Multiple Destination Multiple Route Recommendation for Evacuation in Emergency Situation

Chayanon Sub-r-pa[1] and Goutam Chakraborty[2]
Department of Software and Information technology, Iwate Prefectural University
Sugo, Takisawa, Iwate 020-0693, Japan
E-mail : [1]chayanon.s@gmail.com, [2]goutam@iwate-pu.ac.jp

Abstract: In road network multiple destinations route searching is common when users are searching for service instead of destination. Navigation system will present list of destination, then user will choose the specific destination. For more constrained situation such as request for parking-lot near to festival ground or request for safety shelter for disaster situation. Navigation system will show list of destinations in the area, then user will choose specific destination by themselves. In general, user will choose the closest destination and shortest route. If there are many users searching the same service simultaneously, the closest destination and the shortest route will get traffic congestion quickly. Vehicles will face traffic congestion when follow the route, and may not get service at destination, because each destination has limited resource. It would be better to choose different route or different destination from beginning of travel. In this paper, we proposed road network routing algorithm considering multiple destinations, multiple routes, and resources at destinations are limited.
Keywords—**Route Recommendation,Road network**

1. Introduction

When an user is searching for a service instead of specific destination, a specific destination cannot defined. Navigation system will show all service points in that area. User will take responsibility to choose destination and select route by themselves. One such situation is, when a large number of people are searching for parking-lot near festival area. Navigation system will show all parking-lot. User chooses specific parking-lot by themselves. In general, user will choose parking-lot that is closest to festival. Traffic congestion will appear quickly in this route. With limited space in each parking-lot, available space may run out when user arrive and a user needs to search a new destination and new route again.

Emergency situation such as Tsunami evacuation. Most of the populations from a wide are needed to evacuate out of danger area. Fig. 1 is the map from tsunami disaster on March 11th, 2011 in Miyako [1], Iwate, Japan. Red zone is area that got high level of tsunami water and purple line is evacuate route. Vehicle in risk or damaged area need to escape to high altitude area. Most of user will choose the closest destination. With large number of people choosing the same destination, traffic congestion to that destination will appear in very short time. When vehicles are held in traffic congestion, it is difficult to take a U-turn, or change route. If a vehicle cannot move, or stop for long time in risky area, they may get affected by tsunami. It is better to choose a different destination, like using the highway to escape from the risk area.

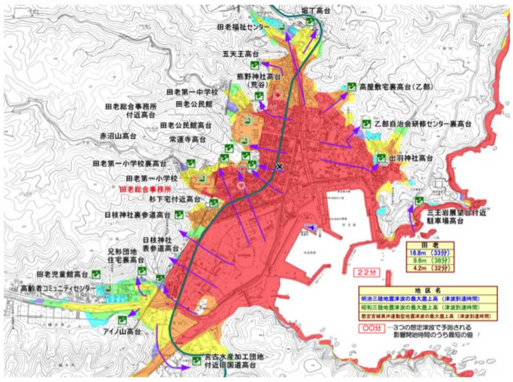

Figure 1. Hazard map from tsunami disaster on March 11th, 2011 in Miyako

Moreover some safety shelter, that is easy to access, may already be not suitable for shelter at all, because they are affected by natural calamities, or already too crowded to accept more people. If people who attempt to evacuate to safety shelter, using navigation system that can use information of destinations and road traffic condition while traveling, they can get navigated for optimal route and destination, and can access safe shelter in reasonable time.

There is no support for multiple destinations problem or destination specified by its property only (like a parking place or a safe shelter). In this paper, we propose road network routing algorithm based on Yen's k-shortest path algorithm, to support multiple destination problem with limited resource at service points. The algorithm reduces traffic congestion by navigating vehicles to different good routes.

The rest of this paper is organized as follows. Section 2. discusses related works. Section 3. describes Multi-Destination Multi-Route (MDMR) navigation system. Section 4. presents the simulation environment, experimental results and their analysis. The paper is concluded in section 5.

