Algorithms to Quantify the Impacts of Congestion on Time-Dependent Real-World Urban Freight Distribution Networks

Ryan G. Conrad

conradr@cecs.pdx.edu

Miguel Andres Figliozzi*

figliozzi@pdx.edu

Portland State University

Department of Civil and Environmental Engineering

*Corresponding author

Submitted to the 89th Annual Meeting of the Transportation Research Board

January 10–14, 2010

September 15, 2009

Number of words: 5118 + 8 Figures = 7118

ALGORITHMS TO QUANTIFY THE IMPACTS OF CONGESTION ON TIME-DEPENDENT REAL-WORLD URBAN FREIGHT DISTRIBUTION NETWORKS

Ryan G. Conrad

Miguel Andres Figliozzi

Abstract

Urban congestion presents considerable challenges to time-definite transportation service providers. Package, courier, and less-than-truckload (LTL) operations and costs are severely affected by increasing congestion levels. With congestion increasing at peak-hour morning and afternoon periods, public policies and logistics strategies that avoid or minimize deliveries during congested periods have become crucial for many operators and public agencies. However, in many cases these strategies or policies can introduce unintended side-effects such as higher labor costs, shorter working hours, and tighter customer time windows. Research efforts to analyze and quantify the impacts of congestion are hindered by the complexities of vehicle routing problems with time-dependent travel times and the lack of network-wide congestion data. This research utilizes: (a) real-world road network data to estimate travel distance and time matrices, (b) land-use and customer data to localize and characterize demand patterns, (c) congestion data from an extensive archive of freeway and arterial street traffic sensor data to estimate time-dependent travel times, and (d) an efficient time-dependent vehicle routing (TDVRP) solution method to design routes. Novel algorithms are developed to integrate real-world road network and travel data to TDVRP solution methods. Results are presented to illustrate the impact of congestion on depot location, fleet size, and distance traveled.

1. Introduction

Congested urban areas present considerable challenges for LTL (less-than-truckload) carriers, courier services, and industries that require frequent and time-sensitive deliveries. With congestion increasing at peak-hour morning and afternoon periods, public policies and logistics strategies that avoid or minimize deliveries during congested periods have become crucial for many operators and public agencies. However, in many cases, these strategies and policies can introduce unintended side-effects such as higher labor costs, shorter working hours, and tighter customer time windows.

While current research on vehicle routing algorithms is extensive, much less attention has been devoted to investigating the impacts of congestion on carrier operations. Furthermore, most algorithms to solve the time-dependent vehicle routing problem (TDVRP) found in the existing literature do not deal with the estimation of distance and time-dependent travel time matrices. Thus, this research focuses on two primary objectives: (a) develop efficient algorithms to apply TDVRP solution methods to actual road networks using historical traffic data with a limited increase in computational time and memory, and (b) to utilize Google Maps™ open-source application programming interface (API) and network data to produce distance and travel time matrices. To the best of the authors’ knowledge, no research effort has integrated time-dependent routing algorithms, historical traffic data, real-world road network data, and public open-source APIs to incorporate the impacts of congestion on delivery routes.

2. Literature Review

This section covers two main areas of research: (a) the effects of congestion and travel time variability on vehicle routes and logistics operations, and (b) TDVRP solution algorithms and their application to urban areas.

Direct and indirect costs of congestion on passenger travel time, shipper travel time and market access, production, and labor productivity have been widely studied and reported in the available literature. The work of Weisbrod et al. [1] provides a comprehensive review of this literature. Substantial progress has also been made in the development of econometric techniques to study the joint behavior of carriers and shippers in regards to congestion [2, 3].

Survey results suggest that the type of freight operation has a significant influence on how congestion affects carriers’ operations and costs. Survey data from California indicate that congestion is perceived as a serious problem for companies specializing in LTL, refrigerated, and intermodal cargo [4]. Similar conclusions are reached by reports analyzing the effects of highway limitations and traffic congestion in the Portland region [5, 6]. A positive relationship between the level of local congestion and the purchase of routing software is identified by Golob and Regan [7]. Carriers that do not follow regular routes, e.g. for-hire carriers, tend to place a higher value on the usage of real-time information to mitigate the effects of congestion and logistical services to plan fleet deployments [8]. Other researchers attribute the scant usage of TDVRP algorithms to the lack of reliable time-dependent travel time data, which can be particularly expensive or difficult to obtain for small carriers [9]. These authors recommend the implementation of open-access online TDVRP and data services in order to increase the efficiency of routes in congested urban areas.

Another line of research has investigated carriers' reactions to toll measures intended to shift freight traffic to off-peak hours. Holguin-Veras et al. [10] investigate the effects of congestion charges in New York City and find that delivery times are heavily dictated by customer time windows. Congestion charges increase carriers' operating costs while inducing little shifting of deliveries from peak to off-peak hours. This suggests an inelastic relationship between freight congestion charges and routes with time definite delivery times. Quak and Koster [11] present a methodology to quantify the impacts of delivery constraints and urban policies utilizing a fractional factorial regression. Quak and Koster find that vehicle restrictions and delivery curfews have a compounding effect on customer costs whereas vehicle restrictions alone are costlier only when vehicle capacity is limited. Research into the effects of congestion on vehicle route characteristics is limited. Figliozzi [12] analytically models routes, extending Daganzo's continuous approximations [13], and analyzes how routing constraints and customer service durations affect route characteristics using a classification based on supply chain characteristics. This analysis shows that a decrease in travel speed severely affects total distance traveled for routes with time window constraints while capacity constrained routes are less affected. The impact of travel time reliability on LTL delivery is also analyzed using continuous approximations and real-world data [14]. This research concludes that travel time variability has a significant impact in carriers' costs when average distance to delivery areas increases and average travel speed decreases.

Classic versions of the vehicle routing problem (VRP) such as the capacitated VRP (CVRP) or VRP with time windows (VRPTW) have been widely studied. However, time-dependent problems have received considerably less attention. A comprehensive review of TDVRP approaches and an efficient TDVRP algorithm is presented by Figliozzi [15]. This work also creates benchmark problems for the TDVRP altering the classical VRPTW Solomon instances. Fleischmann et al. [16] reviews the adaptation of the VRP algorithms to time-dependent data from traffic information systems in the city of Berlin. The construction of a time-dependent travel time database is also analyzed by Eglese et al. [17] utilizing Dijkstra's algorithm for time-dependent links. Eglese et al. apply their methodology to a real-world network in England. However, these research efforts [16, 17] do not incorporate into their analysis the influence of time windows and recurring bottlenecks or the impacts of congestion on fleet size and total distance traveled.

3. Portland Case Study

Considered a gateway to international sea and air freight transport, the city of Portland has established itself as an important hub for international and domestic freight movements. Its favorable geography to both International Ocean and domestic river freight via the Columbia River is complemented by its highway connections. Interstate-5 (I-5) is the most important freeway connecting the West Coast from Mexico to Canada as well as southern California ports and main West Coast population centers [5]. The I-5 freeway is also used by many carriers delivering in Portland and the city's surrounding suburbs because it also provides the main north-south freight corridor through the city of Portland itself.

Recent increases in regional traffic congestion have negatively impacted freight operations. A recent report investigates the impacts of congestion on Portland-area businesses and LTL deliveries [5]. This report provides insightful, yet qualitative information, on various strategies employed by businesses to cope with congestion, additional delivery costs, and uncertainty. The report indicates that congestion has made some afternoon deliveries completely infeasible which requires deliveries during non-business hours early in the morning. However, avoiding congesting by shifting deliveries to early morning periods generate additional costs by reducing route durations. In some cases, early deliveries are not feasible in close proximity to residential areas where parking problems and noise can lead to sound/traffic ordinance violations and conflicts with residents [5].

