A Proposal of Road Network Hierarchization Method Based on Betweenness Centrality for Application to Vehicle Routing Problems

Masayuki Shimizu
Graduate School of Information Science and Technology, Hokkaido University
Sapporo, Hokkaido, Japan
email: s.masayuki_0426@ist.hokudai.ac.jp

Soichiro Yokoyama
Faculty of Information Science and Technology, Hokkaido University
Sapporo, Hokkaido, Japan
email: yokoyama@ist.hokudai.ac.jp

Tomohisa Yamashita
Faculty of Information Science and Technology, Hokkaido University
Sapporo, Hokkaido, Japan
email: tomohisa@ist.hokudai.ac.jp

Hidenori Kawamura
Faculty of Information Science and Technology, Hokkaido University
Sapporo, Hokkaido, Japan
email: kawamura@ist.hokudai.ac.jp

***Abstract*—In this study, we propose a road network hierarchization method for pathfinding that takes into account drivers' avoidance of narrow roads. The proposed method identifies nodes with high betweenness centrality, one of the centrality measures in network analysis, to connect disconnected subnetworks within each hierarchical level. To validate the effectiveness of the proposed method, multiple driver preferences regarding narrow roads are prepared, and computational experiments are conducted on a road network covering a 14 km square area in central Sapporo. Compared to performing pathfinding without hierarchical networks, the calculation time of individual pathfinding was reduced to 4-6% with the previous method and to 3-6% with the proposed method. Additionally, by using the proposed method, the average cost of routes improved compared to the previous method, and even when compared to the minimum cost paths, the worsening of route costs was about 4-9%. From the above, the superiority of the proposed method, which connects disconnected subnetworks using betweenness centrality, was confirmed.**

Keywords-Road Network Hierachization; Pathfinding; Vehicle Routing Problem.

I. INTRODUCTION

With the recent proliferation of online shopping, delivery companies are required to deliver goods to customers efficiently. In previous research, the problem of efficiently delivering goods to customers has been treated as a vehicle routing problem, in which the optimal route is identified to satisfy multiple constraints while minimizing delivery time as a delivery cost. The objective of the vehicle routing problem is to efficiently deliver goods to nearby customers by coordinating the timing of multiple delivery vehicles, to decrease travel distance, and to reduce the overall delivery time by reducing the number of deliveries.

When optimizing delivery plans, it is essential to have distance information for routes between a delivery base and customers. In order to efficiently compute a route between two arbitrary points, routing algorithms such as the Dijkstra algorithm [1] and the A* algorithm [2] have been developed, but when the number of customers is large, it is difficult to compute all the necessary routing information in a practical time. For this reason, the Euclidean distance is often used because it is easy to calculate. However, as the Euclidean distance does not take into account the actual path, there can be a discrepancy with the actual travel distance and the required time, making it sometimes inappropriate for solving real-world delivery planning problems. Therefore, there is a demand for an approach that can solve a vast number of pathfinding tasks within a practical timeframe. In addition, when the vehicle routing problem is applied to actual deliveries, various factors such as weather, road conditions, and traffic congestion must be incorporated into the vehicle routing problem.

We regard the kerosene delivery planning as an inventory routing problem and formulate it as an optimization problem that reflects the actual consumption of kerosene and have advanced research on an approximate solution method using Tabu Search [3]. In pathfinding for kerosene delivery planning, it is necessary not only to reduce the computation time for pathfinding but also to take into account the delivery driver's avoidance of snow-covered narrow roads. Kerosene delivery is mainly conducted during the busy winter season in cold regions, and tank trucks are required to travel on snow-covered roads. However, many narrow roads are not cleared of snow during the winter. Tank trucks carrying kerosene may get stuck on narrow roads that have not been cleared of snow, and delivery drivers may choose routes that avoid narrow roads depending on the road conditions. Therefore, it is necessary to reflect the behavior of delivery drivers in route planning, taking into account their tendency to avoid narrow roads under conditions of snow accumulation and snowfall.

In this paper, an application to a vehicle routing problem is assumed, where delivery routes are coordinated and planned over a long period of time in a specific region. This requires performing hundreds of thousands of pathfinding tasks, taking into consideration the driver preferences.

 ISBN: 978-1-68558-132-9

One way to speed up pathfinding is to improve the efficiency of existing pathfinding algorithms. However, in cases where pathfinding needs to be performed hundreds of thousands of times, it becomes challenging to process all pathfinding tasks within a practical time frame, even with an efficient pathfinding algorithm.

While improving the efficiency of existing pathfinding algorithms aims for exact solutions, there are also developments in methods that preprocess graphs as an approximate solution approach. These methods accelerate pathfinding without guaranteeing the optimality of solutions. The vehicle routing problem assumed in this study involves long-term delivery planning in a specific region, requiring repeated use of the road network in that area. For problems with these characteristics, it is believed that a road network hierarchization method is effective. The reasons for this are as follows:

- Once a hierarchical network is constructed, it can be reused as long as the network remains unchanged.
- The use of simplified networks is expected to reduce the computation time for pathfinding.
- By setting the link costs during pathfinding individually according to the driver preferences, pathfinding that takes into account various driver preferences can be performed on the same hierarchical network.

