TIME-DEPENDENT VEHICLE ROUTING PROBLEM IN BANGKOK

Warinda Kiatdherarat, Department of Electrical and Software Systems Engineering
The Sirindhorn International Thai-German Graduate School of Engineering (TGGS)
King Mongkut's University of Technology North Bangkok
kamoll@kmutnb.ac.th

Pathai Suwannawiwat, Department of Electrical and Software Systems Engineering
The Sirindhorn International Thai-German Graduate School of Engineering (TGGS)
King Mongkut's University of Technology North Bangkok
kamoll@kmutnb.ac.th

Kamol Limtanyakul, Department of Electrical and Software Systems Engineering
The Sirindhorn International Thai-German Graduate School of Engineering (TGGS)
King Mongkut's University of Technology North Bangkok
kamoll@kmutnb.ac.th

Abstract

We suggest a scheme to solve a time dependent vehicle routing problem in Bangkok. In fleet transportation, we usually prefer routes aiming to minimize the number of vehicles and the total distance. However, traffic congestion has significant impact on cost and time for transportation. As traffic information in Bangkok is provided real-time in the internet, we can collect congestion levels of roads and estimate travelling times of vehicles. Based on time-dependent shortest paths between locations, we determine a route of each vehicle to minimize the total travel time, while all vehicles depart from a depot together.

Our preliminary result based on the sampled data is further presented. The change of traffic situation causes ineffectiveness of routes based only on minimizing the total distance. Solving the routing problem using the traffic information can help reduce the total travel time. This study could be further applied to verify and improve the performance of logistic process.

Keyword: Vehicle Routing, Time dependent, Traffic, Logistics, Optimization

INTRODUCTION

We study a time-dependent vehicle routing problem in Bangkok based on available traffic information. In a standard routing problem, we aim to minimize a number of required trucks and the total distance to travel from a distribution center to customers. Nuangrit (2007) studied the case for convenient stores in Thailand and suggested his solving algorithms. He also mentioned a limitation as the travel time can be simply calculated by the constant speed limits of roads.

In fact, traffic congestion significantly varies during a whole day in a big city like Bangkok and has an impact on the actual travel time and the delay of vehicles. Therefore, the time-dependent vehicle routing problem should be studied. Although the real traffic situation can be different from the available information used for planning, the result could help us obtain a routing schedule which is more realistic for time and resource management.

Nowadays, traffic information becomes widely accessible in the internet. In Bangkok, there are several providers e.g. Intelligent Traffic Information Center (www.iticfoundation.org), Traffy(www.traffy.in.th), TSquare Traffic (https://app.rtic-thai.info/tsquare), Longdo Traffic (http://traffic.longdo.com). These systems collect and present information from various sources like probing data from buses, taxis, fleets and passenger cars. In addition, they present congestion levels of roads obtained from Bangkok Metropolitan Administration (http://www.bkktraffic.com).

Moreover, Traffy is a research project to estimate and report traffic information under National Electronics and Computer Technology Center (NECTEC). First, Traffy system collects data from similar sources as mentioned. Also, the system gets accident reports from traffic police and images from CCTV located around Bangkok. After data processing, traffic information is reported on social media like Twitter and Facebook. Programmers can further develop an application using Traffy API (Application Program Interface). Its output files consist of XML, HTML, JSON, CSV and JS.

In this paper, we apply Traffy API to store samples of traffic information in our database. Congestion levels on roads are used to estimate the speed of a vehicle and a travel time. Afterward, we solve a time-dependent shortest path problem as traffic changes from time to time. That means we determine a path using minimal time to travel between any two points. This path can change depending on a departure time. For example, we could prefer another path to avoid congestion during rush hours.

After that, we solve a vehicle routing problem (VRP) to find out customer visits of each vehicle with respect to capacity constraint. We assume a sample of the location of convenient stores in the urban area. There are several advanced techniques for solving a large scale vehicle routing problem. In this preliminary research, we apply a software tool named Drools Planner version 5.5 to solve our small cases. Note that the software has been recently renamed to OptaPlanner (www.optaplanner.org). We assume all vehicles departing from a distribution center at the same initial time. We focus on minimizing the total travel time for all vehicles based on the obtained time-dependent shortest paths between customers. We have not yet considered a departure time of each vehicle as another decision variable. Otherwise, we could select the most suitable initial time for each vehicle depending on traffic situation.
Section 2 reviews algorithms or methods to solve both time-dependent shortest paths and vehicle routing problems. The methods using in this paper are discussed in Section 3. Computational results are shown in Section 4 before ending with conclusion.

