalignments occur between bus routes and road network. Since the gap is larger than $d=w$, the computational procedure translated the ends of segments to obtain the final result shown by solid lines.

Figure 12 shows the tradeoff between gaps and intersections in a bus route map. We set α to 0 and 1 in the two figures, respectively. As a result, Figure 12a contains a gap between bus routes, which is not observed in Figure 12b. On the other hand, the latter has more intersections of bus routes; bus routes indicated by red and dark blue lines intersect three times in Figure 12b while they do not intersect at all in Figure 12a.

Figure 13 shows a case where a manual editing was necessary. We revised the map in the rectangular area of Figure 13, which is enlarged in Figure 14. Figure 14a shows the road sections and their ends that correspond to bus stops. Nine bus routes terminate at bus stop B_1. Figure 14b shows a part of bus route map obtained by the automated procedure. Short black lines indicate the initial location of segments assigned to bus routes of orange, red, and dark green lines. Since road section RS_1 is very short, the segments of the three bus routes on RS_1 and RS_2 do not intersect at adequate positions. The computational procedure thus removed the three segments on RS_1 to generate the result shown in Figure 14b. This map, however, is misleading since the nine bus routes do not seem to terminate at the same bus stop. We thus moved the segments on RS_1 upward and those on RS_2 downward by hand in such a way that they intersect appropriately to reach the result shown in Figure 14c. Manual editing becomes necessary especially when many bus routes are densely running.

Concluding discussion

This paper has developed a computational method for designing bus route maps. The method is effective for resolving the six types of undesirable elements in map design, i.e., intersections, gaps, shifts, overlaps, misalignments, and acute bends of bus routes. The method consists of topological and cartographic phases each of which resolves three problems, respectively. The paper applies the proposed method to the design of bus route maps in Chiba, Japan. The result supports the effectiveness of the method as well as revealing situations where manual editing is still necessary.

We finally discuss some limitations of the paper and potential directions for future research.

First, bus route maps generated by the proposed method still requires manual editing.

Problems tend to occur especially when many bus routes are running on a dense road network. We should further improve the method to reduce the time-consuming manual processes.

Second, implementation of the proposed method as an integrated computational package should be considered. We need to run NUOPT, ArcGIS, and Java scripts separately to obtain the final product at present. Conversion and transfer of spatial data between different platforms decrease the efficiency of overall procedure. An integrated package that generates bus route maps directly from spatial data in a standard format such as XML should be developed.

Third, the proposed method does not fully schematize bus routes as done in Beck's Underground Map of London. If further schematization is necessary, we may use the existing methods developed for metro maps mentioned in Section 1. The methods, however, may not work successfully for bus route maps since they are far more complicated than metro maps. We may need to develop a new method for further schematizing bus route maps.

Fourth, the present paper does not consider the arrangement of bus stops. It is because the arrangement of bus routes is more crucial in map design. Bus stops are usually located by simply following the arrangement of bus routes. This method, however, may not work especially when many bus stops are located along many bus routes. Bus stops may overlap with each other, which requires a revision of the arrangement of bus routes. Simultaneous arrangement of bus routes and bus stops should be examined.

Fifth, color assignment of bus routes is an important topic. Unlike metro lines, bus routes usually do not have their own colors that are consistently used in route maps, at bus stops and inside vehicles. Since the choice of color scheme is flexible, we can choose a color scheme that minimizes visual conflicts such as the close location and intersection of similar colors. We should incorporate color design into the proposed method in future research.

References

Argyriou, E., M. A. Bekos, M. Kaufmann, and A. Symvonis. 2009. "Two Polynomial Time Algorithms for the Metro-line Crossing Minimization Problem." In *Graph Drawing*, edited by I. G. Tollis and M. Patrignani, 336-47. Springer Berlin Heidelberg.

———. 2010. "On Metro-Line Crossing Minimization." Review of. *Journal of Graph Algorithms and Applications* 14 (1):75-96.

Asquith, M., J. Gudmundsson, and D. Merrick. 2008. "An ILP for the metro-line crossing problem." In *Proceedings of the fourteenth symposium on Computing: the Australasian theory - Volume 77*, 49-56. Wollongong, NSW, Australia: Australian Computer Society, Inc.

Bekos, M. A., M. Kaufmann, K. Potika, and A. Symvonis. 2008. "Line Crossing Minimization on Metro Maps." In *Graph Drawing*, edited by S.-H. Hong, T. Nishizeki and W. Quan, 231-42. Springer Berlin Heidelberg.