Based on the prior method that implements the degree of avoidance of narrow roads in the form of driver preferences in a hierarchical pathfinding algorithm [4], this paper proposes a road network hierarchization method aimed at optimizing the delivery planning problem.

This paper is structured as follows. In Section 2, we review related work on road network hierarchization methods to clarify the positioning of this paper. In Section 3, we propose a method for connecting subnetworks based on betweenness centrality in road network hierarchization. In Section 4, focusing on actual urban road networks, the computational time for pathfinding and the cost of routes obtained using both the proposed method and previous methods are compared to confirm the effectiveness of our approach. Finally, in Section 5, the paper is concluded.

II. Related Work

A. Hierarchical Pathfinding

In hierarchical pathfinding algorithms, there are primarily two methods of road network hierarchization: one is classification-based hierarchization, and the other is aggregation-based hierarchization. In this Section, we will overview the previous methods for these two types of hierarchization.

1) Classification-Based Hierarchization: Classification-based hierarchization uses the attributes of each road, such as road type (expressway, national road, prefectural road, city road, etc.) and the number of lanes, to divide the hierarchical level. Fukuda et al. [4] define fixed attributes, which are the same for every driver, and variable costs, which allow for differences among drivers. They propose a hierarchical pathfinding algorithm that employs fixed attributes for the hierarchization of the road network and uses variable costs for pathfinding. Fukuda et al., based on the findings of previous research [5], which suggests that general drivers tend to prefer roads with a greater number of lanes, have assigned lane count as a fixed attribute. In this hierarchization based on number of lanes, the initially classified upper-level networks often become disconnected. However, their proposed hierarchical pathfinding algorithm prohibits searching from the upper-level networks to the lower-level networks. This leads to failures in pathfinding when any of the networks at the upper levels are disconnected. Therefore, it is necessary to correct these upper-level networks to resolve their disconnectedness. In the method by Fukuda et al. (hereinafter referred to as the 'previous method'), they define a threshold H_{max} to control the extent to which disconnected subnetworks are connected during corrections. To validate the effectiveness of the hierarchization method, they performed pathfinding by representing the driver's avoidance of narrow roads as a preference. As a result, it was confirmed that effective pathfinding results could be outputted using the same hierarchical network for three types of drivers with varying degrees of avoidance of narrow roads. However, a limitation of this method is that the cost of the routes outputted can significantly worsen compared to the minimum cost routes.

2) Aggregation-Based Hierarchization: Aggregation-based hierarchization [6][7][8] applies community detection methods to road networks, treating each detected community as a new node to construct a upper-level network. Mahyar et al. [7] employ the Louvain method [9], known for its speed and accuracy, for community detection as part of their hierarchization process. Additionally, in their work, to realize hierarchization that takes into account congestion conditions, they assign travel times or traffic conditions to the link costs of the network. This allows for grouping nodes with similar congestion conditions into the same community, making it possible to achieve hierarchization that takes into account the state of congestion.

Classification-based hierarchization has the advantage of allowing pathfinding that takes into account the road structure, but it has the disadvantage of difficulty in reflecting dynamic conditions during the hierarchization process. On the other hand, aggregation-based hierarchization has the advantage of being able to reflect dynamic conditions during hierarchization, but it has the disadvantage of making pathfinding that considers road structure difficult due to changes in topology. Therefore, it is necessary to choose the method of hierarchization based on the objective. However, this paper focuses on driver preferences based on static attributes such as distance and road width and does not consider dynamic changes in the network (such as road closures due to construction) or traffic conditions. Consequently, we examine a classification-based hierarchization method.

 ISBN: 978-1-68558-132-9

III. Our Proposed Method

A. Positioning of the Proposed Method

For application to the assumed delivery planning problems, this paper aims to develop a road network hierarchization method that reduces the sum of the construction time of a hierarchical road network and the computation time for hundreds of thousands of pathfinding tasks and takes into account various driver preferences.

We propose a road network hierarchization method that consider the importance of individual nodes in the network base on our previous work [10]. This method is based on the hypothesis that by constructing the upper-level networks around nodes of high importance, pathfinding using the hierarchical road network can effectively pass through these important nodes, and, as a result, calculate routes with less deterioration in the cost of routes.

When constructing a hierarchical road network, the upper-level networks are extracted based on road attributes such as the number of lanes, but these networks often become disconnected. However, The hierarchical pathfinding algorithm proposed by the previous method [4] is adopted in our approach, and this algorithm prohibits searching from upper-level networks to lower-level networks. As a result, pathfinding failures can occur when networks at upper levels are disconnected. To address this, the disconnections are resolved by adding nodes and links to the upper-level networks. The major difference between our method and the previous method for constructing the hierarchical network is that the nodes to be added are selected based on their importance in our approach.