The recurrent effects of traffic congestion at peak periods present daily challenges to LTL carriers in the Portland metropolitan area. The numerical analysis presented in Section 7 aims to represent the above mentioned conditions. Customer data and depot locations are generated using a land-use zoning map of the Portland metropolitan area as detailed in Section 6. Network and congestion data sources, including recurrent bottlenecks, are described in Section 4. A methodology to apply TDVRP algorithms to real-world networks is described in Section 5. The methodologies and algorithms developed in this research assume that customers' demands and time windows are known a priori, e.g. the night before delivery. Congestion data related to non-recurrent conditions, e.g. due to accidents, is not analyzed and left as a future research topic.

4. Data Sources

Two main data sources were utilized in this research: Google Maps API for the implementation of the TDVRP algorithm and the Portland Transportation Archived Listing (PORTAL) for obtaining historical travel time data. These two sources are described in the following subsections.

Overview of the Google Maps API

The use of the Google Maps API allows access to up to date street network in the studied region with a high level of geographical detail. The open-source nature of the application also allows for considerable freedom in modifying the program and user interface [18]. Figure 1 shows the process of creating customer distributions and obtaining optimized routes from the TDVRP algorithm as implemented with the API. The API consists of several interfaces:

- A customer selection screen where a set of customers and a single depot can be created by clicking on locations on the map. A coordinate output is provided that is then copied into a text (.txt) file
- An interface that calculates the shortest paths between pairs of customers and constructs the distance and travel time Origin-Destination (O-D) matrices. Distance and travel time matrices are estimated and stored as text files.
- Travel speed, occupancy and vehicle flow data from traffic sensors are used to incorporate the impact of congestion on travel times.
- A solution interface where solution sets outputted from the TDVRP algorithm can be loaded and plotted to provide a visual verification of results.

Perhaps the greatest advantage of the API is that the open-source software and high quality network data can be accessed free of charge[1]. This together with the TDVRP solution algorithm developed to interface with the API offers the potential for very low cost solutions for route planning and optimization while accessing detailed and accurate network data such as road hierarchy and restrictions (e.g. one-way streets or no-left turn movements at intersections). The effects of congestion are included by modifying the travel times initially calculated by Google Maps. After the TDVRP algorithm design the routes, the API interface can be utilized to obtain detailed driving directions.

<< INSERT FIGURE 1>>

[1] http://code.google.com/apis/maps/

Simulating Congestion Effects

Google Maps already provides reasonable travel time estimations during uncongested periods. However, to increase the accuracy of travel time estimations highway sensor data are utilized. For example, segments along Interstate 5 located in proximity to traffic bottlenecks are selected to represent areas of decreased travel speed. The selected segments are between freeway interchanges and/or on/off-ramps where vehicle detector loops are located.

Detailed traffic data are obtained PORTAL, Portland's implementation of an Archived Data User Service (ADUS) which coordinates and obtains data from approximately 436 inductive loop detectors along interstate freeways in the Portland metropolitan area. A description of this transportation data archive is given by Bertini et al. [19]. Bottlenecks are modeled as point locations surrounded by areas of reduced travel speed. Travel in proximity to a bottleneck is expressed as a percent reduction in travel speed proportional to the speed reduction at the bottleneck location. Figure 2 shows the bottleneck locations and areas of effective travel speed reduction.

<< INSERT FIGURE 2>>

Data obtained from PORTAL are also used to model the impacts of traffic queuing on the surrounding network. The areas of reduced travel speed for each bottleneck location are assumed as a function of the measured occupancy and vehicle inflow and outflow rates at each bottleneck location. Research has shown that traffic queues often begin to form at occupancies approximately equal to or greater than 20% [20], but according to speed flow data queues may form at occupancies as low as 13%. Utilizing these queuing concepts and assumptions, the radius of the area of travel speed reduction around each bottleneck where vehicle travel speed reduced is varied in proportion to the difference in the inflow and outflow rates multiplied by average vehicle spacing when the occupancy is above a certain threshold value. Strictly, this assumes that there is conservation of vehicles (i.e. no vehicles enter or exit the road segment in question) and ignores the presence of moving traffic queues.

The travel speeds used in this research are calculated from 15 minute archived travel time data averaged over the year 2007 along the I-5 freeway corridor spanning from the Portland suburb of Wilsonville to Vancouver, Washington. These data are sufficient for purposes of demonstrations of the proposed methodology, but consideration of seasonal or monthly variability in travel time is important for many LTL carriers and is entirely feasible via PORTAL. In this research it is assumed that carriers only account for recurrent congestion and plan their routes the night before making the deliveries.

5. Methodology

TDVRP Algorithm Overview

A description of the TDVRP algorithm used in this experiment along with a full TDVRP formulation is presented by Figliozzi [15]. With hard time window constraints the primary objective is the minimization of the number of vehicles or routes; the secondary objective is the minimization of the travel time or distance. The TDVRP solution algorithm consists of a route construction phase and a route improvement phase, each utilizing two separate algorithms (FIGURE 3). During route construction, the auxiliary routing algorithm $\boldsymbol{H_r}$ determines feasible routes with the construction algorithm $\boldsymbol{H_c}$ assigning customers and sequencing the routes. Route improvement is done first with the route improvement algorithm $\boldsymbol{H_i}$ which compares similar routes and consolidates customers into a set of improved routes. Lastly, the service time improvement algorithm $\boldsymbol{H_y}$ eliminates early time window violations, and then reduces the route duration without introducing additional early or late time window violations; these tasks are accomplished by using the arrival time and departure time algorithms $\boldsymbol{H_{yf}}$ and $\boldsymbol{H_{yb}}$, respectively, and customers are subsequently re-sequenced as necessary. It is with these algorithms that the PORTAL data and shortest-path travel speeds generated by the Google Maps API are inserted into the solution algorithm.

Notation

For the following travel time algorithms, the total depot working time $[e_{\#}, l_{\#}]$ is partitioned into a set of p time periods $\mathbf{T_p} = \{T_1, T_2, \dots, T_p\}$. Each traffic bottleneck locations $\beta_m \in \boldsymbol{\beta_n} = \{\beta_1, \beta_2, \dots \beta_n\}$ is assigned the following data at each time partition $T_k \in \mathbf{T_p}$:

- $\mathbf{O_n^p} = [O_{km}]_{p\times n}$: The table of occupancy values for each time period $T_k \in \mathbf{T_p}$ and bottleneck $\beta_m \in \boldsymbol{\beta_n}$
- $\mathbf{U_{n+1}^p} = [U_{km}]_{p\times n+1}$: Table of vehicle flow inflow and outflow rates for each time period and bottleneck locations. The inflow and outflow rates at time period T_k for bottleneck β_m are U_{km} and $U_{k,m+1}$, respectively
- $\mathbf{v_n^p} = [v_{km}]_{p\times n}$: Table of congested travel speeds obtained from PORTAL

All data are collected from PORTAL and the point source location of each traffic bottleneck is assumed to be midway between detector loops. The algorithms also include the following adjustable parameters for each bottleneck location:

- $\bar{R}_m \in \bar{\mathbf{R}}_\mathbf{n} = \{\bar{R}_1, \bar{R}_2, \dots, \bar{R}_m, \dots, \bar{R}_n\}$: A set of initial radius values at time $t = 0$
- $\bar{L}_m \in \bar{\mathbf{L}}_\mathbf{n} = \{\bar{L}_1, \bar{L}_2, \dots, \bar{L}_m, \dots, \bar{L}_n\}$: A set of average vehicle spacing values
- $\bar{O}_m \in \bar{\mathbf{O}}_\mathbf{n} = \{\bar{O}_1, \bar{O}_2, \dots, \bar{O}_m, \dots, \bar{O}_n\}$: A set of threshold occupancy percentages that determine the expected onset of traffic queuing
- $\bar{v}_m \in \bar{\mathbf{v}}_\mathbf{n} = \{\bar{v}_1, \bar{v}_2, \dots, \bar{v}_m, \dots, \bar{v}_n\}$: A set of free-flow speeds

For the sake of readability, the Appendix contains a complete listing of variable and function definitions as well as notational conventions.