2. Related Works

Road network navigation system are based on shortest-path algorithm such as dijkstra algorithm [2], bellman-ford algorithm [3], or $A*$ algorithm [4]. The single source shortest path algorithm can be used to find the shortest path from a source vertex to all the other vertices in a graph. The single destination algorithm is used to find the shortest path in a graph, from all vertices to a single destination. It can be also be seen as a single source problem by reversing the arcs.

For more intelligent navigation system, or road traffic management, Dynamic Traffic Assignment (DTA) [5][6] is proposed, which leads to either system-optimal or user-optimal route assignments. DTA was designed to help explain the basic concepts and definitions of DTA models and addresses application, selection, planning, and execution of a DTA model. It also describes the general DTA modeling procedure and modeling issues that may be of concern to the model user. J. Pan et al [7] improved DTA by extending DTA algorithm for traffic optimization, while avoiding the issues that make DTA impractical, such as the lack of scalability, robustness, and high computation time.

Existing road network routing or path planning algorithm could navigate a vehicle to a destination with different criteria. But none of the existing proposals focus on disaster environment (like Tsunami or land slide), sudden increase in traffic (like festival), or multiple destination problems. In this paper, we propose routing algorithm that supports multiple destinations and multiple routes, to avoid congestion at service points as well as road network.

3. Multi-Destination Multi-Route (MDMR) Recommendation

In our problem, the cost metric is time of travel, as well as the state of service at the destination. Road network traffic information such as travel time can be obtained by loop detector, road network sensor or from mobile phone network. The idea to minimize traffic congestion is to distribute vehicles over multiple good routes. To distribute the burden at different service points, we navigate vehicles to different destinations over different routes to avoid overloading of one. In this section we introduce method to calculate k-shortest routes considering multiple destinations simultaneously. We introduce distribution method be sending vehicles on different good routes depending on travel time and destination service available status, to prevent traffic congestion and reduce number of vehicles arriving at crowded destinations.

3.1 Virtual Destination

Shortest path algorithms such as dijkstra algorithm can calculate shortest path from source node to all nodes in the network. It is not suitable to calculate shortest path to all nodes because there are huge number of nodes and edges in road network. If navigation system needs to calculate k-shortest paths to specific destination, route calculation time will increase by k. To calculate k-shortest routes to multiple destinations, route calculation time will increase by number of destinations and k.

We proposed routing algorithm based on Yen's k-shortest path algorithm [8]. Support for multiple destinations was achieved by introducing a new destination node, named *Virtual Destination* (VD). We add VD node to network, and create link connecting between destinations to VD, call virtual links. Instead of running shortest path or k-shortest path algorithm to each destination, we calculate route from source to VD, the result contains shortest route to specific destination. Real destination will be included in the route, as a node before VD. In the beginning we set virtual link cost as 0. To consider limited resource at destination we design system to increase virtual link cost by available resource at destination. Detail is explained in the next section.

We use Yen's k-shortest path algorithm [8] to find a set of k paths: the 1st.-best route, the 2nd.-best route,... the kth.-best route, the result is a set of k-shortest routes.

In each route is contain only one specific destination, though more than one route may contain the same destination. This is possible that k-shortest routes may not include all real destinations, because some destination is too far from starting point, or some link converging to some destination is too congested.

With k-shortest path algorithm, this is possible that ith.-best route may contain more than one real destination. This is not practical to suggest a vehicle to travel through one real destination to a different destination. To avoid this situation, we ignore ith.-best route that contain more than one real destination, and replace it with next.-best route.

In the simulation experiments, described in the next section, our algorithm finds k shortest routes and update road traffic information at regular intervals. Updating link status at regular interval would avoid recently congested routes automatically. Vehicles may change route when new route recommendation is received.

3.2 Road Traffic Distribution

In general, when navigation system recommend shortest route, and optional route. User will choose shortest route. In case of high population in same area request same service simultaneously, shortest route will become congested quickly. To prevent traffic congestion in this environment by using traffic distribution over a number of near optimal routes, i.e., 1st.-best or 2nd.-best routes etc. To distribute traffic in MDMR, the system will recommend vehicle to follow routes over different destinations, and/or over different routes to the same destination.