LITERATURE SURVEY

In this section, we show some research articles related to solving a time-dependent shortest path and a time-dependent vehicle routing problem (TDVRP). In a map containing all roads and intersections, we need first to determine a shortest path from one point to another. We mainly have to calculate the shortest paths between a distribution center and all customers.

Dean (2004) studied a property of this problem and summarized several shortest path algorithms for a time-dependent network in both continuous and discrete time cases. Dehne and Omran (2012) focused on two types of the time-dependent FIFO network, the availability of the links with their change over time and the cost of using the link. Moreover, Liang Zhao et al. (2011) presented a novel generalized A* algorithm and an ALT algorithm using an appropriate estimator to reduce computation time. Their methods are more suitable especially for a large network than a normal Dijikstra algorithm.

The vehicle routing problem consists of a number of vehicles with its own capacity. There are customer demands to be supplied by a distribution center. We need to find a route representing customers that each vehicle has to visit. The problem aims to minimize a number of vehicles and total distance, while the capacity constraint must be satisfied.

Kumar and Panneerselvam (2012) presented a survey on the vehicle routing problem and its variants. They introduced several algorithms based on exact methods, heuristic approaches, meta-heuristic, and hybrid method. Their study showed that only few researchers use hybrid methods to solve a vehicle routing problem due to the time needed to find the optimal solution will be prohibitive. Rousseau and Gendreau (2002) applied the operator for searching on large neighborhood. They used the pruning and propagation techniques of constraint programming to achieve more efficient search.

As the time-dependent vehicle routing problem is NP-Hard as a generalization of the travelling salesman problem which, several solving methods and heuristics are developed. For example, Potvin et al. (2004) introduced an algorithm to calculate the travel time. Figliozzi (2009) suggested how to construct a route and re-route according to the traffic congestion. Balserio et al. (2011) applied Ant Colony Optimization. Moreover, Kim and Jeon (2010) introduced Genetic Algorithm and Mathematical Programming for make decision with the traffic congestion.

METHODOLOGY

We describe here three main procedures applied to solve our problem. First, traffic information from Traffy has to be estimated in order to calculate a travel time. Then, we select the Dijikstra algorithm modified for finding a shortest path in a time-dependent network. Finally, our method to solve the TDVRP is briefly discussed. In this preliminary step, simple algorithms used in this paper are not yet appropriate for solving very large scale cases.

1. Processing traffic information from Traffy

Traffy API allows us to check traffic status by executing a command Get Traffic Congestion. This API returns three levels of Traffic congestion which consist of high, medium and low. The website of Intelligent Traffic Information Center (www.iticfoundation.org/node/41) suggests a possible range of velocity for roads in a city: High status represents a speed lower than 15 km/hr, Medium status represents a speed between 15-25 km/hr, Low status represents a speed more than 25 km/hr.

To determine a travel time from these congestion levels, we need to assume that the High, Medium, and Low status equivalent to 15 km/hr, 25 km/hr, and 60 km/hr, respectively. Each location in a map is represented by its latitude and longitude. Thus, a distance between two nodes is approximately determined by using the Eucladian norm. Thus, a travel time depends on the distances and congestion levels of roads needed to travel along.

2. Solving Time-dependent Shortest Path

To determine a shortest path from a source node to a destination node, Dijikstra's algorithm is one of standard algorithms. Its computation based on constant values of distances between nodes. However, in a time-dependent network we need to consider a travel time which varies according to current values of congestion levels. Thus, the travel time for each time interval can be different.