B. Problem Formulation

In this paper, a problem that extends the general shortest path problem is addressed, but first, the formulation of the general shortest path problem is explained. Given a graph $G = (V, E)$ and the weights of each link $e \in E$ denoted as $w_e \in \mathbb{R}_+$, a sequence of vertices $P = (v_1, v_2, ..., v_k)$ satisfying $e_i = (v_i, v_{i+1}) \in E, i = 1, ..., k-1$ is called a path. The variable x_e indicates whether a link e is included in the path: $x_e = 1$ if it is included, and $x_e = 0$ otherwise. The problem of finding the shortest path from a given start node $s \in V$ to an target node $t \in V$ can then be formulated as follows.

$$\min \sum_{e \in E} w_e x_e \tag{1}$$

$$\text{s.t.} \sum_{e \in \delta^+(v)} x_e - \sum_{e \in \delta^-(v)} x_e = 0, \quad \forall v \in V \setminus \{s, t\} \tag{2}$$

$$\sum_{e \in \delta^+(s)} x_e = 1 \tag{3}$$

$$\sum_{e \in \delta^-(t)} x_e = 1 \tag{4}$$

$$x_e \in \{0, 1\}, \quad \forall e \in E \tag{5}$$

Here, $\delta^+(v)$ denotes the set of links that have vertex v as their starting point, and $\delta^-(v)$ denotes the set of links that have vertex v as their endpoint. The constraint in (2) represents that, for each visited vertex v, exactly one incoming link and one outgoing link are selected. The constraints in (3) and (4) signify that exactly one link leaving the start node s and one link entering the target node t are selected.

In this paper, a classification-based hierarchical road network is used. In constructing the hierarchical road network, a function $L(e)$ is defined to indicate the hierarchical level to which each link e in the original network G belongs. Then, based on the set of links E^n belonging to hierarchical level $n(n = 1, 2, ..., N)$ and the set of nodes V^n that are endpoints of links $e \in E^n$, the network G^n for hierarchical level n is constructed.

In pathfinding using a classification-based hierarchical road network, starting from the original network G, a progressive transition between hierarchical levels is made. At each level, it can be considered that the shortest path problem specific to that level is being solved. Finally, the paths calculated at each level are concatenated and outputted. Here, the path at each hierarchical level is denoted as P^n. Below, a detailed formulation of the extended pathfinding problem is presented.

$$\min \sum_{n=1}^{N} \sum_{e \in E^n} w_e^n x_e^n \tag{6}$$

$$\begin{aligned} \text{s.t.} \sum_{e \in \delta^+(v)} x_e^n - \sum_{e \in \delta^-(v)} x_e^n = 0, \\ \forall v \in V^n \setminus \{s^n, t^n\}, \forall n \in \{1, \ldots, N\} \end{aligned} \tag{7}$$

$$\sum_{e \in \delta^+(s^n)} x_e^n = 1, \quad \forall n \in \{1, \ldots, N\} \tag{8}$$

$$\sum_{e \in \delta^-(t^n)} x_e^n = 1, \quad \forall n \in \{1, \ldots, N\} \tag{9}$$

$$x_e^n \in \{0, 1\} \quad \forall e \in E^n, \quad \forall n \in \{1, \ldots, N\} \tag{10}$$

Here, N represents the maximum hierarchical level, w_e^n is the weight of link e at hierarchical level n, and x_e^n is a variable that takes the value of $x_e^n = 1$ if link e is included in the path P^n, and $x_e^n = 0$ otherwise. Additionally, s^n and t^n denote the start and target nodes, respectively, at hierarchical level n. Equations (7) to (9) represent the application of the general pathfinding constraints at each hierarchical level.

C. Hierarchization of Road Network

1) Extraction of Upper-Level Networks: The road network addressed in this paper is implemented as an undirected graph, which serves to reduce the complexity of the network and enhance computational efficiency. As an example of driver preference, avoidance of narrow roads is adopted, and the upper-level networks are extracted based on the number of lanes, which is closely related to this preference. This driver preference was chosen based on the needs of delivery drivers in our kerosene delivery planning research, who want to take routes that avoid narrow roads not well-cleared of snow.

For a link e, let l_e be the smaller number of lanes on one side of the road. However, if link e is a one-way link with zero lanes on one side, then $l_e = 1$. When n is the hierarchical level of the network, define the set of links satisfying $l_e \geq n$ as E^n, and the set of nodes at both ends of each link in E^n as V^n. The subnetwork composed of V^n and E^n is denoted as $G^n = (V^n, E^n)$, and G^n is the network at level n. The original road network is G^1, and the upper-level networks $G^n (2 \geq n \geq N)$ are extracted sequentially from G^1 $(V^1 \supseteq V^2 \supseteq \cdots \supseteq V^N, E^1 \supseteq E^2 \supseteq \cdots \supseteq E^N)$. It is noted that N is a parameter that sets the highest hierarchical level.