<< INSERT FIGURE 3>>

Traffic Queuing Algorithm

The following is a summary of the $\boldsymbol{H_{yc}}$ algorithm that assembles a table of bottleneck radii R_{km} for each bottleneck β_m and time period T_k. The algorithm requires the input data arrays $\mathbf{O}_\mathbf{n}^\mathbf{p}$ and $\mathbf{U}_\mathbf{n+1}^\mathbf{p}$ as well as the adjustable parameters $\bar{\mathbf{R}}_\mathbf{n}$, $\bar{\mathbf{L}}_\mathbf{n}$ and $\bar{\mathbf{O}}_\mathbf{n}$. The output table $\mathbf{R}_\mathbf{n}^\mathbf{p}$ contains the radius value for each time period T_k at each bottleneck β_m in a $p \times n$ array. The complete pseudo-code is provided in the Appendix; beginning with the conditional statement within the nested for-loop for a particular β_m and starting at $t = 0$, the algorithm can be described as follows:

1. First assign the variable R the base parameter value $\bar{R}_m$ at $t = 0$
2. Begin the k iteration; if the occupancy O_{km} at a given k iteration is greater than the threshold value $\bar{O}_m$, add the differences in the outflow and inflow traffic volumes multiplied by the duration of the time partition $t_{\bar{k}} - t_{\underline{k}}$ by the average vehicle spacing $\bar{L}_m$ to the variable R
3. If the occupancy O_{km} is less than $\bar{O}_m$ and the radius variable R is greater than the base parameter $\bar{R}_m$, then subtract the quantity from step 2 from R.
4. Take the maximum of the set $[R, \bar{R}_m]$; this and the second condition of step 3 prevent R from being assigned a negative value and ensures that $\bar{R}_m$ is a lower bound for the variable R when the predicted traffic queue is dispersing
5. Otherwise, retain $R = \bar{R}_m$

6. Construct a column vector $\mathbf{R^p}$ of R values obtained from each k iteration
7. Repeat steps 1 through 6 n times and construct the output matrix $\mathbf{R_n^p}$ from the column vectors $\mathbf{R^p}$ obtained from each iteration.

In summary, the $\boldsymbol{H_{yr}}$ algorithm adds or subtracts expected lengths of traffic queues to the radius of the effective area of each bottleneck which is dependent on whether the measured occupancy is above or below each threshold value contained in $\mathbf{\bar{O}_n}$. The table of values in $\mathbf{R_n^p}$ is referenced by the $\boldsymbol{H_{yf}}$ and $\boldsymbol{H_{yb}}$ algorithms described in detail in the following section. The objective is to extrapolate travel time trends from the data that are available and apply them to the surrounding road network.

Arrival and Departure Time Algorithms

The following is a summary of the arrival time and departure time algorithms $\boldsymbol{H_{yf}}$ and $\boldsymbol{H_{yb}}$ adapted from Figliozzi [15] that estimate travel times between pairs of customers β_i and β_j using the travel time data. The $\boldsymbol{H_{yf}}$ algorithm calculates the expected arrival time at a customer β_j when departing from a previous customer β_i using a forward-iterative process. Similarly, the $\boldsymbol{H_{yb}}$ algorithm utilizes a backward iterative process and simultaneously calculates the required departure time from customer β_i to reach customer β_j.

The impact of bottlenecks as vehicles are moving through different periods of time is a function of the estimated distance between the vehicle and the bottleneck at the beginning of each time period. A linear approximation of the vehicle location is used to reduce computational complexity because shortest path and Euclidean distances are highly correlated. High levels of correlation between Euclidian and shortest path distances are usually found in urban areas [21]. The distance traveled along the Euclidean connecting line is calculated as a percentage of the actual route traversed such that

$$d' = \frac{d}{d_{ij}} D_{ij}. \tag{1}$$

Using the law of cosines (see FIGURE 4) the distance from a point on the Euclidian connecting line to each bottleneck at a given time iteration in the forward iterative calculation can be shown to be

$$r_m = \sqrt{\left(\frac{d}{d_{ij}} D_{ij}\right)^2 + D_{jm}^2 - \frac{d\left(D_{ij}^2 + D_{jm}^2 - D_{im}^2\right)}{d_{ij}}}. \tag{2}$$

Similarly for the backwards iterative process of the departure time algorithm the distance from the nearest bottleneck is

$$r_m = \sqrt{\left(\frac{d}{d_{ij}} D_{ij}\right)^2 + D_{im}^2 - \frac{d\left(D_{ij}^2 + D_{im}^2 - D_{jm}^2\right)}{d_{ij}}}. \tag{3}$$

In equations (2) and (3) D_{ij}, D_{im}, and D_{jm} are the Euclidean distances between customers i and j; customer i and bottleneck β_m; and customer j and bottleneck β_m, respectively; d_{ij} is the shortest-path driving distance from customer i to customer j calculated by the API; and d is the iterated distance from i to j along the actual driving route. A derivation of this function can be found in the Appendix.

<< INSERT FIGURE 4>>

The travel speed function s_m is applied at each time iteration T_k and calculates a speed value for each bottleneck. This function calculates congested travel speeds s_m as reductions in the API-derived speed u_{ij} proportional to the speed reduction measured at the traffic bottlenecks such that $\frac{s_m}{u_{ij}} = \frac{v_{km}}{\bar{v}_m}$ if the virtual location on the Euclidean connecting line is within the radius R_{km}. Here v_{km} is the time-varying speed obtained from PORTAL and $\bar{v}_m$ is an adjustable parameter that may represent the freeway free-flow speed. In other words, the reduction in travel speed due to congestion in the surrounding network is assumed to be proportional to the reduction observed from the PORTAL freeway data at the bottleneck (detector station) with the slowest travel speed. This function can be expressed as

$$s_m = \begin{cases} \frac{u_{ij} v_{km}}{\bar{v}_m} & r_m \leq R_{km} \\ u_{ij} & r_m > R_{km} \end{cases} \tag{4}$$

where r_m is the distance from a point along the Euclidean connecting line to a bottleneck β_m.

The following is a summary of the $\boldsymbol{H_{yf}}$ algorithm; the pseudo-code can be found in the Appendix:

1. First determine if the arrival time a_i is less than the lower time window e_i at customer i
   a. If so, then the vehicle waits and the expected departure time is e_i plus the service time g_i
   b. If not, then the departure time is simply the arrival time plus the service time
2. Determine k for the discrete time period T_k with bounds $[t_{\underline{k}}, t_{\overline{k}}]$ that the expected departure time b_i lies in. This is the initial value for the iterator in the while loop
3. Determine the Euclidean distance of each traffic bottleneck to the location $\beta_i = (x_i, y_i)$ of customer i; the speed function is calculated for each value m and a row vector $\mathbf{S_n}$ of speeds is assembled. The initial travel speed of the vehicle in the subsequent forward-iterative process is calculated as the minimum value of $\mathbf{S_n}$, i.e. the travel speed is only as fast as that imposed by the bottleneck with the worst travel speed (only among the subset of bottlenecks whose area of influence affects the path between customers at a given time).
4. Terminate the while loop when the vehicle has reached its destination. In each period speeds are recalculated and distances accumulated until the vehicle has reached its destination.

Output: the expected arrival time a_j at customer j when departing from customer i at time b_i.