Figure 1 shows a modified Dijkstra's alogrihm given by Dean (2004). This algorithm determines $EA_{si}(t)$ which is a shortest path from a source node s to any destination node i, when leaving from the source node at time t. At the beginning, $EA_{si}(t)$ is initialized with a large value. The search explores from the start node to all nodes in neighborhood. To maintain a minimal value, the algorithm finds out whether there exists a better path visiting other node j before going to node i, $a_{ij}(EA_{si}(t))$. Note that this algorithm might not be efficient enough for solving a large network.

```
Initialization
        For all i $\epsilon$ N\{s}: $EA_{si}(t) \leftarrow \infty$
        $EA_{ss}(t) \leftarrow t$
        S $\leftarrow N$
Main Loop
        While S $\neq \emptyset$
          Select i $\epsilon$ S minimizing E$A_{si}$(t)
          S $\leftarrow S\backslash\{i\}$
          For all j such that (i,j) $\epsilon$ A
             $EA_{si}(t) \leftarrow \min\{EA_{si}(t), a_{ij}(EA_{si}(t))\}$
```

Figure 1: Dijikstra's algorithm for solving time-dependent shortest path

3. Solving Time-dependent Vehicle Routing Problem

Drools Planner is a library platform written in Java to solve planning and scheduling problems. It has several internal heuristics and metaheuristics based on score calculation. Many examples are included, e.g. vehicle routing problem, nurse scheduling, bin packing. We further modify its VRP example to calculate a departure time of each node and use a time dependent shortest path for optimization process. Available optimization techniques are applied.

First, a construction heuristic builds a pretty good initial solution in a finite length of time. This initial solution may not be feasible but can help metaheuristics to finish the solving process. In our case, the First Fit Decreasing is applied to allocate the most difficult planning entity/variable first since it is critical to fit in our limited resource. Other remaining entities have less degree of difficulty and could be allocated later.

Afterward, local search starts searching from the initial solution to find out a better one. A number of moves for each solution are evaluated before the best move is applied. The program searches until reaching termination criteria like timeout. Tabu list is also used to improve the hill climbing process and avoid the search getting stuck in local optima.

COMPUTATION RESULTS

The traffic network obtained from Traffy consists of 450 nodes. Figure 2 shows the locations of these nodes and the connected links in red color, while actual roads are in yellow color. We notice that the links obtained from Traffy are not completely connected like a real road network. On some roads traffic information are still missing. Thus, we have to create additional links and assume that the congestion levels of these links are always low.

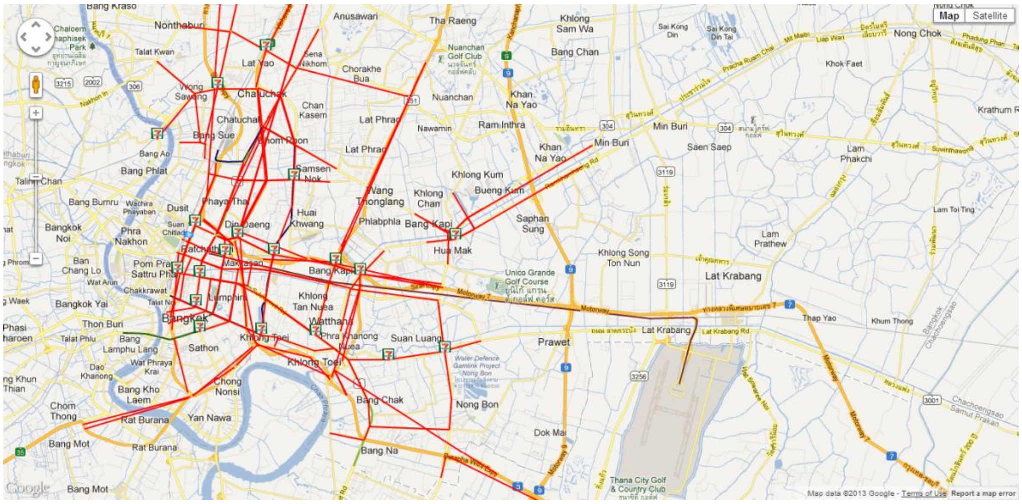

Figure 2 – Available Traffic Information on Links highlighted by Red Color

Moreover, we select 19 nodes around the city center to represent locations of convenient stores. As our primary objective is to minimize a number of required vehicles, we found that three trucks are needed to support the total demand of all stores. The sample data used in our experiment was taken every 30 minutes on Monday, March 11, 2013 between 5.00-9.00 a.m. We assume congestion levels do not change during the 30-min interval.

Table 1 compares two solutions finding routes with the minimum total travel distance and the minimum total travel time. In the first case, we achieve an optimal solution based on the selection of paths using only a shortest distance. The second case applies traffic information to select a path with a shortest travel time. In both cases, all vehicles have to leave from a distribution center at the same initial time. Drools planner and Dijkstra’s algorithm need together around 5 minutes to achieve the solutions.