2) Extraction of Representative Node Set: In the pathfinding method using a hierarchical road network in this paper, the search is started from the original road network and progressively transitions to the upper-level networks. With each transition to an upper-level network, the route candidates are narrowed down. Since the route candidates heavily depend on the nodes included in the upper-level networks, if many nodes of low importance, which are rarely used in routes between any Origin-Destination pair (commonly referred to as OD pair), are included in the upper-level networks, routes that pass through these less important nodes may be output, potentially leading to a significant deterioration in route cost. Therefore, adding nodes of high importance to the upper-level networks is crucial for ensuring the accuracy of approximate solutions in pathfinding using the hierarchical road network.

Centrality measures are useful in measuring the importance of nodes. These measures evaluate how central elements are within a network, with common types including degree centrality, closeness centrality, and betweenness centrality [11]. Especially, betweenness centrality, which evaluates each node based on the frequency of its use in the shortest paths between any OD pair, is valuable as a measure of node importance when constructing hierarchical road networks.

While betweenness centrality typically evaluates each node individually, it is important to consider the interdependencies and cooperative relationships among nodes when taking betweenness centrality into account, as nodes are mutually dependent on each other.

Therefore, Fushimi et al. [12] have proposed what is known as set betweenness centrality, which considers the centrality of a set of nodes.

Betweenness centrality and set betweenness centrality are calculated by (11) and (12), respectively.

$$bwc(v) = \sum_{s \in V} \sum_{t \in V} \frac{\sigma_{s,t}(v)}{\sigma_{s,t}} \tag{11}$$

$$SB(R) = \sum_{s \in V} \sum_{t \in V} \frac{\sigma_{s,t}(R)}{\sigma_{s,t}} \tag{12}$$

Here, $\sigma_{s,t}$ represents the number of shortest paths between nodes s and t, $\sigma_{s,t}(v)$ is the number of these paths passing through node v, and $\sigma_{s,t}(R)$ denotes the number of shortest paths from node s to node t that pass through $\forall r \in R$. The set of nodes R that maximizes (12) is considered the representative node set.

It should be noted that calculating the exact betweenness centrality requires computing the shortest paths for every OD pair in the target road network. However, this computation requires $O(V^3)$ time complexity and is challenging for large-scale networks. Therefore, a method has been proposed that approximates betweenness centrality by randomly sampling nodes from the network, performing single-source shortest pathfinding from these sampled nodes, and using the obtained paths information for the approximation [13]. This method has been shown to have good approximation accuracy in the paper by Wandelt et al. [14], and it is also adopted in this paper.

While betweenness centrality can be calculated by counting the number of times each node in the network based on the obtained paths information, for set betweenness centrality, it is necessary to consider which combination of nodes maximizes the objective function in (12). Therefore, in this paper, the greedy algorithm proposed by Fushimi et al. is used to find the set R that maximizes (12). The number of elements in the representative node set R, denoted as $|R|$, is adjusted by the parameter K. For specific procedures, refer to the paper by Fushimi et al. [12].

3) Correction of Upper-Level Networks: The correction is applied to the upper-level networks from G^2 to G^N. This correction process mainly consists of two stages: resolving disconnections and adding the Representative node set R identified in Section III-C2.

When extracting the upper-level networks based on the attributes of links, these networks may be disconnected. In such cases, an operation to resolve this by adding nodes and links is necessary. Therefore, our proposed method first resolves the disconnection in the upper-level networks. However, merely resolving the disconnection in the upper-level networks might lead to an excessive simplification, potentially resulting in the output of routes from pathfinding using the hierarchical road network with significantly worse costs compared to the minimum cost routes. To improve the accuracy of the approximate solutions in pathfinding using the hierarchical road network, the representative node set R is added.

4) Pathfinding Using Hierarchical Network: In this paper, the pathfinding method proposed by the previous method [4] is adopted. This method's overview involves bi-directional searching from both the start and target nodes, and the transition to an upper-level network occurs only when searches from both directions reach nodes that exist in the network of the next upper hierarchical level. It should be noted that, the transition to an upper-level network is restricted only to cases that meet the above condition, preventing transitions from upper-level to lower-level networks. While any pathfinding algorithm can be used, this paper employs the A* algorithm.