The $\boldsymbol{H_{yb}}$ algorithm works in a similar fashion; given a customer j at location ν_j with an expected arrival time a_j obtained from the $\boldsymbol{H_{yf}}$ algorithm, determine the required departure time b_i from customer i at location β_i to make the trip between β_i and β_j without allowing for late time window violations.

Calibration

Travel times can be calibrated by adjusting $\overline{\mathbf{R}}_{\mathbf{n}}, \overline{\mathbf{L}}_{\mathbf{n}}, \overline{\mathbf{O}}_{\mathbf{n}}, \overline{\mathbf{v}}_{\mathbf{n}}$ parameters as well as the time dependent travel speeds provided by PORTAL ($\mathbf{v}_{\mathbf{n}}^{\mathbf{p}}$). Directional and time of day effects can be incorporated. Memory requirements are reduced because the algorithms work with one travel time and distance matrix. Simple linear functions and intuitive parameters are used to adapt free-flow travel times to congested conditions.

6. Experimental Setting

Sensitivity Analysis and Constraint Modeling

To test the model using real-world constraints, two delivery periods are modeled and analyzed: (1) An early morning delivery period that avoids most of the morning peak-hour traffic congestion but with tighter time windows; and (2) an extended morning delivery time that increases the feasible working time but with increased travel during morning peak-hour. Figure 6 provides a qualitative comparison of the simulated delivery times.

A total of 50 customer locations are utilized (FIGURE 5), with constraints assigned according to the zoning criteria. All customers normally served after 9AM are assumed to be able to shift delivery times prior to this time. Time windows of 15 minutes are randomly assigned to all customer types. Additionally, deliveries to all customers in mixed-use and residential areas are prohibited before 7AM to model required compliance with local noise ordinances. In the early morning delivery option, this reduces the effective depot working time to just two hours for these customers. The extended morning delivery option provides a 4-hour working time for these customers but includes the effects of the morning peak-hour congestion to a greater degree. The calibration of the model was tested by varying the travel speed parameters $\bar{\mathbf{v}}_{\mathbf{n}}$ to alter the simulated travel speed derived from the PORTAL travel time data and contained in the travel speed table $\mathbf{v}_{\mathbf{n}}^{\mathbf{p}}$.

<< INSERT FIGURE 5>>

<< INSERT FIGURE 6>>

7. Experimental Results

Results comparing the number of vehicles and total distance traveled during the morning and extended morning delivery periods are presented in this section. In addition, to incorporate the impact of travel time reliability, time-varying travel speed from PORTAL are decreased by a coefficient δ. This adjustment maintains the overall trend in travel speed variation throughout the delivery period, but allows for adjustments to the travel time to more accurately reflect real-world differences between *average* travel speeds and the actual *distribution* of travel speeds. A value $\delta = 1$ utilizes *average* time-varying travel speed PORTAL data and assumes that no hard time window violations take place if realized travel times are at least the average travel speed. However, if the carriers would like to account for travel time unreliability a value of $\delta < 1$ can be used in the calculations as follows:

$$s_m = \begin{cases} \delta \dfrac{u_{ij} v_{km}}{\bar{v}_m} & r_m \leq R_{km} \\ u_{ij} & r_m > R_{km} \end{cases} \tag{5}$$

A value of $\delta < 1$ guarantees a higher value of customer service [14]. The sensitivity to travel time unreliability and buffer times was tested by setting the parameter $\delta = \{0.4, 0.6, 0.8, 1\}$.

Impact of congestion on the number of vehicles

For the number of required vehicles (FIGURE 7), the central depot showed less sensitivity to changes in travel time reliability than the suburban depot. As expected [14], reduced travel speed appears to have a greater impact on fleet size when the depot has a suburban location. The number of vehicles required is consistently less for the extended early morning delivery period and larger fleet is still required when the depot has a suburban location.

<< INSERT FIGURE 7>>

Impact of congestion on the total distance traveled

Comparisons of total vehicle miles traveled (VMT) are provided in FIGURE 8. Similar to the required number of vehicles, total VMT is significantly higher for tours originating at the suburban depot location. Constrained service times for customers in the early morning delivery period also appear to impact total VMT to a slightly greater extent than travel speed.

<< INSERT FIGURE 8>>

8. Conclusions

This research proposes a new methodology for integrating real-world road networks and travel data to time-dependent vehicle routing solution methods. The use of traffic sensor data and Google Maps API provides a unique approach to interface routing algorithms, travel time and congestion data. Intuitive algorithms and parameters are used to incorporate the impacts of congestion on time-dependent travel time matrices. The proposed methodology is a significant improvement in terms of representing the impacts of congestion in congested urban areas leveraging on existing open source data and applications.

The results show the dramatic impacts of congestion on carriers' fleet sizes and distance traveled. The results also suggest that congestion has a significant impact on fleet size, particularly for depots located in suburban areas outside of the customer service area.

Acknowledgements

The authors extend their sincere gratitude to the Oregon Transportation, Research and Education Consortium (OTREC), the Port of Portland, and Portland State University Research Administration for sponsoring this research. The authors also thank Myeonwoo Lim, Computer Science Department at Portland State University, for assisting with the computer programming aspects of this project, and Nikki Wheeler, Department of Civil and Environmental Engineering at Portland State University, for compiling and aggregating the PORTAL travel time data used in the final computer model.

Appendix A

Notational Conventions

$$\mathbf{A}_{\mathbf{n}}^{\mathbf{p}} = \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1m} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2m} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots & \cdots & \vdots \\ A_{k1} & A_{k2} & \cdots & A_{km} & \cdots & A_{kn} \\ \vdots & \vdots & \cdots & \vdots & \ddots & \vdots \\ A_{p1} & A_{p2} & \cdots & A_{pm} & \cdots & A_{pn} \end{pmatrix}$$: A matrix with p rows and n columns

$\mathbf{A}_{\mathbf{n}} = \{A_1, A_2, \dots, A_m, \dots, A_n\}$: Row vector with n elements

$$\mathbf{A}^{\mathbf{p}} = \left\{\begin{matrix} A_1 \\ A_2 \\ \vdots \\ A_k \\ \vdots \\ A_p \end{matrix}\right\}$$: Column vector with p elements

$\mathbf{A}_{\mathbf{m}} \cup A_{m+1} \equiv \{A_1, A_2, \dots, A_m\} \leftarrow A_{m+1} = \{A_1, A_2, \dots, A_m, A_{m+1}\}$

$$\mathbf{A}_{\mathbf{m}}^{\mathbf{p}} \cup \mathbf{A}^{\mathbf{p}} \equiv \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1m} \\ A_{21} & A_{22} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ A_{p1} & \cdots & \cdots & A_{pm} \end{pmatrix} \leftarrow \begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_p \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1m} & A_{1,m+1} \\ A_{21} & A_{22} & \cdots & A_{2m} & A_{2,m+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ A_{p1} & \cdots & \cdots & A_{pm} & A_{p,m+1} \end{pmatrix}$$

$A_m \in \mathbf{A}_{\mathbf{n}}$: m^{th} element of a row vector with n elements

$A_k \in \mathbf{A}^{\mathbf{p}}$: k^{th} element of a column vector with p elements

$A_{km} \in \mathbf{A}_{\mathbf{n}}^{\mathbf{p}}$: element in the k^{th} row and m^{th} column of a $p \times n$ matrix (p rows and n columns)