Our proposed routing algorithm of finding k shortest routes achieve this goal efficiently due to the introduction of virtual destination node. The proposed system will distribute vehicles in k routes depending on their respective costs, i.e., travel time. The probability that a particular route will get selected depends on the calculated travel time for that route. Probability is inversely proportional to the travel time. Out of k shortest routes, the ith.-best route will be selected with probability P_i, expressed in Eq.(1), where C_i is cost of ith.-best route.

$$P_i = \left(1 - \frac{C_i}{(\sum_{j=1}^{k} C_j)}\right) \times \frac{1}{k-1} \quad (1)$$

3.3 Destination resource limited

To include resource constraint at destination, the idea is to decrease traffic to a destination where resource has plummeted. We decrease distribution ratio of that destination by increasing virtual link cost. As we mention at beginning, we initially set virtual link cost equal to 0. When the resource at a destination is completely used up, the virtual link cost would

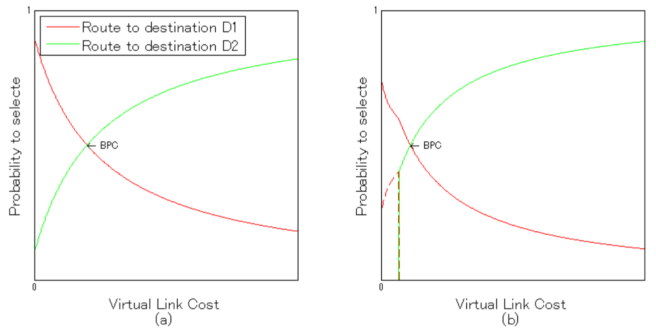

Figure 2. Virtual link cost vs. probability to selected

be ∞. Thus, it is difficult to assign the value of the virtual link cost, because it will vary from 0 to ∞. From Fig 2(a), we plot all probability of distribution P_i when $k = 2$, and each route has different destination. We start virtual link cost of closest destination to VD as 0, and increase that virtual link cost to ∞. We found crossing point when virtual link cost increase, denote this virtual link cost value as Balance Probability Cost (BPC). This point is determines where distribution probabilities are same. Fig 2(b), is when at the beginning all routes lead to the same destination. The plot shown that probability of distribution will be equal with different destination.

We formulate virtual link cost function to control priority of each destination as follows. N is the number of nodes and μ is set of nodes, $\mu = \{\mu_1, \mu_2, \ldots, \mu_N\}$ and $|\mu| = N$. D is the number of destinations and δ is the set of destinations, $\delta = \{\delta_1, \delta_2, \ldots, \delta_D\}$ and $|\delta| = D$. Eq. (2) is present matrix Ç when ç_{ij} is shortest path cost for node μ_i to destination δ_j. Eq.(3) is 1st-best shortest path cost for node μ_i to destination δ_j where $j = 1$ to D. Eq. (4) is present BPC for node μ_i to destination δ_j where $j = 1$ to D.

We design to increase virtual link cost between node δ_j to $\text{\H{V}}$ in range of [0, $\beta \times BPC(\mu_i, \delta_j)$] where β is static number. With this virtual link cost function, when some destination low resource available it will not be included in k-shortest routes, or it will included with high cost (low probability to selected).

$$\text{Ç} = \begin{array}{c} \\ \mu_1 \\ \mu_2 \\ \cdots \\ \mu_N \end{array} \begin{array}{c} \begin{array}{cccc} \delta_1 & \delta_2 & \cdots & \delta_D \end{array} \\ \begin{bmatrix} \text{ç}_{11} & \text{ç}_{12} & \cdots & \text{ç}_{1D} \\ \text{ç}_{21} & \text{ç}_{22} & \cdots & \text{ç}_{2D} \\ \cdots & \cdots & \cdots & \cdots \\ \text{ç}_{N1} & \text{ç}_{N2} & \cdots & \text{ç}_{ND} \end{bmatrix} \end{array} \quad (2)$$

$$\hat{\text{Ç}}(\mu_i) = max(\{\text{ç}_{i1}, \text{ç}_{i2}, \ldots, \text{ç}_{iD}\}) \quad (3)$$

$$BPC(\mu_i, \delta_j) = \hat{\text{Ç}}(\mu_i) - \text{ç}_{ij} \quad (4)$$

$$VLC(\mu_i, \delta_j) = BPC(\mu_i, \delta_j) \times \beta \times \frac{\delta_j.used}{\delta_j.limit} \quad (5)$$