The details are described below. The input consists of a hierarchical network $\{G^1, G^2, ..., G^N\}$, a start node s, and a target node t, with the output being the path from s to t. In the the following procedure, 'advancing the search by one step' refers to extracting the node with the minimum cost from the queue, calculating and updating the costs for all nodes adjacent to the extracted node in G^n, and then adding the updated nodes

Algorithm 1 Correction of Hierarchical Network

Require: Hierarchical network $\{G^1, G^2, \ldots, G^N\}$, Representative node set R
Ensure: Corrected hierarchical network
for $n = 2$ to N **do**
  if G^n is disconnected **then**
    Select node with highest betweenness centrality in each subnetwork
    Form pairs of selected nodes
    for each pair **do**
      Search for a minimum cost path on G^{n-1}
      Add nodes and links from the path to G^n
    end for
  end if
  Select node v^* with highest betweenness centrality in G^n
  Add all nodes from R to G^n
  Form pairs of each $v \in R \cup \{v^*\}$
  for each pair **do**
    Search for a minimum cost path on G^{n-1}
    Add nodes and links from the path to G^n
  end for
end for

Figure 1. Algorithm for correction of hierarchical network.

to the queue.

Step 1: Start the search with the level of the network to be searched set to $n = 1$.
Step 2: Place the start node s in the forward search queue, and the target node t in the backward search queue.
Step 3: If the forward search has not yet reached a node contained in G^{n+1}, advance the forward search by one step.
Step 4: If the backward search has not yet reached a node contained in G^{n+1}, advance the backward search by one step.
Step 5: If there are nodes already searched from both directions, conclude the search and determine the route.
Step 6: If searches from both directions have reached nodes contained in G^{n+1}, use these nodes as new start and target nodes, clear the queues, update $n \leftarrow n+1$, and return to Step 2. Otherwise, maintain n as it is and return to Step 3.

IV. Experiment

Hierarchical road networks for Japan's heavy snowfall areas are constructed using three different methods, each with a partially different process after the extraction of the upper-level networks. the construction times of the hierarchical road networks and the sizes of the networks at each hierarchical level are compared. Pathfinding using the hierarchical networks of each method in the road network is also performed for OD pairs. the sum of the construction time of the hierarchical networks and the total computation time for individual pathfinding are then evaluated, as well as the increase in cost relative to the minimum cost paths.

A. Experimental Setup

A 14 km square area centered on Sapporo Station was extracted from OpenStreetMap [15] as an area where kerosene delivery is routinely conducted due to heavy snowfall in winter, and was used as the road network for the experiment (Figure 2). Note that approximately 85% of this road network consists of links with only one lane. Therefore, hierarchization based solely on the number of lanes might lead to excessively simplified upper-level networks, and some ingenuity is required to correct the upper-level networks. Lane information and link lengths are acquired from the OpenStreetMap road network for use in the hierarchization of the network and the calculation of route costs.

In this paper, three different hierarchization methods are evaluated, each differing in how disconnections are resolved, and a representative node set is added to the upper-level networks, which are extracted based on the number of lanes. Linkage method 1 resolves the disconnections using the previous method and does not add a representative node set. Linkage method 2 first resolves the disconnections using the previous method and then adds the representative node set using the proposed method. Linkage method 3, which is the method proposed in this paper, uses the proposed method to both resolve the disconnections and add the representative node set.

The parameter settings for each method are shown in Table II. Within Table II, H_{max} is a threshold value that controls the extent to which disconnected subnetworks are linked during the correction process. For linkage methods 1 and 2, $H_{max} = 100$ was chosen because it yielded the highest accuracy of approximate solutions in preliminary experiments. K represents the number of elements in the representative node set to be added to the upper-level networks. The upper limit of hierarchical levels for each method was set to $N = 3$.

When adding the representative node set in linkage methods 2 and 3, an approximation of betweenness centrality is necessary. The number of nodes randomly sampled from the network for the approximation of betweenness centrality was set to 1,000. This number can vary depending on the network used and its scale and is not strictly required to be this value.

Pathfinding is performed using the hierarchical networks constructed by each method, and each method is evaluated based on the results. Additionally, as an exact solution method, pathfinding without using a hierarchical network is also performed. When performing pathfinding using a hierarchical network, the method described in Section III-C4 is adopted, and when performing pathfinding without hierarchical networks, the unidirectional search A* algorithm is adopted.

OD pairs are classified based on Euclidean distance and divided into seven intervals at every 2.5 km. From each interval, 5,000 pairs are randomly selected, and pathfinding is performed for each OD pair, with the results compared.