Table 1 – Comparison of Solutions: Minimizing Total Distance and Minimizing Total Time

Vehicle	Route Minimizing Total Distance		Route Minimizing Total Time	
	Total Distance (km)	Total Time (min)	Total Distance (km)	Total Time (min)
1	41	199	64	225
2	71	198	121	383
3	43	338	32	106
Sum	196	735	217	714

The route with minimizing total time clearly requires less travel time, while its total travel distance becomes larger. However, the difference is not so much because we assume low congestion levels for additional links created to fulfill missing traffic information. The program can always avoid roads with traffic congestion. Also, our primary objective limits us to always use three vehicles for both cases. The vehicles still have to significantly travel across the city.

We can notice also that it is better to estimate the travel time of each vehicle by using traffic information. When the constant speed of 60 km/hr is assumed for all links, the obtained travel time becomes significantly different from the reality. For example, we cannot expect the vehicle 1 spends just 41 minutes to cover its travel distance of 41 km. The traffic information can help predict when each truck can come back to depot and become available for next delivery.

CONCLUSION

Our research solves a vehicle routing problem based on real traffic information. We suggest a scheme to collect and interpret congestion levels from the available resource in the internet. The Dijkstra's algorithm is applied to determine a time-dependent shortest path. In this work, we still assume that all vehicles have to leave a depot at the same time, while a fastest path is selected depending on a current arrival time of each node. Drools planner further solves a vehicle routing problem to minimize the total travel time.

The computation results verify that it could be more efficient to apply a route using less travel time. Although vehicles have to travel longer distance, they can return to a distribution center earlier and can be reused for other deliveries. That means we can reduce the size and fix cost to operate the fleet.

The impact of wasting time could be higher when the number of store increases and transportation during the peak rush hour is inevitable. However, the cover areas, reliability and accuracy of traffic information are still main limitations and have to be evaluated more in practice.

In the near future, the quality of information could be further improved by the service providers since there are more vehicles sharing their location signals to traffic centers via smart phones and a modern mobile network. In addition, our methods to solve a time-dependent shortest path and a time-dependent routing problem have to be improved to achieve better solutions especially for a large network.

ACKNOWLEDGEMENT

The work is supported by a grant from Science and Technology Research Institute, King Mongkut's University of Technology North Bangkok.

REFERENCES

1. Balseiro, SR, Loiseau, I & Ramonet, J 2011, 'An Ant Colony algorithm hybridized with insertion heuristics for the Time-dependent Vehicle Routing Problem with Time Windows'.
2. Dean, B 2004, 'Shortest Paths in FIFO Time-Dependent Networks: Theory and Algorithms', Technical Report, MIT, Cambridge.
3. Dehne, F & Omran, M 2012, 'Shortest Paths in Time-Dependent FIFO Networks', Algorithmica, vol 62, pp. 416-435.
4. Figliozzi, MA 2009, 'A Route Improvement Algorithm for the Vehicle Routing Problem with Time-dependent Travel Times', the 88th Transportation Research Board Annual Meeting.
5. Fleischmann, B, Gnutzmann, S & Sandvoß, E 2004, 'Dynamic Vehicle Routing Based on Online Traffic Information', TRANSPORTATION SCIENCE, vol 38, no. 4, pp. 420-433.
6. Kim, TK & Jeon, G 2010, 'A vehicle routing problem considering traffic situation with time windows by using hybrid genetic algorithm', 40th International Conference on Computers & Industrial Engineering (CIE-40), pp. 1-6.
7. Kumar, S & Panneerselvan, R 2012, 'A Survey on the vehicle routing problem and its variants', Intelligent Information Management, vol 4, pp. 66-74.
8. Nuangrit, A 2007, 'An application of computer programs for the routing of convenience store deliveries', Master Thesis (in Thai), Chulalongkorn University.
9. Roussean, L & Gendreau, M 2002, 'Using Constraint-based operators to solve the vehicle routing problem with time windows', Journal Heuristics, pp. 43-58.
10. Zhao, L, Oshima, T & Nagamochi, H 2011, 'Algorithm for time-dependent shortest path problem', in Computational Geometry, Graphs and Applications, Springer Berlin Heidelberg.