Benkert, M., M. Nöllenburg, T. Uno, and A. Wolff. 2007. "Minimizing Intra-edge Crossings in

Wiring Diagrams and Public Transportation Maps." In *Graph Drawing*, edited by M. Kaufmann and D. Wagner, 270-81. Springer Berlin Heidelberg.

Chazelle, B., and H. Edelsbrunner. 1992. "An optimal algorithm for intersecting line segments in the plane." Review of. *J. ACM* 39 (1):1-54. doi: 10.1145/147508.147511.

Garland, K. 1994. *Mr. Beck's Underground Map*. Harrow, UK: Capital Transport Publishing.

Hong, S.-H., D. Merrick, and H. A. D. do Nascimento. 2006. "Automatic visualisation of metro maps." Review of. *Journal of Visual Languages & Computing* 17 (3):203-24. doi: 10.1016/j.jvlc.2005.09.001.

Hong, S.-H., D. Merrick, and H. A. D. Nascimento. 2005. "The Metro Map Layout Problem." In *Graph Drawing*, edited by J. Pach, 482-91. Springer Berlin Heidelberg.

Mulmuley, K. 1990. "A fast planar partition algorithm, I." Review of. *Journal of Symbolic Computation* 10 (3–4):253-80. doi: http://dx.doi.org/10.1016/S0747-7171(08)80064-8.

Nöllenburg, M. 2010. "An Improved Algorithm for the Metro-line Crossing Minimization Problem." In *Graph Drawing*, edited by D. Eppstein and E. R. Gansner, 381-92. Springer Berlin Heidelberg.

Nöllenburg, M., and A. Wolff. 2006. "A Mixed-Integer Program for Drawing High-Quality Metro Maps." In *Graph Drawing*, edited by P. Healy and N. S. Nikolov, 321-33. Springer Berlin Heidelberg.

Nonobe, K., and T. Ibaraki. 1998. "A tabu search approach to the constraint satisfaction problem as a general problem solver." Review of. *European Journal of Operational Research* 106 (2–3):599-623. doi: http://dx.doi.org/10.1016/S0377-2217(97)00294-4.

———. 2001. "An improved tabu search method for the weighted constraint satisfaction problem." Review of. *INFOR* 39 (1).

Ovenden, M. 2003. *Metro Maps of the World*. Harrow, UK: Capital Transport Publishing.

Stott, J. M., and P. Rodgers. 2004. Metro map layout using multicriteria optimization. Paper presented at the Information Visualisation, 2004. IV 2004. Proceedings. Eighth International Conference on, 14-16 July 2004.

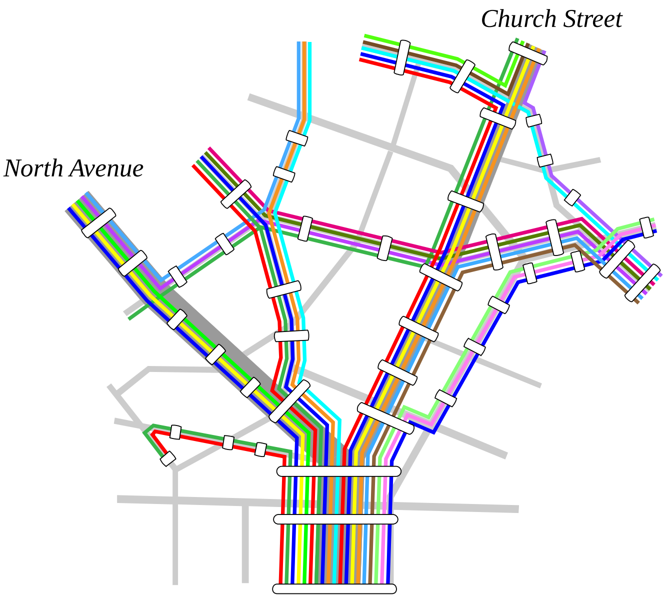
Stott, J. M., P. Rodgers, R. A. Burkhard, M. Meier, and M. T. J. Smis. 2005. Automatic layout of project plans using a metro map metaphor. Paper presented at the Information Visualisation, 2005. Proceedings. Ninth International Conference on, 6-8 July 2005.

