Automatic Generation Method for Geographically Accurate Bus Route Maps from Bus Stops

Sogo Mizutani
Nagoya Institute of Technology
Nagoya, Japan.
e-mail: s.mizutani.814@stn.nitech.ac.jp

Yonghwan Kim
Nagoya Institute of Technology
Nagoya, Japan.
e-mail: kim@nitech.ac.jp

Daisuke Yamamoto
Nagoya Institute of Technology
Nagoya, Japan
e-mail: daisuke@nitech.ac.jp

Naohisa Takahashi
Nagoya Institute of Technology
Nagoya, Japan
e-mail: naohisa@nitech.ac.jp

***Abstract*- There are two types of route maps: deformed route maps, which schematically show the locations and connections of stations and bus stops, and geographically accurate route maps, which are drawn on a map based on the actual locations of routes. While there have been many studies on the automatic generation of deformed route maps, only a few studies have focused on the automatic generation of geographically accurate route maps linked to road networks, such as bus route maps. Most conventional geographically accurate route maps are created manually, but mapping road data to bus route data is complicated by various constraints. In particular, it is necessary to draw parallel routes so that they do not overlap, and to consider the order in which they are arranged so that they cross each other minimally when turning right or left. In this study, we propose an automatic estimation method for bus route maps using bus stop coordinate series and strokes. This method generates bus stop nodes on the road network from the bus stop coordinate series. Furthermore, the bus route is estimated as a route with the least number of right/left turns between the bus stop nodes using the road priority search method. In addition, when drawing multiple routes, the order of placement of overlapping segments is dynamically determined so that there are fewer intersections when turning right or left. This enables the drawing of geographically accurate route maps with fewer intersections at right and left turns.**

Keywords-network; bus route map; stroke.

I. INTRODUCTION

When using public transportation, such as trains and buses, we often look at route maps to determine how to reach our destination. There are two types of route maps: deformed route maps that schematically show the locations and connections between stations and bus stops, and geographically accurate route maps that are drawn on a map based on the exact location of routes. Deformed route maps, as shown in Figure 1, do not require geographic accuracy in terms of direction and distance, but only that the relative station locations and connections on the route should be known. As shown in Figure 2, a geographically accurate route map can be drawn on a road map based on accurate location information to provide information on bus stops and facilities along the route. The problem of route placement is quite complex, especially when considering that multiple routes may be drawn on a single road.

Figure 1. Example of deformed route map. (Cited from the Nagoya City Transportation Bureau.)

Figure 2. Example of a geographically accurate route map. (Cited from the Nagoya City Transportation Bureau.)

In addition, many web maps, such as Google Maps have become popular in recent years, and it has become necessary to superimpose geographically accurate route maps on them. Furthermore, with the proliferation of the General Transit Feed Specification (GTFS) standard, transit system data for bus and subway routes has been made publicly available;

 ISBN: 978-1-61208-983-6

GTFS includes not only timetable data, but also the latitude and longitude coordinates of bus stops and connection relationships between routes.

The purpose of this study is to propose a method for estimating routes from bus stop coordinates and route system data in the GTFS, and automatically generate highly visible and geographically accurate bus route maps from the estimated results.

Although there have been many studies on the automatic generation of deformed route maps, only a few studies have focused on the automatic generation of geographically accurate route maps in conjunction with road networks, such as bus route maps. Most conventional geographically accurate route maps are created manually, but the task setup is complicated by the need to map the coordinates of roads and bus routes. In particular, it is necessary to draw parallel routes so that they do not overlap. It is also necessary to consider the order in which the routes are arranged to minimize their crossing when turning right or left. The following issues must be addressed to automatically generate geographically accurate bus route maps.

Issue 1) The latitude and longitude coordinates of bus stops published in the GTFS and elsewhere indicate the location of boarding points, which do not necessarily exist on the road. In contrast, a road network consists of nodes indicating intersections and turns and links connecting the nodes, and it is necessary to set starting and ending nodes to estimate routes. Therefore, to estimate a bus route, a bus stop node must be added to the road network based on the location coordinates of the bus stop.

Issue 2) To minimize the overlap of routes, a method to determine the order of placement of routes by brute force for each road link is considered, but such a method would be computation-intensive.

Issue 3) To improve the visibility of the route maps, it is necessary to draw the routes by shifting them from one another, but the order in which the routes are arranged results in a route map with many intersections. Therefore, it is necessary to dynamically determine the order in which the routes with the fewest intersections are placed on the map.