4. Simulation, Result and Analysis

We simulated the proposed routing algorithm, and set environment to similar tsunami evacuation situation. All vehicles are needed to be evacuated to safety place. We used Simulation of Urban Mobility (SUMO) 0.24 [9] to simulate vehicle mobility. SUMO is an open-source highly portable microscopic road traffic simulation package designed to handle large road network.

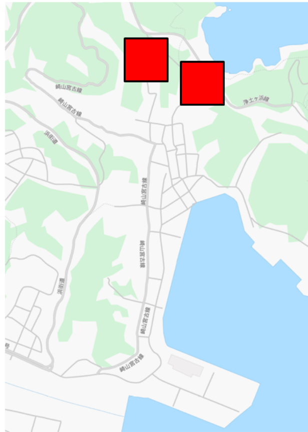

Figure 3. Miyako area

We used a part of map data from tsunami disaster area recorded on March 11th, 2011 in Miyako, Iwate. The road network is shown in Fig 3. This road map data is downloaded from OpenStreetMap website in OSM format [10][11]. And used SUMO to generate random trips for each vehicle in this simulation. Speed of all vehicles are limited to 15-20km/hr. This speed is adequate to evacuate from tsunami [7]. We assume to have 2 destinations marked as red rectangle, and available resources for each destination is 50% of the number of vehicles in network. And road traffic information and route calculation is updated every 5 minutes.

In each scenario, we generate vehicles joining the road network start from 1000 vehicles for low density network and increase to 2000 vehicles for high density network. Initially all vehicles travel to randomly selected destination. After 10 minute, when tsunami warning starts, all vehicles will head to proper destination using proposed MDMR route recommendation. After vehicles received alarm, the system will use algorithm to find proper route and destination for each starting node. Vehicles will receive route recommendation message and use it as route to destination.

All simulations used proposed virtual destination. We compare results using k=1 where only 1st-best route is recommended, and using k=2 where 1st-best and 2nd-best are recommended and traffic is distributed depending on cost to each destination as in Eq. (1). When using k=2, we compare results using $0 \leq \beta \leq 2$ in Eq.(5) to change virtual link cost and distribution ratio to each destination when resources are depleted.

4.1 Number of Vehicle take U-turn at service point

Number of vehicles take U-turn at service point is the number of vehicles, which follow recommended route to destination but can not get service. They need to take a U-turn and follow a new route to a new destination. This situation can happen