List of Figures

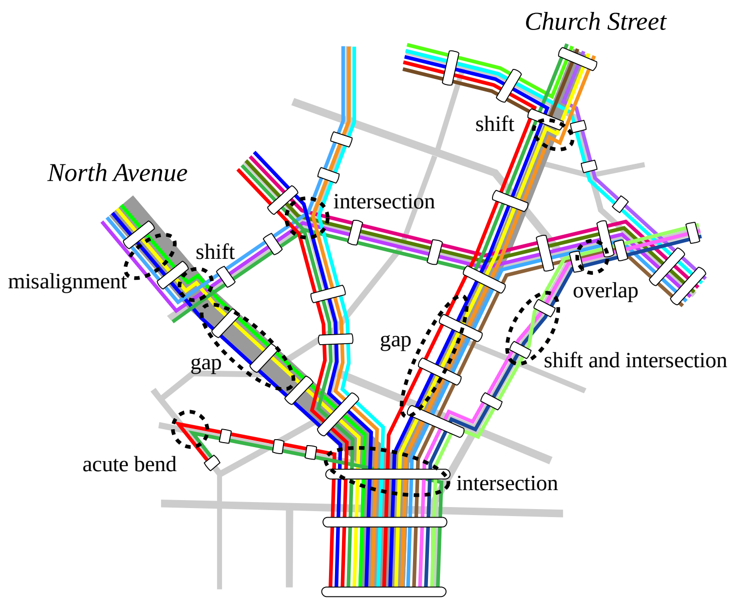
Figure 13. An example of bus route map obtained by the proposed method.

Figure 14. Manual revision of bus route map. (a) Bus stops and bus routes. (b) Bus route map obtained by the computational procedure. (c) Bus route map after revision.

List of Tables

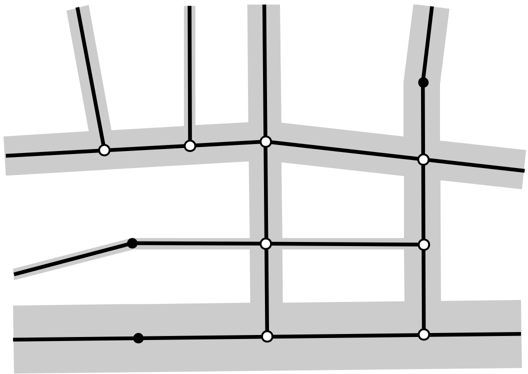

Church Street
North Avenue

(a)

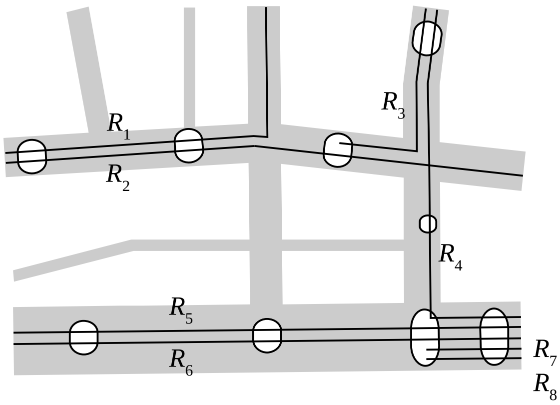

Church Street
North Avenue
shift
intersection
shift
misalignment
overlap
gap
gap
shift and intersection
acute bend
intersection

(b)

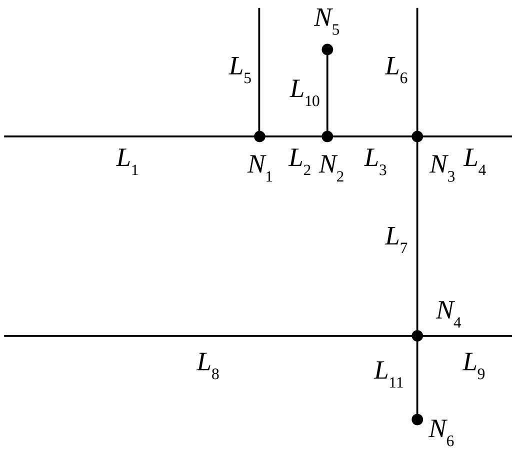
Figure 1



(a)

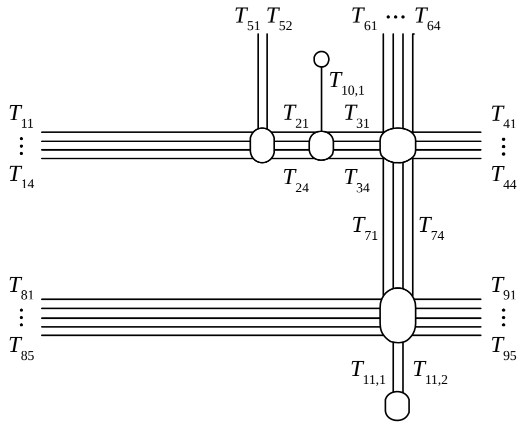

R_1
R_2
R_3
R_4
R_5
R_6
R_7
R_8

(b)