Therefore, we propose a system with the following features.

Feature 1) Route generation function

From the road network and bus stop coordinates, a bus stop node is added to the nearest point on the road link closest to the bus stop. In addition, the routes between adjacent bus stop nodes are estimated by a way-first search and used as bus routes.

Feature 2) Bus stroke/bus stroke fragment generation function

This realizes the function of generating Bus Stroke (BS) using road strokes and bus routes, and generates the overlap between bus routes as a Bus Stroke Fragment (BSF). The BS/BSF model is used to efficiently determine the order of placement.

Feature 3) Route placement function

This function determines the order of placement when drawing multiple routes staggered on a single road link.

In this paper, we describe related studies in Section 2, describe the proposed system in Section 3, discuss the evaluation of the proposed system in Section 4, and summarize in Section 5.

II. RELATED WORK

Many of the existing methods for drawing route maps are intended for train route maps. Onda et al. [1] proposed a method for automatic generation of railroad route maps for Tokyo's train route network. Bast et al. [2] proposed a method for automatic generation of geo-informative railroad route maps with station connectivity and route location as inputs. Other studies [3][4][5] have been conducted to improve the quality of deformed subway route maps.

Wang and Peng [6] proposed an interactive editing system for subway route map layouts; Lloyd et al. [7] proposed a method for color coding subway route maps.

Some studies are based on strokes. A stroke is a grouping of road networks based on cognitive psychology and represents a road [8][9] that follows a path. Zhang et al. [10] have realized a total road drawing by selecting distinctive roads based on their connection relations. As for the use of road strokes, studies have been made on total road drawing. Road drawing is a method to draw only major roads in a road network based on the length of road strokes. Methods to realize road drawing from facility search results [11][12] and a method to realize road drawing in the Fisheye view format [13] have been proposed. A stroke-based path finding method [14] has also been proposed.

III. PROPOSED SYSTEM

Figure 3 shows the configuration of the proposed system. In the route generation function, bus stop nodes are generated from bus information and road data published in the GTFS, and routes are generated by routing the bus stop nodes. Next, BS/BSF is generated from the generated route data; BS/BSF is explained in detail in Section C. Next, the route placement function determines the placement order of overlapping BSFs. Finally, the system draws the routes based on the order of placement and provides them to the user.

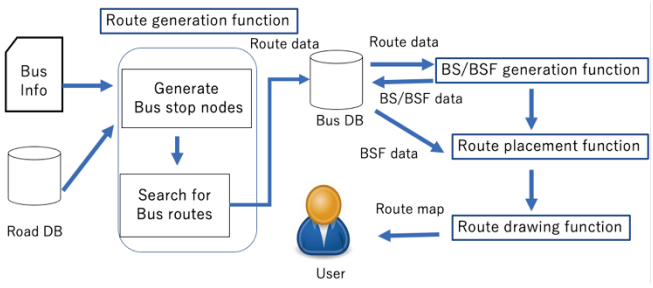

Figure 3. System configuration.

A. Data format

The data and definitions of terms used in this study are described below.

1) Road data format

The road data provided by OpenStreetMap consists of nodes and links. A node represents an intersection on the road network and a road link connects nodes along a point, which is a series of latitude and longitude coordinates. A link connects a start node to an end node, and the road network is a directed graph, given the direction of the road. Road geometry is represented by geometry-type arcs along a point sequence.

OpenStreetMap also provides data representing road types, such as highways, national highways, and pedestrian-only roads, which are stored as road classes.

2) Stroke data format

A stroke is a set of road networks grouped according to cognitive psychology. The example on the left in Figure 4 shows a road network consisting of a set of links between nodes. The example on the right of Figure 4 shows a stroke network consisting of color-coded strokes.

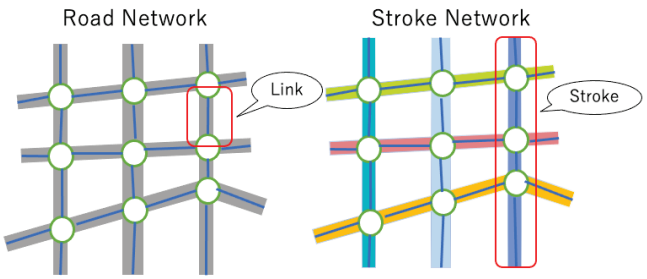

Figure 4. Examples of a road and stroke network.