TABLE I
VALUES OF $w_{e,i}$ FOR THE LINK COSTS ACCORDING TO EACH DRIVER PREFERENCE TYPE FOR NARROW ROADS

Weight	**Number of lanes**		
	$l_e = 1$	$l_e = 2$	$l_e \geq 3$
$w_{e,1}$	10	5	1
$w_{e,2}$	4	2.5	1
$w_{e,3}$	2.768	1.607	1
$w_{e,4}$	1	1	1

TABLE II
PARAMETER SETTINGS FOR EACH LINKAGE METHOD

Algorithm	**Resolution of disconnection**	**Addition of representative node set**	H_{max}	K
Linkage method1	previous method	-	100	-
Linkage method2	previous method	proposed method		10
				20
				:
				100
Linkage method3	proposed method	proposed method	-	10
				20
				:
				100

Next, multiple types of driver preferences are prepared, each with different degrees of avoidance of narrow roads. As the degree of avoidance of narrow roads increases, the cost of such roads also increases, which may change the total cost of the candidate routes and thus the final route selected. In this experiment, four types of driver preferences A_1, A_2, A_3, A_4 are prepared in order of increasing the degree of avoidance of narrow roads. For each type of driver preference $A_i (i = 1, 2, 3, 4)$, the link cost $\tilde{c_{e,i}}$, considering the number of lanes l_e of link e, is given by (13). Here, c_e is the original link cost, which in this paper is given as the Euclidean distance between the endpoints of the link. $w_{e,i}$ is the weight applied to the link cost according to the driver's avoidance of narrow roads. The values of link cost used in this experiment for A_1, A_2, A_3 are adopted from those calculated in [4]. The weight $w_{e,3}$ was set based on actual probe data in [4], and $w_{e,1}$ and $w_{e,2}$ were values inferred by the authors of [4] intended to represent varying degrees of avoidance to narrow roads. Additionally, this experiment incorporates a driver preference type A_4, representing drivers with no avoidance of narrow roads. This driver type was added to evaluate the performance of the proposed method with and without avoidance to narrow roads.

$$\tilde{c}_{e,i} = w_{e,i} c_e \tag{13}$$

This experiment was conducted in a computing environment equipped with an AMD EPYC 7402 24-Core CPU and 128GB of memory.

B. Size and Construction Time of Hierarchical Network

The sizes and construction times of the hierarchical networks constructed by each linkage method are shown in Table III. For each method, $n = 1$ is the original road network, and for $n \geq 2$, networks of different sizes are constructed depending on the method and parameters.

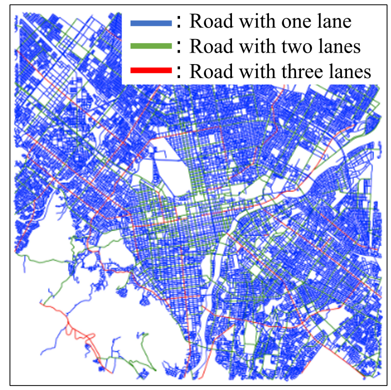

Figure 2. Road network around Sapporo Station.

As shown in Table III, the network at hierarchical level 2 in linkage method 3, even with $K = 100$, has approximately 43% fewer nodes and about 52% fewer links compared to the network at hierarchical level 2 in linkage method 1. On the other hand, no significant differences were observed in the number of nodes and links in the hierarchical networks for levels $n = 2, 3$ between linkage method 1 and linkage method 2.

While the construction of the hierarchical network using linkage method 1 takes about 1 minute, the construction time using linkage methods 2 and 3 increases by approximately 600 times for $K = 100$. However, the hierarchical networks constructed in this paper are intended for application in the vehicle routing problem where hundreds of thousands of pathfinding tasks occur, and where delivery routes are repeatedly adjusted and planned in specific regions. Therefore, once a hierarchical network is constructed, it can be reused as long as there are no changes to its topology or target area. Considering this, the construction time for the hierarchical networks in linkage methods 2 and 3 is acceptable.

C. Results of Pathfinding Using Hierarchical Network

The results of pathfinding using the hierarchical networks constructed by each method were compared with the results of pathfinding using the unidirectional search A* algorithm, which does not use hierarchical networks. Table IV shows the average costs and computation times for each section for the driver preference type A_1, which has the highest degree of avoidance of narrow roads among four driver preference types. Note that only a portion of the parameter K results for linkage methods 2 and 3 are presented. In Table IV, T_{ave} and C_{ave} represent the average computation time and cost of the routes for each section, respectively, with bold figures indicating the lowest average costs among the routes using hierarchical networks in each section. Uni-A* refers to the unidirectional search A* algorithm.

The results in Table IV show that compared to the unidirectional search A* algorithm, the calculation time of individual

TABLE III
SIZE AND ELAPSED TIME OF HIERARCHICAL NETWORKS CONSTRUCTED BY EACH METHOD

Algorithm	H_{max}	K	n	number of nodes	number of links	elapsed time
-	-	-	1	31,139	49,967	-
Linkage method1	100	-	2	12,691	15,765	57s
			3	5,267	5,790	
Linkage method2	100	10	2	12,691	15,765	6h28m21s
			3	5,315	5,849	
		100	2	12,946	16,065	9h40m33s
			3	6,648	7,319	
Linkage method3	-	10	2	7,219	7,618	6h26m51s
			3	2,798	2,865	
		100	2	7,724	8,211	9h38m7s
			3	5,076	5,321	

pathfinding is reduced to 4-6% with linkage method 1 or 2, and to 3-6% with linkage method 3, confirming that the hierarchization of road networks contributes significantly to reducing the calculation time of pathfinding.