N_5
L_5
L_6
L_{10}
L_1
N_1
L_2
N_2
L_3
N_3
L_4
L_7
N_4
L_8
L_{11}
L_9
N_6

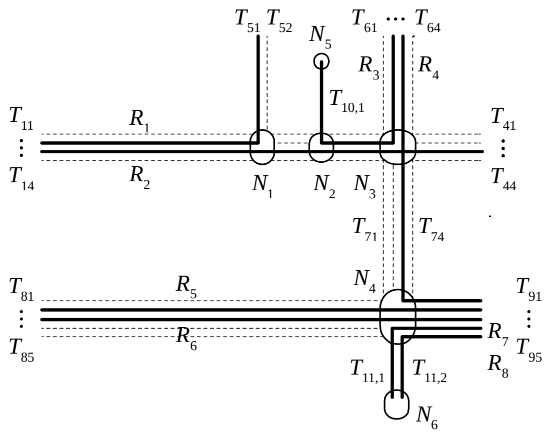
(c)

Figure 2



T_{51} T_{52}
T_{61} ⋯ T_{64}
$T_{10,1}$
T_{11}
T_{14}
T_{21}
T_{31}
T_{24}
T_{34}
T_{41}
T_{44}
T_{71}
T_{74}
T_{81}
T_{85}
T_{91}
T_{95}
$T_{11,1}$
$T_{11,2}$

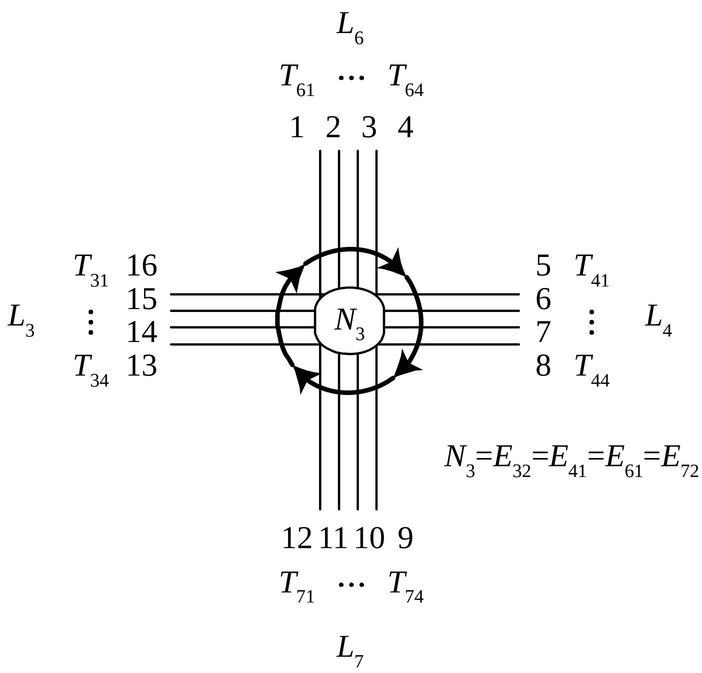
(a)



T_{51} T_{52}
N_5
T_{61} ⋯ T_{64}
R_3
R_4
$T_{10,1}$
T_{11}
T_{14}
R_1
R_2
N_1
N_2
N_3
T_{41}
T_{44}
T_{71}
T_{74}
N_4
T_{81}
T_{85}
R_5
R_6
T_{91}
T_{95}
R_7
R_8
$T_{11,1}$
$T_{11,2}$
N_6

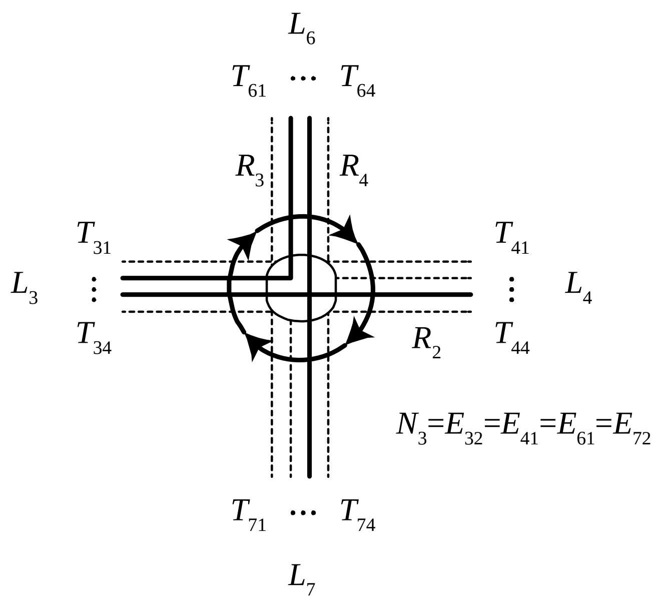
(b)