3) Bus data format

This study uses bus data in the GTFS-JP format, which were released as open data by the Nagoya City Transportation Bureau in 2017. The following three items were used in this study.

Bus stop data
: The name, identification (ID), and latitude and longitude coordinates of the bus stop are stored.

System data
: The operational data for each system is stored. The system table contains information on the start and end points of the service section within the system, apart from other information. The system data is uniquely identified by system, route code, direction codes.

Bus stop series
: The series of bus stops that pass through the system from the start point to the end of the system's service section are stored.

B. Route generation function

The route generation function generates bus route link data from road data, stroke data, and bus information. The main flow generates bus stop nodes from the road network and bus stop series, creates split links, and searches for routes between bus stops.

1) Generation of bus stop nodes

Using the latitude and longitude coordinates of the bus stop, and the road class as inputs, a bus stop node is generated on the road link closest to it in the specified road class.

The method of generating bus stop nodes is shown below. Here, we used PostGIS functions, including ST_DWthin, ST_Distance, ST_LineLocatePoint, and ST_LineInterpolate Point functions.

$L = (l_1, l_2, \cdots, l_i)$: Link set
$Class$: Road class
Bus_{point} : latitude and longitude coordinates of bus stops
Bus_{node} : bus stop node

Step 1) From the link set L, find the nearest link set within a radius of 20 m from Bus_{point} using the ST_DWthin function. However, exclude links whose road class is expressway or a connecting road to an expressway.

Step 2) Find the link with the shortest distance in the set of nearest neighbor links using the ST_Distance function. This link shall be the nearest neighbor link.

Step 3) Find the nearest point in the nearest link from Bus_{point} and get the ratio r of the nearest point to the start and end points of the link using ST_LineLocatePoint function. The obtained ratio r is stored in a table as Bus_{node} using the ST_LineInterpolatePoint function. Table 1 shows the data format of the bus stop nodes.

TABLE1. BUS STOP NODE TABLE

Column name	*Data type*	*Explanation*
id	Integer	Bus stop ID
link_id	Integer	Nearest neighbor road link ID
node_lat	Double	Latitude of bus stop node
node_lng	Double	Longitude of bus stop node
ratio	Double	Percentage on link

2) Creating split links

To perform a route search, it is necessary to split road links using the generated bus stop nodes. In this study, a split link is defined as a set of links in which a road link is divided by a bus stop node. In the example in Figure 5, Link2 of the road link is divided by a bus stop node, increasing the number of links from three to four.

 ISBN: 978-1-61208-983-6

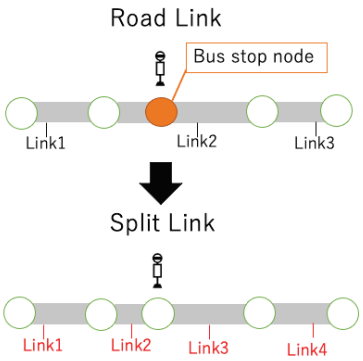

Figure 5. Split link.

3) Route estimation

The variables used for route estimation are defined below, followed by the route search procedure.

ID $= (id_1, id_2, \cdots, id_n)$: List of bus stop IDs for the route to be estimated
$Node = (N_1, N_2, \cdots, N_n)$: Node ID list
$V \ni (s, g)$: Combination of start and end points
$R = (r_1, r_2, \cdots, r_n)$: Route data list

Step 1) Obtain the IDs for the specified system, route, and direction code from the bus stop database.
Step 2) Obtain the bus stop node corresponding to the ID from the bus stop database and add it to the Node.
Step 3) In v $= (N_i, N_{i+1}) \in V$, checks if there is a record, $(s, g) = (id_i, id_{i+1})$ *or* (id_{i+1}, id_i) in the route table.
Step 4) If it exists, the acquired route data is assumed to be r_i.
Step 5) If not, a min stroke shortest path search in v is performed to find r_i.
Step 6) If all routes are found, R is stored in the route table.

In step 3, by checking whether a route is already in the table, the search time can be reduced by finding sections where a route search is not needed.

In step 5, a min stroke shortest path search is performed. Min stroke shortest search is a method that searches for a route with the shortest distance among routes with the fewest number of left-right turns.

C. Bus stroke/bus stroke fragment generation function

1) Bus stroke

In general, bus routes are often set up on routes with few left-right turns. Therefore, expressing them as a set of strokes requires less data and computation. In particular, when considering the determination of whether multiple routes are running side by side, it is possible to reduce the number of calculations if the determination is made on a stroke-by-stroke basis rather than on a road-by-road basis. Figure 6 shows an example of converting a link set into a BS set. In this example, the red bus routes are represented as {BS1, BS2, BS3}.