Variables	Definition
i, j, m	Indices for set of consecutive customers (i, j) and bottlenecks (m)
$\beta_i = (x_i, y_i);\ \beta_j = (x_j, y_j);\ \beta_m = (x_m, y_m)$	Geographic coordinates of customer i, customer j and bottleneck m, respectively
a_j	Arrival time at customer j
b_i	Departure time from customer i
e_i	Lower time window for customer i
g_i	Service time at customer i
d	Iterated driving distance variable
d_{ij}	Driving distance between customers i and j calculated by the Google Maps API
t_{ij}	Free-flow travel time between customers i and j calculated by the Google Maps API
$u_{ij} = \frac{d_{ij}}{t_{ij}}$	"Free-flow" speed used in TDVRP algorithm
Array/Vector quantities	**Definition**
$T_k \equiv [t_{\bar{k}}, t_{\underline{k}}] \in \mathbf{T_p} = \{T_1, T_2, \ldots T_p\}$,	Set of time periods as fraction of depot working time
$\bar{R}_m \in \bar{\mathbf{R}}_\mathbf{n} = \{\bar{R}_1, \bar{R}_2, \ldots, \bar{R}_n\}$	A set of initial radius values at each bottleneck location at time $t = 0$
$\bar{L}_m \in \bar{\mathbf{L}}_\mathbf{n} = \{\bar{L}_1, \bar{L}_2, \ldots, \bar{L}_n\}$	A set of average vehicle spacing values for each bottleneck location
$\bar{O}_m \in \bar{\mathbf{O}}_\mathbf{n} = \{\bar{O}_1, \bar{O}_2, \ldots, \bar{O}_n\}$	A set of threshold occupancy percentages that determine the expected onset of traffic queuing
$\bar{v}_m \in \bar{\mathbf{v}}_\mathbf{n} = \{\bar{v}_1, \bar{v}_2, \ldots, \bar{v}_n\}$	Bottleneck speed parameters
$U_{km}, U_{k,m+1} \in \mathbf{U}^\mathbf{p}_\mathbf{n+1} = [U_{km}]_{p \times n+1}$	Table of vehicle flow inflow and outflow rates for each time period and bottleneck
$\mathbf{O}^\mathbf{p}_\mathbf{n} = [O_{km}]_{p \times n}$	Table of occupancy values for each time period and bottleneck
$\mathbf{v}^\mathbf{p}_\mathbf{n} = [v_{km}]_{p \times n}$	Speed at bottleneck B_m for the k^{th} time period entered as a $p \times n$ array
Functions	**Definition**
$f(x_\mu, x_\nu, y_\mu, y_\nu)$	Euclidean distance between two sets of x-y coordinates

Bottleneck Radius Algorithm $\boldsymbol{H_{yr}}$

Input

$\mathbf{O_n^p}$, $\mathbf{R_n}$, $\mathbf{U_{n+1}^p}$, $\mathbf{L_n}$, $\mathbf{\bar{O}_n}$

START $\boldsymbol{H_{yc}}$

$\mathbf{R_n^p} \leftarrow \emptyset$

For $m \in \mathbb{N} = 1$ to n

    $\mathbf{R^p} \leftarrow \emptyset$

    $R \leftarrow R_m$

    For $k \in \mathbb{N} = 1$ **To** p

        If $O_{km} \geq \bar{O}_m$ **Then**

            $R \leftarrow R + |U_{k,m+1} - U_{km}|(t_{\bar{k}} - t_{\underline{k}})L_m$

        Else

            If $O_{km} < \bar{O}_m$ **And** $R > \bar{R}_m$ **Then**

                $R \leftarrow R - |U_{k,m+1} - U_{km}|(t_{\bar{k}} - t_{\underline{k}})L_m$

            Else

                $R \leftarrow R_m$

            End If

        End If

        $\mathbf{R^p} \cup R$

    Next k

    $\mathbf{R_n^p} \cup \mathbf{R^p}$

Next m

Output: $\mathbf{R_n^p} = [R_{km}]_{p\times n}$

Arrival time algorithm $\boldsymbol{H_{yf}}$

Input

$\mathbf{R_n^p}$, $\mathbf{T_p}$, $a_i, b_i, e_i, g_i, v_i, v_j, u_{ij}$

START $\boldsymbol{H_{yf}}$

If $a_i < e_i$ **Then**

$b_i \leftarrow e_i + g_i$

Else

$b_i \leftarrow a_i + g_i$

End If

$D_{ij} \leftarrow f(x_i, x_j, y_i, y_j)$

Find $k, t_{\underline{k}} \leq b_i \leq t_{\bar{k}}$

$k \leftarrow k$

$\mathbf{S_n} \leftarrow \emptyset$

For $m = 1 \in \mathbb{N}$ **To** n

$D_{jm} \leftarrow f(x_j, x_m, y_j, y_m)$

$D_{im} \leftarrow f(x_i, x_m, y_i, y_m)$

$r \leftarrow \sqrt{\left(\frac{d}{d_{ij}} D_{ij}\right)^2 + D_{jm}^2 - \frac{d}{d_{ij}}\left(D_{ij}^2 + D_{jm}^2 - D_{im}^2\right)}$

If $r \leq R_{km}$ **Then**

$s_m \leftarrow u_{ij} \dfrac{v_{km}}{w_m}$

Else

$s_m \leftarrow u_{ij}$

End If

$\mathbf{S_n} \leftarrow \mathbf{S_n} \cup \{s_m\}$

Next m

$s \leftarrow \min[\mathbf{S_n}]$

$d \leftarrow d_{ij}$

$a_j \leftarrow b_i + \frac{d}{s}$

$t \leftarrow b_i$

While $a_j > t_{\bar{k}}$ **Do**

$d \leftarrow d - (t_{\bar{k}} - t)s$

$t \leftarrow t_{\bar{k}}$

$k \leftarrow k + 1$

$\mathbf{S_n} \leftarrow \emptyset$

For $m \in \mathbb{N} = \mathbf{1}$ **To** n

$D_{jm} \leftarrow f(x_j, x_m, y_j, y_m)$

$D_{im} \leftarrow f(x_i, x_m, y_i, y_m)$

$r \leftarrow \sqrt{\left(\frac{d}{d_{ij}} D_{ij}\right)^2 + D_{jm}^2 - \frac{d}{d_{ij}}\left(D_{ij}^2 + D_{jm}^2 - D_{im}^2\right)}$

If $r \leq R_{km}$ **Then**

$s_m \leftarrow u_{ij} \dfrac{v_{km}}{w_m}$

Else

$s_m \leftarrow u_{ij}$

End If

$\mathbf{S_n} \leftarrow \mathbf{S_n} \cup \{s_m\}$

Next m

$s \leftarrow \min[\mathbf{S_n}]$

$a_j \leftarrow t + \dfrac{d}{s}$

End While

Output: a_j

Departure time algorithm $\boldsymbol{H_{yb}}$

Input

$\mathbf{R_n^p}$, $\mathbf{T_p}$, $\mathbf{v_n^p}$ $a_j, e_i, \ g_i, v_i \ ,v_j, u_{ij}$

START

$D_{ij} \leftarrow f(x_i, x_j, y_i, y_j)$

Find $k, t_{\underline{k}} \le a_j \le t_{\bar{k}}$

$\quad k \leftarrow k$

$\mathbf{S_n} \leftarrow \emptyset$

For $m = 1 \in$ □ **To** n

$\quad D_{jm} \leftarrow f(x_j, x_m, y_j, y_m)$

$\quad D_{im} \leftarrow f(x_i, x_m, y_i, y_m)$

$\quad r \leftarrow \sqrt{\left(\frac{d}{d_{ij}} D_{ij}\right)^2 + D_{im}^2 - \frac{d}{d_{ij}}\left(D_{ij}^2 + D_{im}^2 - D_{jm}^2\right)}$