Figure 3



L_6
T_{61} ⋯ T_{64}
1 2 3 4
T_{31} 16
15
L_3 ⋮ 14
T_{34} 13
5 T_{41}
6
7 ⋮ L_4
8 T_{44}
N_3
N_3=E_{32}=E_{41}=E_{61}=E_{72}
12 11 10 9
T_{71} ⋯ T_{74}
L_7

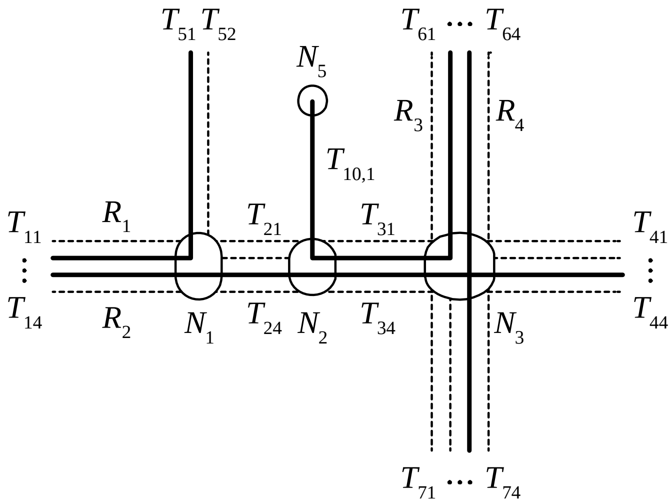
(a)



L_6
T_{61} ⋯ T_{64}
R_3
R_4
T_{31}
L_3 ⋮
T_{34}
T_{41}
⋮ L_4
T_{44}
R_2
N_3=E_{32}=E_{41}=E_{61}=E_{72}
T_{71} ⋯ T_{74}
L_7

(b)