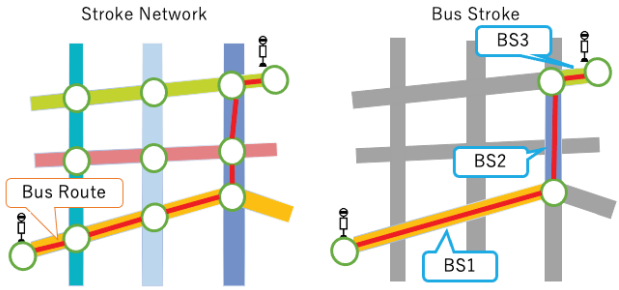

Figure 6. Example of bus stroke generation.

2) Bus stroke fragment

To generate bus strokes for multiple parallel bus routes, some strokes need to be divided into shorter strokes. The divided strokes are called BSFs. Figure 7 shows an example of the division of two routes into BSFs. In this example, BS1 in red is broken down into BSF1 and BSF2. BS2 in blue also includes BSF2. When expressed in BSs, blue routes are represented as {BS1, BS2, BS3} and red routes as {BS1, BS2, BS3}. When expressed in BSFs, blue routes are expressed as {BSF5, BSF2, BSF6} and red routes as {BSF1, BSF2, BSF3, BSF4}. Some BSFs are shared by each route.

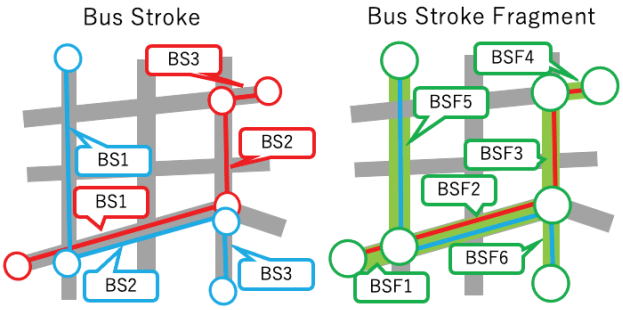

Figure 7. Example of bus stroke fragment generation.

D. Route placement functions

1) Route placement function

The route placement function determines the order of placement when drawing multiple routes staggered on a single road link. This section describes a method for determining the placement order of multiple routes using route BSFs as an input. The procedure is as follows.

Step 1) First, BSF series are obtained from two routes.
Step 2) From the origin side of the routes, determine the order of BSF placement according to Rules 1 and 2 described below. However, if

necessary, the previous placement order can be reused.

Step 3) Next, to add one unprocessed line to the current result, obtain the placement order according to Rules 1 and 2 described below, as in Step 2.

Step 4) Repeat Step 3 until there are no more unprocessed routes.

In Steps 2 and 3 above, the following two rules are applied to determine the placement order.

Rule 1) The placement order of the target BSF is based on the angle formed by the target BSF and the previous BSF, as shown in Figure 8.

Rule 2) For routes where the placement order is not uniquely determined, the BSFs are determined backward until the placement order is determined.

Figure 8 shows an example of route placement. In Rule 1, the order of placement of routes is determined by the angle between the previous BSF and the target BSF on the start point. The PostGIS functions ST_Azimuth and degree were used to calculate the angles: the ST_Azimuth function returns the rightward radians with respect to north, and the degree function converts radians to degrees. These are used to calculate the angle between the previous BSF and the target BSF (Base). In the example in Figure 8, (Route_a, Route_b, Route_c, Base) = (0°, 270°, 225°, 90°). Here, sorted in a descending order based on the angle with the target BSF, the result is (Base, Route_a, Route_b, Route_c). Therefore, the order of placement of the target BSF is Route_a, Route_b, and Route_c.

In the example on the right of Figure 8, both the previous BSF of the blue route and the previous BSF of the yellow route have the same angle. Therefore, Rule 1 cannot be applied as it is in the example on the left side of Figure 8. Therefore, in Rule 2, the placement order of one more previous BSF is directly used as the placement order of the target BSF.