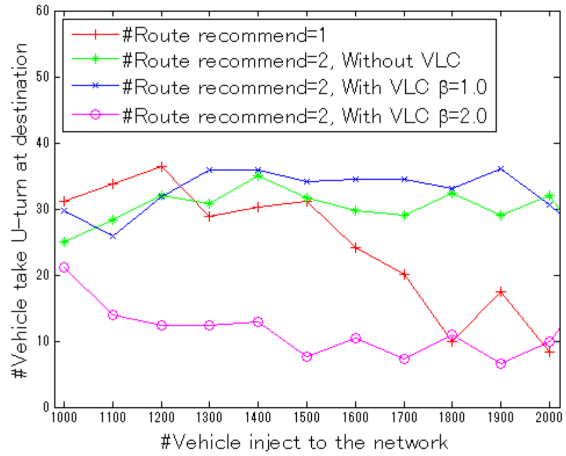

Figure 4. Number of vehicle take U-turn at unserviceable destination

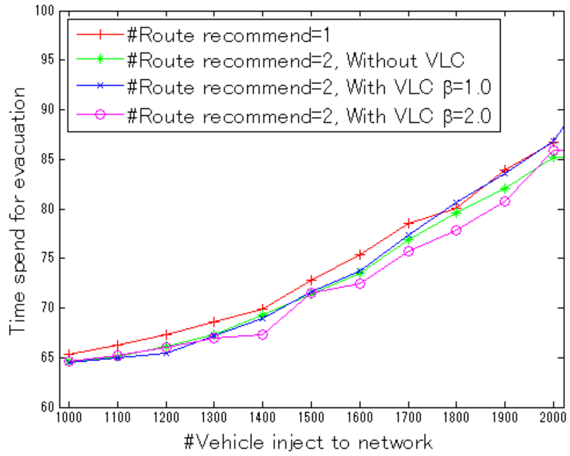

Figure 5. Evacuation time spend

after a while following the start of the evacuation. The easy to access destination will run out of resources quickly. From Fig 4, using k=1 the number of vehicles taking U-turn is more in the beginning. When the number of vehicle on the road network is increased they get better result than using VLC with $\beta = 1$. Using VLC with $\beta = 2$ is shows the lowest number of vehicles take U-turn for all traffic scenario.

4.2 Evacuation Time

Evacuation time is the time required from starting of evacuation warning until all vehicles are in safety shelter. Fig 5 shows the result of evacuation time required. From the plot, evacuation time is increased as the number of vehicles injected to the network is increased. Using K=1 it takes more evacuation than using k=2 without VLC, and β=1. When the number of vehicles increased to 1800, the result is not different between using k=1, k=2 with β=0, and k=2 with β=1. It is clearly seen, that using $k = 2$ with β=2 give the best result.

5. Conclusion and Future Work

In this paper, we proposed road network routing algorithm support for multiple destination request, road traffic distribution, and recommend route considering limited resource at destination. We have shown the effectiveness of our proposed algorithm using simulation program based on actual tsunami evacuation environment. We succeed to recommend route to multiple destinations when number of destination equal to 2 in terms of number of vehicle arrived at destination but cannot get the service, and evacuation time spend from beginning of evacuation until all vehicle arrive at destination. This result strongly depends on the road network of a particular town. For a town prone to tsunami or similar natural disaster, we can find the minimum value of k and β from its road network, safety shelter locations and number of vehicles.

In future our aim is to apply the algorithm to support more realistic situations, when number of destinations and destination resources are limited will be different depend on the area.

References

[1] http://www.city.miyako.iwate.jp/kikikanri/miyako_hazardmap.html/. [Online Available;].

[2] E. W. Dijkstra, "A note on two problems in connexion with graphs," *NUMERISCHE MATHEMATIK*, vol. 1, no. 1, pp. 269–271, 1959.

[3] R. Bellman, "On a Routing Problem," *Quarterly of Applied Mathematics*, vol. 16, pp. 87–90, 1958.

[4] D. Delling, P. Sanders, D. Schultes, and D. Wagner, *Algorithmics of Large and Complex Networks: Design, Analysis, and Simulation*, ch. Engineering Route Planning Algorithms, pp. 117–139. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009.

[5] Y. C. Chiu, J. Bottom, M. Mahut, A. Paz, R. Balakrishna, T. Waller, and J. Hicks, "Dynamic traffic assignment: A primer," 2011.

[6] S. Peeta and A. K. Ziliaskopoulos, "Foundations of dynamic traffic assignment: The past, the present and the future," *Networks and Spatial Economics*, vol. 1, pp. 233–265, September 2001.

[7] J. Pan, I. Popa, K. Zeitouni, and C. Borcea, "Proactive vehicular traffic rerouting for lower travel time," *Vehicular Technology, IEEE Transactions on*, vol. 62, pp. 3551–3568, Oct 2013.

[8] J. Y. Yen, "Finding the k shortest loopless paths in a network," *Management Science*, vol. 17, no. 11, pp. 712–716, 1971.

[9] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of SUMO - Simulation of Urban MObility," *International Journal On Advances in Systems and Measurements*, vol. 5, pp. 128–138, December 2012.

[10] M. Haklay and P. Weber, "Openstreetmap: User-generated street maps," *Pervasive Computing, IEEE*, vol. 7, pp. 12–18, Oct 2008.

[11] http://www.openstreetmap.org/. [Online Available;].