Figure 4



$N_1 = E_{12} = E_{21} = E_{51}$

$N_2 = E_{22} = E_{31} = E_{10,1}$

$N_3 = E_{32} = E_{41} = E_{61} = E_{72}$

$N_5 = E_{10,2}$

Figure 5

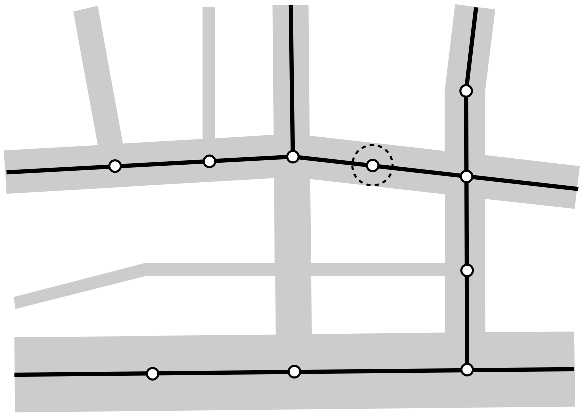


Figure 6

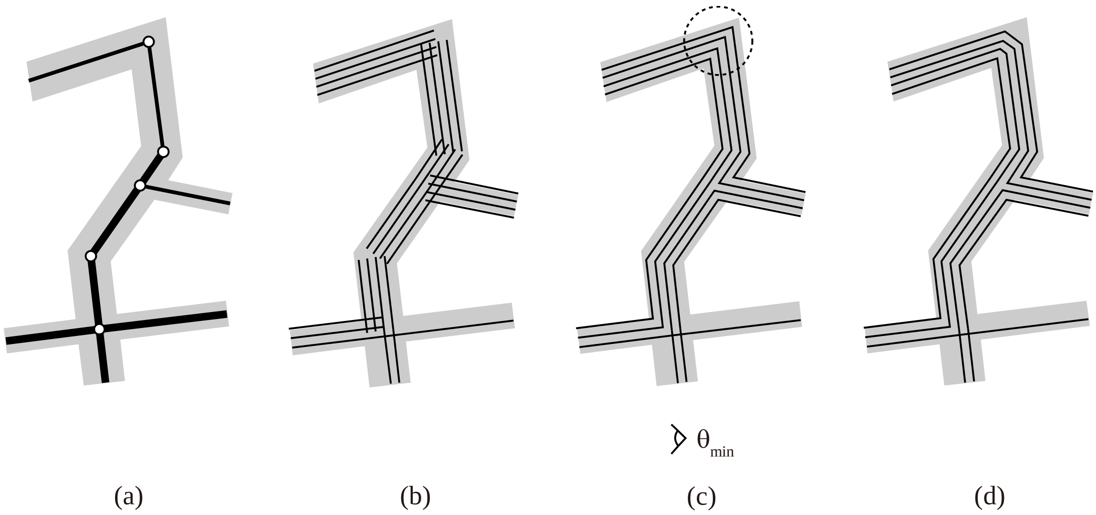


Figure 7

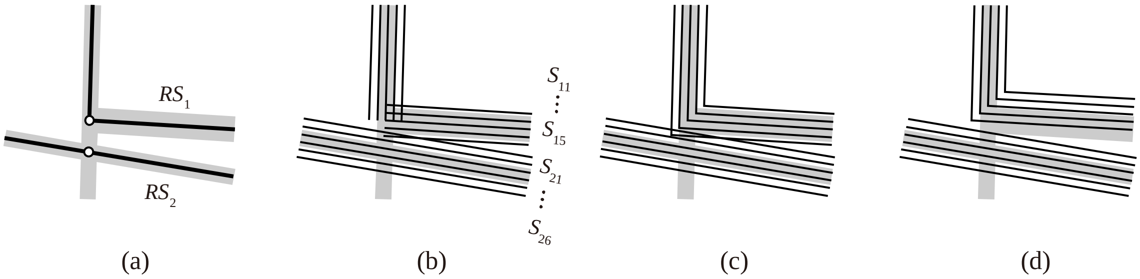
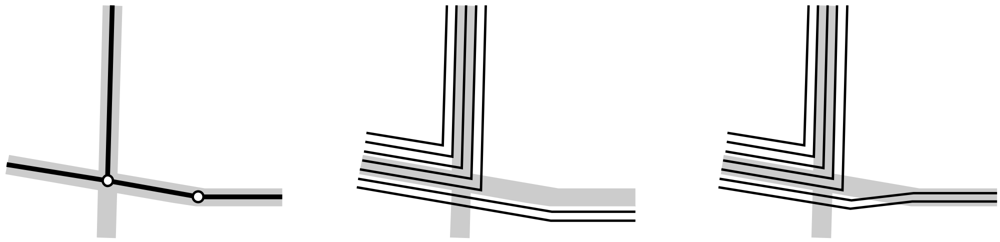


Figure 8



(a) (b) (c)