Regarding the average cost of routes for each section, the case of $K = 100$ in linkage method 3 had the minimum average route cost in all sections among the three linkage methods. The worsening of route costs compared to the average cost of routes from the unidirectional search A* algorithm was about 4-9%. This characteristic was consistent for other driver preference types as well, suggesting that linkage method 3 could be an effective approach for constructing hierarchical networks that consider the different preferences for each driver.

D. Discussion

The construction time of the hierarchical road network in linkage method 3 was approximately 9 hours and 40 minutes, about 600 times longer than that of linkage method 1. The processes involved in constructing the hierarchical network in linkage method 3 are: (1) extraction of the upper-level networks, (2) sampling of nodes and single-source shortest pathfinding starting from the sampled nodes, (3) approximation of set betweenness centrality and extraction of the representative node set, and (4) correction of the upper-level networks. The most time-consuming process among these is process (3), which takes approximately 8 hours and 40 minutes, or 90% of the construction time. This is because the paths obtained in process (2) are used as references to calculate the representative node set. However, as the number of paths referenced increases, the computation time of process (3) also increases. Therefore, the computation time of process (3) depends on the number of paths obtained from process (2). Single-source shortest pathfinding was performed, starting from sampled nodes (for this paper, 1,000 nodes were randomly sampled), with paths being computed to all other nodes in the network. However, it may not be necessary to compute paths to all other nodes, and reducing the number of paths to be calculated could potentially reduce the construction time of the hierarchical network.

Figure 3 shows the cost ratio compared to the minimum cost route for the driver preference type A_1. Note that Figure 3 is a histogram of the cost ratios for all OD pairs. From Figure 3, it can be observed that using linkage method 3 allows for the calculation of routes with cost ratios closer to 1.0 for more OD pairs compared to linkage methods 1 and 2. However, the number of OD pairs with a cost ratio exceeding 2.0 was 47 for linkage method 1, 15 for linkage method 2, and 30 for linkage method 3, indicating that even with linkage method 3, some OD pairs experienced routes with significantly worse costs compared to their respective minimum cost routes. The worsening of route costs occurred in cases where the OD pairs were closely located. In response to this issue, hierarchical networks are currently used regardless of the OD pair distance, but we will consider improvements, such as deciding whether to use hierarchical networks based on the distance of the OD pairs.

Moreover, there is potential for improvement in the method of extracting the representative node set. In the proposed method, the set betweenness centrality is calculated from the entire target network, and the representative node set is extracted based on this calculation. However, road networks have regional characteristics, and the nodes that are frequently traversed should differ depending on the movement between and within regions. The proposed method extracts the representative node set without taking this into account, and thus may miss frequently traversed nodes, especially within a specific region. We will consider an approach that divides the road network into multiple regions, calculates the set betweenness centrality for each movement between and within regions, and extracts the representative node set.

V. CONCLUSION AND FUTURE WORK

In this paper, we proposed a road network hierarchization method aimed at optimizing vehicle routing problems. Our method involves constructing a hierarchical road network that incorporates the concept of set betweenness centrality, an extension of betweenness centrality, which is one of the centrality measures. Both the previous method and the proposed method were applied for pathfinding on the road network around Sapporo Station, and for various types of driver preference with different degrees of avoidance of narrow roads. Compared to performing pathfinding without hierarchical networks, the calculation time of individual pathfinding was reduced to 4-6% with the previous method and to 3-6% with our method. Additionally, by using our method, the average cost of routes improved in all sections compared to the previous method,

TABLE IV
RESULTS OF PATHFINDING BY EACH METHOD (DRIVER PREFERENCE TYPE A_1)

Algorithm	H_{max}	K	Section[km]													
			0-2.5		2.5-5.0		5.0-7.5		7.5-10.0		10.0-12.5		12.5-15.0		15.0-17.5	
			T_{ave} [sec]	C_{ave} [km]	T_{ave} [sec]	C_{ave} [km]	T_{ave} [sec]	C_{ave} [km]	T_{ave} [sec]	C_{ave} [km]	T_{ave} [sec]	C_{ave} [km]	T_{ave} [sec]	C_{ave} [km]	T_{ave} [sec]	C_{ave} [km]
Uni-A*	-	-	0.50	11.66	1.42	16.92	2.74	22.16	4.15	27.34	5.76	32.75	6.95	37.33	7.62	39.18
Linkage method1	100	-	0.03	13.00	0.07	18.53	0.13	23.96	0.21	29.43	0.30	35.13	0.38	40.11	0.48	42.15
Linkage method2	100	20	0.03	12.97	0.07	18.49	0.13	23.91	0.21	29.38	0.31	35.11	0.40	40.10	0.51	42.14
		60	0.03	12.95	0.07	18.38	0.14	23.72	0.22	28.98	0.33	34.71	0.42	39.76	0.54	42.15
		100	0.03	12.83	0.07	18.22	0.14	23.59	0.23	28.81	0.35	34.51	0.45	39.68	0.56	41.83
Linkage method3	-	20	0.03	12.88	0.05	18.21	0.09	23.67	0.13	29.14	0.18	34.69	0.22	39.34	0.28	41.49
		60	0.02	12.82	0.05	18.12	0.09	23.42	0.14	28.71	0.20	34.28	0.25	39.06	0.30	41.32
		100	0.02	**12.67**	0.05	**18.03**	0.09	**23.31**	0.15	**28.54**	0.21	**34.13**	0.27	**39.05**	0.32	**40.76**