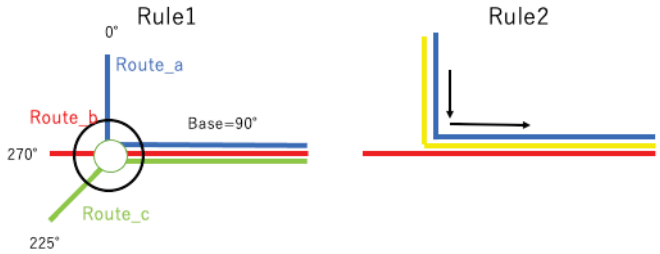

Figure 8. Rules for route placement.

2) Route draw function.

The route draw function draws a route on a web map. In this study, OpenStreetMap was used as the map data and Leaflet was used as the map control library.

The drawing procedure is as follows. First, the result of the route placement function and the geometry of the routes are stored in a GeoJSON format file, and the Leaflet library draws the route on the web map based on the GeoJSON format file. Here, to prevent the overlapping of routes, the Leaflet Polyline Offset is used to shift and draw the overlapping sections of the routes.

IV. EXPERIMENTAL RESULTS

A. Route estimation based on bus stop series

The proposed system estimates routes on the road network from a bus stop series. Therefore, we evaluated the accuracy of the route estimation function by comparing actual bus routes and routes estimated by the proposed system. The evaluation measure is the matching ratio M with the actual route map, and the following evaluation formula (1) is used. Forty bus route maps of the Nagoya City Transportation Bureau were used for the evaluation.

$$M = \frac{Number\ of\ matched\ routes}{Number\ of\ actual\ routes} \times 100 \qquad (1)$$

Table 2 shows the results of the evaluation experiments. The overall estimation accuracy was 93.2%. The results show that the proposed system can estimate bus routes with a high accuracy. The most common example of estimation failure was an estimation error near a bus terminal. This is because bus terminals have many bus stops, with different entrances and exits, which causes the system to generate bus stop nodes on the wrong road links.

TABLE2. RESULTS OF EVALUATION EXPERIMENTS

Total number of routes	***Matched routes***	***Matching ratio M (%)***
890	838	93.2

B. Qualitative evaluation in the results for 30 routes

In this evaluation, the automatically generated routes were evaluated qualitatively. The evaluation criterion was visibility of the route map, which was performed visually by the authors.

Figure 9 shows the results of drawing 30 routes. In this example, all routes were drawn correctly. However, there are some issues, as described below. Although the route drawing function distinguishes routes by color coding, the visibility is reduced because many routes have similar colors. In particular, adjacent routes should be drawn in different colors whenever possible. In addition, BSFs with many overlapping routes and those near bus terminals may be drawn at a distance from the actual road. When there are more overlapping routes near bus terminals, it is necessary to build a drawing function that maintains better visibility.

 ISBN: 978-1-61208-983-6

V. CONCLUSION

In this study, we proposed a method for automatic generation of bus route maps using bus stop series and strokes. This method generates bus stop nodes from the coordinates of the bus stop series and the road networks and estimates bus routes using a min stroke shortest path search method that minimizes the number of right/left turns between bus stop nodes. The estimation accuracy was 93.2%. We also proposed an efficient method for determining the order of placement by generating BSFs. This method dynamically determines the placement order of overlapping segments when drawing multiple routes and realizes map drawing of routes with fewer right/left turn crossings. Future works can include the construction of a drawing function that maintains visibility near bus terminals and when there are several overlapping routes.

ACKNOWLEDGMENT

This work was supported by JSPS KAKENHI Grant Numbers JP19H04115 and JP21K19766.

REFERENCES

[1] M. Onda, M. Moriguchi, and K. Imai, “Automatic Drawing for Metro Maps in Tokyo,” IEICE-COMP / IPSJ-AL, 2017-AL-163, Vol.13, pp.1-8, 2017.

[2] H. Bast, P. Brosi, and S. Storandt, “Efficient Generation of Geographically Accurate Transit Maps,” In Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information, pp.13-22, 2018. DOI: 10.1145/3337790

[3] J. Stott, P. Rodgers, J. C. Martinez-Ovando, and S. G. Walker, “Automatic metro map layout using multicriteria optimization,” IEEE Transactions on Visualization and Computer Graphics, 17(1), 101-114, 2010. DOI: 10.1109/TVCG.2010.24

[4] S. H. Hong, D. Merrick, and H. A. Do Nascimento, “The metro map layout problem,” In Proceedings of the International Symposium on Graph Drawing 2004, pp. 482-491, Springer, 2004. DOI: /10.1007/978-3-540-31843-9_50