Figure 9

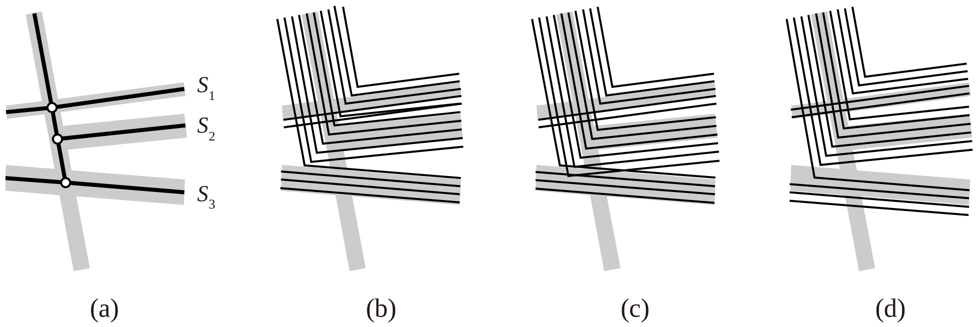


Figure 10

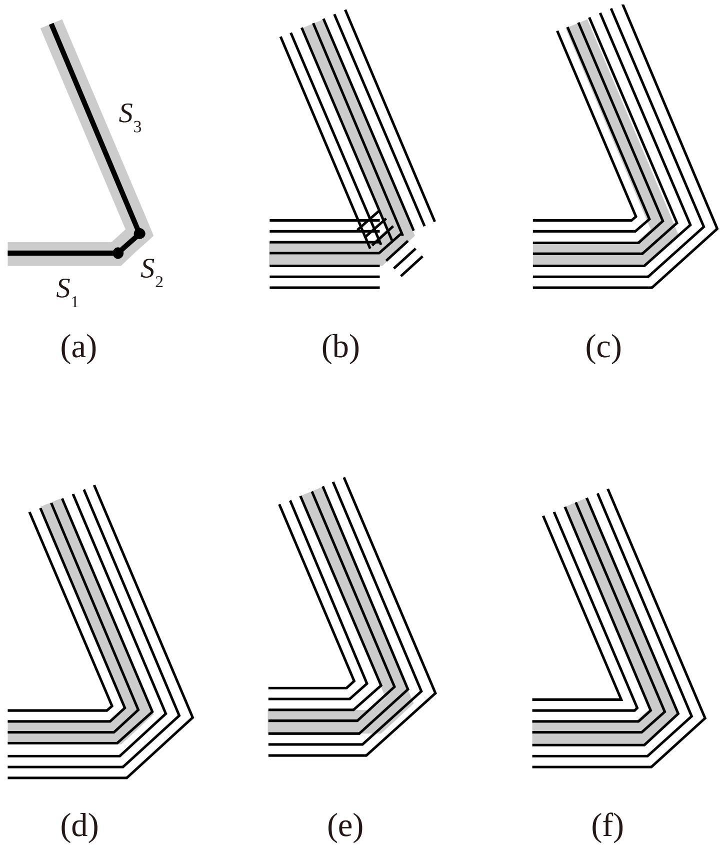


Figure 11

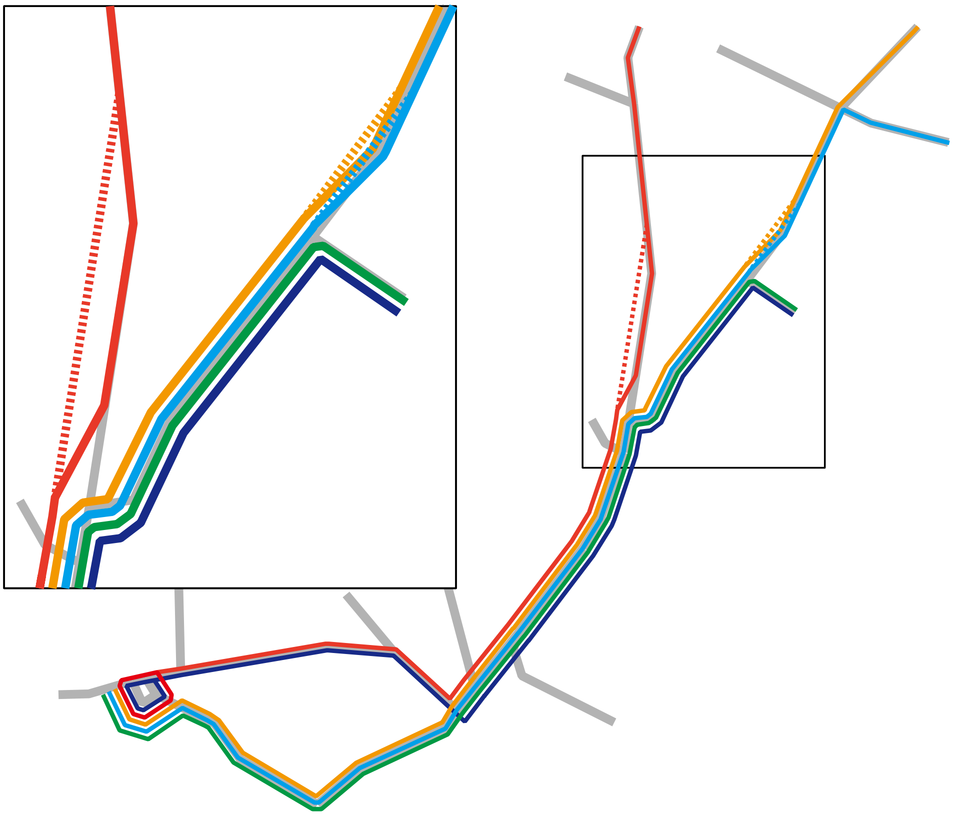


Figure 12

(a) (b)