$\quad$ **If** $r \le R_{km}$ **Then**

$\quad\quad s_m \leftarrow u_{ij} \dfrac{v_{km}}{w_m}$

$\quad$ **Else**

$\quad\quad s_m \leftarrow u_{ij}$

$\quad$ **End If**

$\quad \mathbf{S_n} \leftarrow \mathbf{S_n} \cup \{s_m\}$

Next m

$s \leftarrow \min[\mathbf{S_n}]$

$d \leftarrow d_{ij}$

$t \leftarrow a_j$

$d_k \leftarrow (t - t_{\underline{k}})s$

While $d_k < d$ **And** $k > 0$ **And** $t_{\bar{k}} > e_i$ **Do**

$\quad d \leftarrow d - d_k$

$\quad \mathbf{S_n} \leftarrow \emptyset$

$\quad$ **For** $m \in \mathbb{N} = 1$ **To** n

$\quad\quad D_{jm} \leftarrow f(x_j, x_m, y_j, y_m)$

$\quad\quad D_{im} \leftarrow f(x_i, x_m, y_i, y_m)$

$r \leftarrow \sqrt{\left(\frac{d}{d_{ij}} D_{ij}\right)^2 + D_{jm}^2 - \frac{d}{d_{ij}}\left(D_{ij}^2 + D_{jm}^2 - D_{im}^2\right)}$

If $r \leq R_{km}$ **Then**

$s_m \leftarrow u_{ij} \dfrac{v_{km}}{w_m}$

Else

$s_m \leftarrow u_{ij}$

End If

$\mathbf{S_n} \leftarrow \mathbf{S_n} \cup \{s_m\}$

Next m

$s \leftarrow \min[\mathbf{S_n}]$

$t \leftarrow t_{\underline{k}}$

$b_i \leftarrow t - \frac{d}{s}$

$k \leftarrow k - 1$

$d_k \leftarrow (t - t_{\underline{k}})s$

End While

If $d_k \geq d$ **Then**

$b_i \leftarrow t_{\bar{k}} - \frac{d}{s}$

Else

$b_i \leftarrow -\infty$

Output: b_i

Derivation of bottleneck distance

The following is the derivation of the bottleneck distance function r for the forward-iterative calculation in the AT algorithm. An identical argument with the distance d iterated in the backward direction from a customer j to i obtains the bottleneck distance function for the DT algorithm in a trivial manner.

Let θ_{im} be the angle opposite D_{im}, the Euclidean distance from customer i to bottleneck B_m. Using the law of cosines:

$$D_{im}^2 = D_{ij}^2 + D_{jm}^2 - 2D_{ij}D_{jm}cos(\theta_{im}).$$

$$cos(\theta_{im}) = \frac{D_{ij}^2 + D_{jm}^2 - D_{im}^2}{2D_{ij}D_{jm}} \ . \tag{6}$$

θ_{im} is also the angle opposite to r; equating r and equation (6) and using the law of cosines again:

$$r_m^2 = \left(\frac{d}{d_{ij}}D_{ij}\right)^2 + D_{jm}^2 - 2\left(\frac{d}{d_{ij}}D_{ij}\right)D_{jm}cos(\theta_{im})$$

$$= \left(\frac{d}{d_{ij}}D_{ij}\right)^2 + D_{jm}^2 - 2\left(\frac{d}{d_{ij}}D_{ij}\right)D_{jm}\left(\frac{D_{ij}^2 + D_{jm}^2 - D_{im}^2}{2D_{ij}D_{jm}}\right)$$

$$= \left(\frac{d}{d_{ij}}D_{ij}\right)^2 + D_{jm}^2 - \frac{d}{d_{ij}}\left(D_{ij}^2 + D_{jm}^2 - D_{im}^2\right)$$

$$\Rightarrow r_m = \sqrt{\left(\frac{d}{d_{ij}}D_{ij}\right)^2 + D_{jm}^2 - \frac{d}{d_{ij}}\left(D_{ij}^2 + D_{jm}^2 - D_{im}^2\right)}$$

REFERENCES

1. Weisbrod, G., V. Donald, and G. Treyz, Economic Implications of Congestion. NCHRP Report #463. 2001, National Cooperative Highway Research Program, Transportation Research Board: Washington, DC.
2. Hensher, D. and S. Puckett, *Freight Distribution in Urban Areas: The role of supply chain alliances in addressing the challenge of traffic congestion for city logistics.* Working Paper ITS-WP-04-15, 2004.
3. Hensher, D. and S. Puckett, *Refocusing the Modelling of Freight Distribution: Development of an Economic-Based Framework to Evaluate Supply Chain Behaviour in Response to Congestion Charging.* Transportation, 2005. **32**(6): p. 573-602.
4. Golob, T.F. and A.C. Regan, *Impacts of highway congestion on freight operations: perceptions of trucking industry managers.* Transportation Research Part A: Policy and Practice, 2001. **35**(7): p. 577-599.
5. ERDG, *The Cost of Congestion to the Economy of the Portland Region, Economic Research Development Group, December 2005*. 2005: Boston, MA, accessed June 2008, http://www.portofportland.com/Trade_Trans_Studies.aspx.
6. ERDG, *The Cost of Highway Limitations and Traffic Delay to Oregon's Economy, Economic Research Development Group, March 2007*. 2007: Boston, MA, accessed October 2008, http://www.portofportland.com/Trade_Trans_Studies_CostHwy_Lmtns.pdf.
7. Golob, T.F. and A.C. Regan, *Traffic congestion and trucking managers' use of automated routing and scheduling.* Transportation Research Part E: Logistics and Transportation Review, 2003. **39**(1): p. 61-78.
8. Golob, T.F. and A.C. Regan, *Trucking industry preferences for traveler information for drivers using wireless Internet-enabled devices.* Transportation Research Part C: Emerging Technologies, 2005. **13**(3): p. 235-250.
9. Figliozzi, M.A., L. Kingdon, and A. Wilkitzki, *Analysis of Freight Tours in a Congested Urban Area Using Disaggregated Data: Characteristics and Data Collection Challenges.* Proceedings 2nd Annual National Urban Freight Conference, Long Beach, CA. December, 2007.
10. Holguin-Veras, J., et al., *The impacts of time of day pricing on the behavior of freight carriers in a congested urban area: Implications to road pricing.* Transportation Research Part A-Policy And Practice, 2006. **40**(9): p. 744-766.
11. Quak, H. and M. de Koster, *Delivering Goods in Urban Areas: How to Deal with Urban Policy Restrictions and the Environment.* Transportation Science, 2009. **43**(2): p. 211-227.
12. Figliozzi, M.A., *Analysis of the efficiency of urban commercial vehicle tours: Data collection, methodology, and policy implications.* Transportation Research Part B, 2007. **41**(9): p. 1014-1032.
13. Daganzo, C.F., *Logistics Systems-Analysis*, in *Lecture Notes In Economics And Mathematical Systems*. 1991.
14. Figliozzi, M.A., *The impacts of congestion on commercial vehicle tour characteristics and costs.* Transportation Research Part E, 2009.
15. Figliozzi, M.A. *A Route Improvement Algorithm for the Vehicle Routing Problem with Time Dependent Travel Times*. in *Proceeding of the 88th Transportation Research Board Annual Meeting CD rom*. 2009. Washington, DC, January 2009, also available at http://web.cecs.pdx.edu/~maf/publications.html.
16. Fleischmann, B., M. Gietz, and S. Gnutzmann, *Time-varying travel times in vehicle routing.* Transportation Science, 2004. **38**(2): p. 160-173.

17. Eglese, R., W. Maden, and A. Slater, *Road Timetable (TM) to aid vehicle routing and scheduling.* Computers & Operations Research, 2006. **33**(12): p. 3508-3519.
18. *Google Maps API*. 2009 [cited 2009 July 30]; Available from: http://code.google.com/apis/maps/.
19. Bertini, R.L., et al., *PORTAL: Experience Implementing the ITS Archived Data User Service in Portland, Oregon.* Transportation Research Record 1917, 2005: p. 90-99.
20. Cassidy, M.J., C.F. Daganzo, and K. Jang, *Spatiotemporal Effects of Segregating Different Vehicle Classes on Separate Lanes.* UC Berkeley Center for Future Urban Transport: A Volvo Center of Excellence, 2008.
21. Figliozzi, M.A., *Planning Approximations to the Average Length of Vehicle Routing Problems with Varying Customer Demands and Routing Constraints.* Transportation Research Record 2089, 2008: p. 1-8.