[5] J. M. Stott and P. Rodgers, “Metro map layout using multicriteria optimization,” In Proceedings of the Eighth International Conference on Information Visualization, pp. 355-362, 2004. DOI: 10.1109/IV.2004.1320168

[6] Y. S. Wang and W. Y Peng, “Interactive metro map editing,” IEEE transactions on visualization and computer graphics, Vol 22, No. 2, pp. 1115-1126, 2016. DOI: 10.1109/TVCG.2015.2430290

[7] P. B. Lloyd, P. Rodgers, and M. J. Roberts, “Metro map colour-coding: Effect on usability in route tracing,” In Proceedings of the International conference on theory and application of diagrams, pp. 411-428, Springer, 2018. DOI: 10.1007/978-3-319-91376-6_38

[8] R. Thomson and R. Brooks, “Efficient generalisation and abstraction of network data using perceptual grouping,” In Proceedings of the 5th International Conference on GeoComputation, pp. 23–25, 2000.

[9] R. Thomson and D. Richardson, “’Good continuation’ principle of perceptual organization applied to the generalization of road networks,” In Proceedings of the 19th International Cartographic Conference, pp. 1215–1223, 1999.

[10] Q. Zhang, “Road network generalization based on connection analysis,” In Proceedings of the 11th International Symposium on Spatial Data Handling, pp. 343–353, 2005. DOI: 10.1007/3-540-26772-7_26

[11] D. Yamamoto, M. Murase, and N. Takahashi, “On-Demand Generalization of Road Networks based on Facility Search Results,” IEICE Transactions on Information and System, Vol. E102-D, No. 1, pp. 99-103, 2019. DOI: 10.1587/transinf.2017EDP7405

[12] M. Murase, D. Yamamoto, and N. Takahashi, “On-demand Generalization of Guide Maps with Road Networks and Category-based Web Search, Results,” In Proceedings of the 14th International Symposium on Web and Wireless Geographical Information Systems, Vol. 19, pp. 53-70, 2015. DOI: 10.1007/978-3-319-18251-3_4

[13] D. Yamamoto, S. Ozeki, and N. Takahashi, “Focus+Glue+Context: An Improved Fisheye Approach for Web Map Services,” In Proceedings of the ACM SIGSPATIAL GIS 2009, pp.101-110, 2009. DOI: 10.1145/1653771.1653788

[14] Yuki Hiura, (supervisor: Daisuke Yamamoto), “Proposal of an efficient nth min stroke shortest path search method,” Master’s thesis, Nagoya Institute of Technology, 2020. (In Japanese)

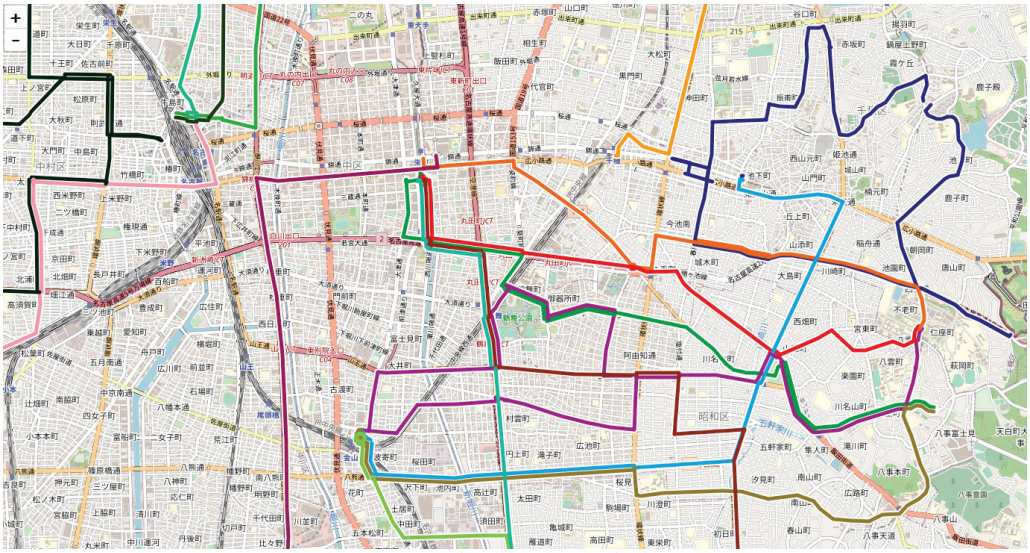

Figure 9. Example of drawing results for 30 routes.