Figure 13

Figure 14

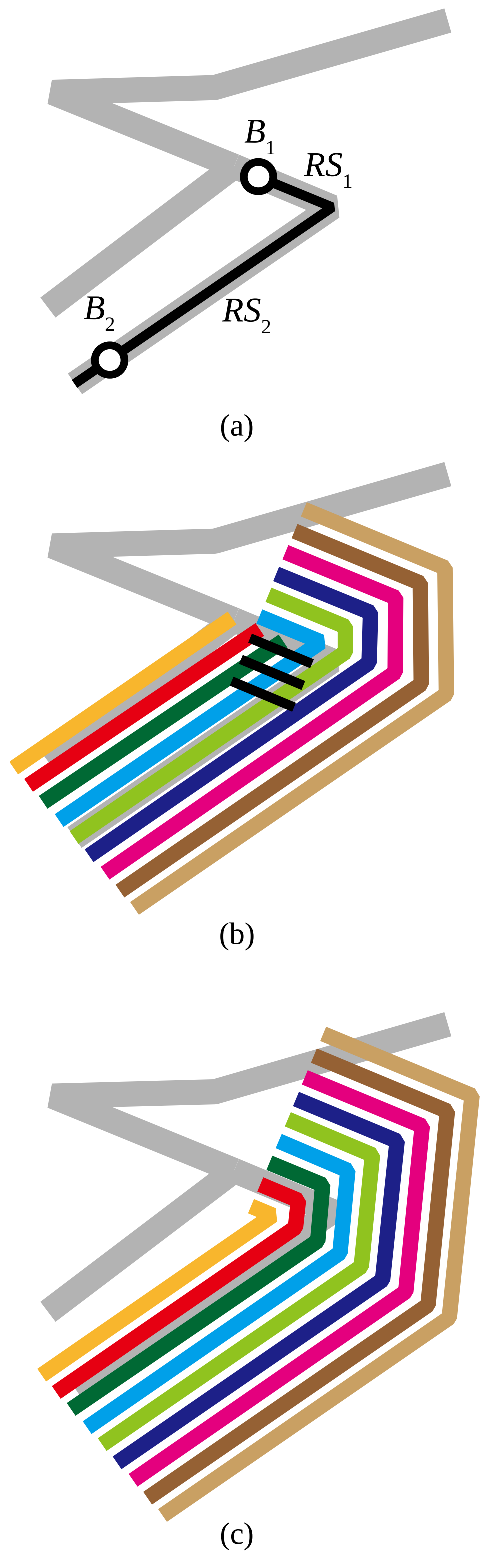


Figure 15

Table 1

Spatial objects

R_r	rth bus route ($1 \le r \le K$)
L_i	ith link ($1 \le i \le M+Q$)
N_j	jth node ($1 \le j \le M'+Q$)
E_{ie}	eth end of link i ($1 \le i \le 2$)
T_{ik}	kth track of link i

Constants

p_{ri}	Indicator of route R_r on link L_i ($1 \le r \le K$, $1 \le i \le M+Q$)
s_{ij}	Indicator of a street shared by L_i and L_j ($1 \le i$, $j \le M+Q$)
c_{iejl}	Relative location of T_{jl} around E_{ie} ($1 \le r \le K$, $1 \le i$, $j \le M$, $1 \le e \le 2$, $1 \le l \le n_j$)

Variables

x_{rikjl}	Indicator of assignment of R_r to T_{ik} and T_{jl} ($1 \le r \le K$, $1 \le i$, $j \le M$, $1 \le e \le 2$, $1 \le l \le n_j$)
y_{ike}	Relative location of track to which the same route is assigned as T_{ik} at E_{ie} ($1 \le i \le M$, $1 \le k \le n_i$, $1 \le e \le 2$)
z_{ike}	Indicator of a terminal of the route assigned to T_{ik} at E_{ie} ($1 \le i \le M+Q$, $1 \le k \le n_i$, $1 \le e \le 2$)
δ_{ike}	Indicator of a route shared by T_{ik} and a track connected at E_{ie} ($1 \le i \le M+Q$, $1 \le k \le n_i$, $1 \le e \le 2$)
φ_{ikme}	Measure of undesirable arrangement of routes assigned to T_{ik} and T_{im} at E_{ie}. ($1 \le i \le M$, $1 \le k \le n_i$, $1 \le m \le 2$, $1 \le e \le 2$)

Table 2

1: Route assignment needs to be symmetrical at every node.

2: Route assignment needs to be consistent at every node.

3: This constraint limits the shift of routes on streets. Every route cannot be assigned to tracks separated further distance β on the same street at each node of the cartographic road model.

4: Every route needs to be fully represented by a set of tracks.

5: Every track of terminals needs to be assigned a single route.

6: Every track of terminals needs to be assigned a single route.

7: Every track can be assigned at most a single route.

8: z_{ike} should be zero if the route assigned to T_{ik} does not terminate at E_{ie}.

9: Definition of δ_{ike}.

10: Definition of y_{ike}.

11: Definition of φ_{ikme}.

12: φ_{ikme} needs to be non-negative.