LIST OF FIGURES

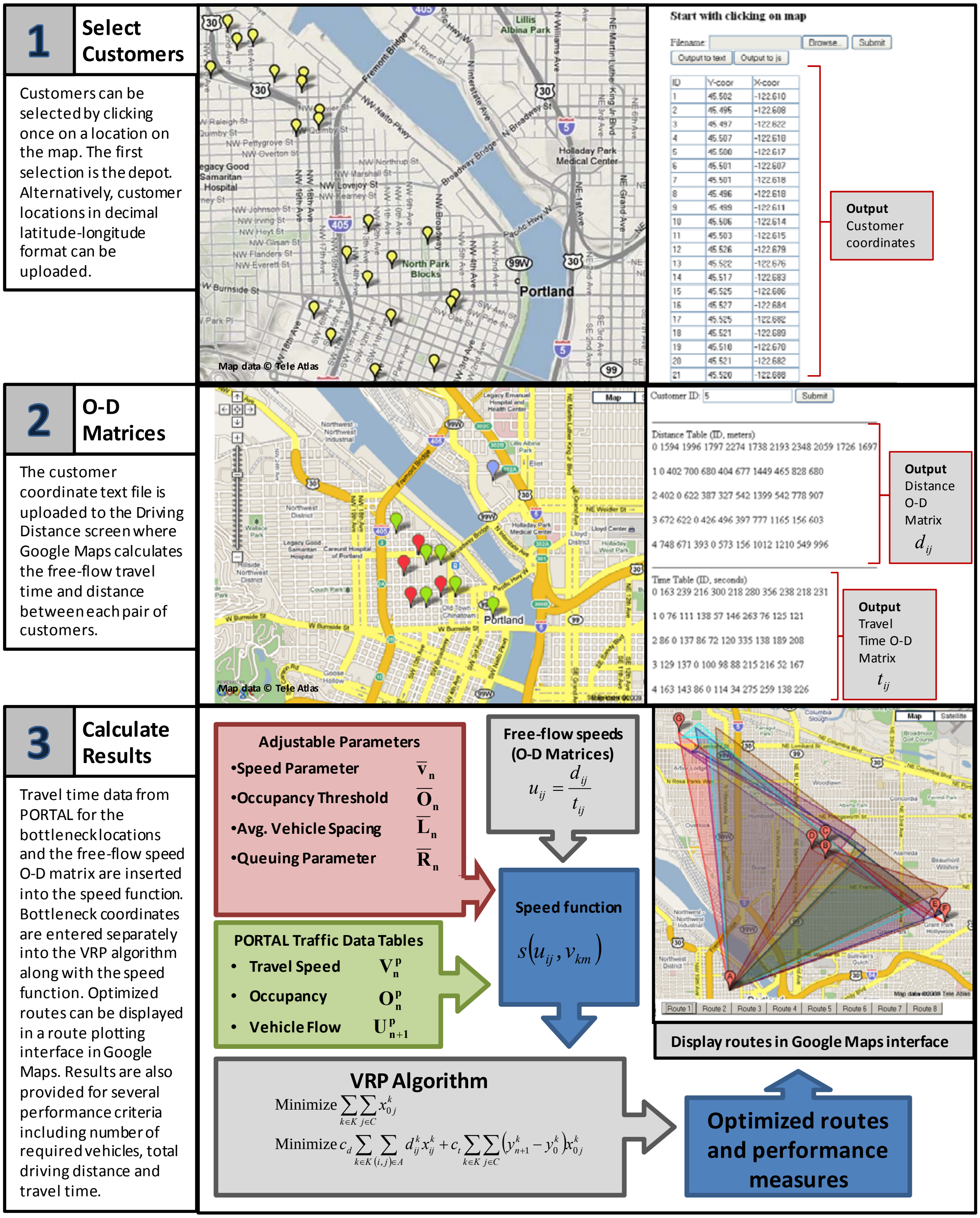

FIGURE 1: Overview of the TDVRP solution methodology and integration of the Google Maps API

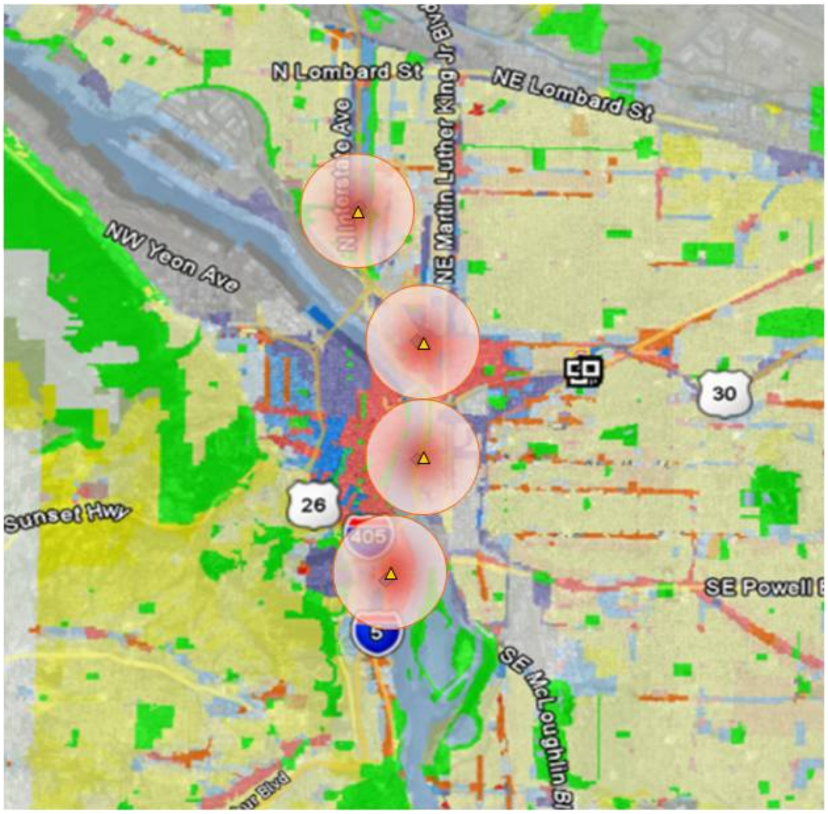

FIGURE 2: Example with bottleneck locations and areas of effective travel speed reduction