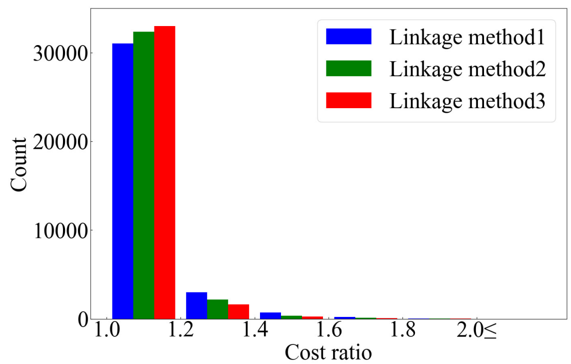

Figure 3. Cost ratio to the minimum cost paths (driver preference type A_1).

and even when compared to the minimum cost paths, the worsening of route costs was about 4-9%. From the above, our method demonstrated superior performance in terms of pathfinding computation time and the cost of the routes obtained, surpassing the previous method.

REFERENCES

[1] E. W. Dijkstra, “A note on two problems in connexion with graphs,” *Numerische Mathematik*, vol. 269, p. 271, 1959.

[2] G. R. Jagadeesh, T. Srikanthan, and K. Quek, “Heuristic techniques for accelerating hierarchical routing on road networks,” *IEEE Transactions on intelligent transportation systems*, vol. 3, no. 4, pp. 301–309, 2002.

[3] H. Ooe, S. Yokoyama, T. Yamashita, H. Kawamura, and M. Tada, “Optimization of kerosene delivery planning using tabu search,” no. 15, 2022, pp. 1–8.

[4] S. Fukuda, K. Abe, H. Fuji, T. Yamada, and S. Yoshimura, “Layered route search method for large-scale multi-agent-based traffic simulation,” *IPSJ Journal*, vol. 59, no. 7, pp. 1435–1444, 2018.

[5] T. Nakamura, T. Yoshii, and R. Kitamura, “Development of a simplified network which represents all roadway links,” *Infrastructure planning review*, vol. 23, pp. 441–446, 2006.

[6] Q. Song and X. Wang, “Efficient routing on large road networks using hierarchical communities,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 1, pp. 132–140, 2011.

[7] M. M. Nejad, L. Mashayekhy, and R. B. Chinnam, “Effects of traffic network dynamics on hierarchical community-based representations of large road networks,” in *2012 15th International IEEE Conference on Intelligent Transportation Systems*, 2012, pp. 1900–1905.

[8] D. N. A. Mensah, H. Gao, and L. W. Yang, “Approximation algorithm for shortest path in large social networks,” *Algorithms*, vol. 13, no. 2, 2020.

[9] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, “Fast unfolding of communities in large networks,” *Journal of Statistical Mechanics: Theory and Experiment*, vol. 10, p. P10008, 2008.

[10] M. Shimizu, S. Yokoyama, T. Yamashita, and H. Kawamura, “A proposal of subnetwork linkage method based on betweenness centrality in road network hierarchization,” vol. 2023-ITS-95, no. 18, 2023, pp. 1–8.

[11] L. C. Freeman, “A set of measures of centrality based on betweenness,” *Sociometry*, vol. 40, no. 1, pp. 35–41, 1977.

[12] T. Fushimi, K. Saito, T. Ikeda, and K. Kazuhiro, “Proposing set betweenness centrality measures focusing on nodes’ collaboratative behaviors and its application,” *The IEICE transactions on information and systems*, vol. J96-D, no. 5, pp. 1158–1165, May 2013.

[13] U. Brandes and C. Pich, “Centrality estimation in large networks,” *International Journal of Bifurcation and Chaos*, vol. 17, no. 7, pp. 2303–2318, 2007.

[14] W. Sebastian, S. Xing, and S. Xiaoqian, “Scalability of betweenness approximation algorithms: An experimental review,” *IEEE Access*, vol. 7, pp. 104 057–104 071, 2019.

[15] OpenStreetMap contributors, “Openstreetmap,” Available: https://www.openstreetmap.org, 2004, [retrieved: January, 2024].