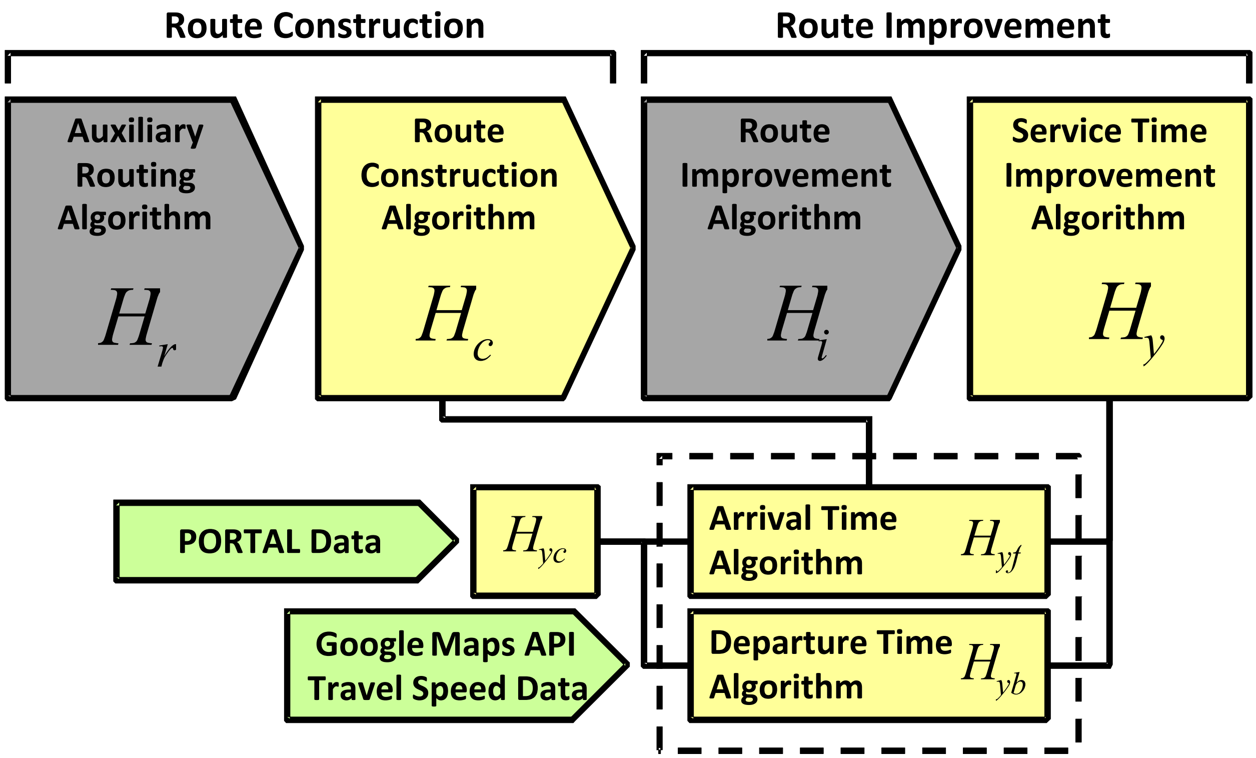

FIGURE 3: TDVRP Solution method

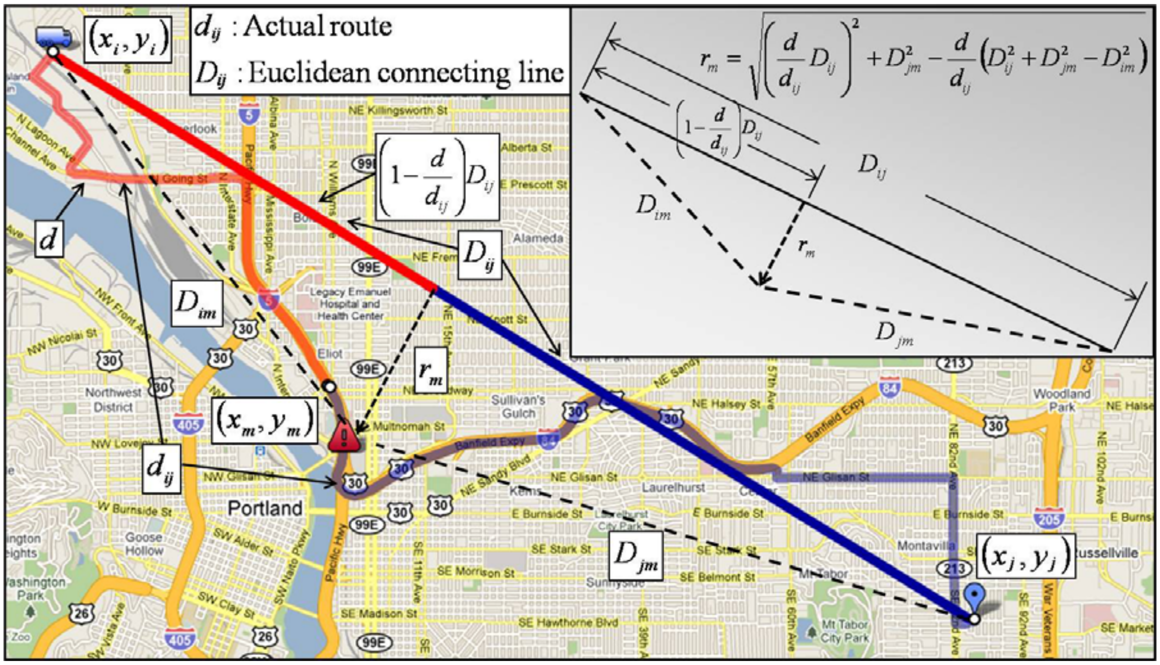

FIGURE 4: Illustration of the method to approximate bottleneck influence

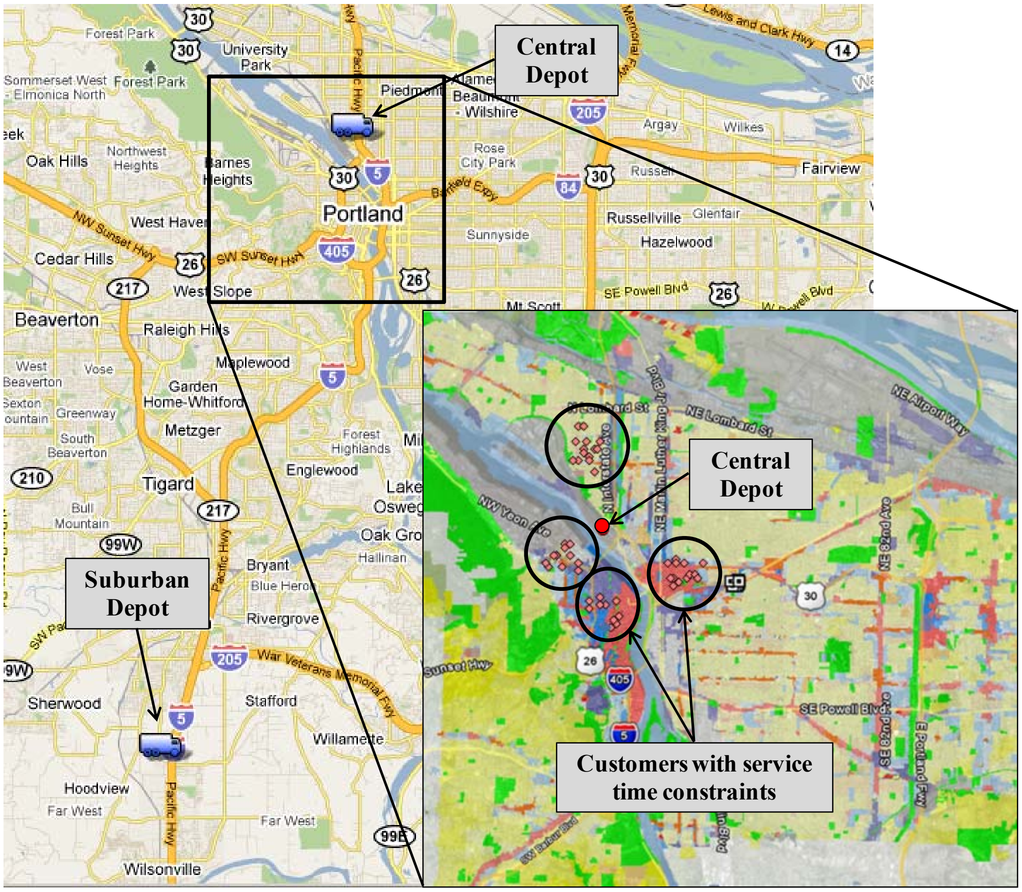

FIGURE 5: Customer service area and depot locations

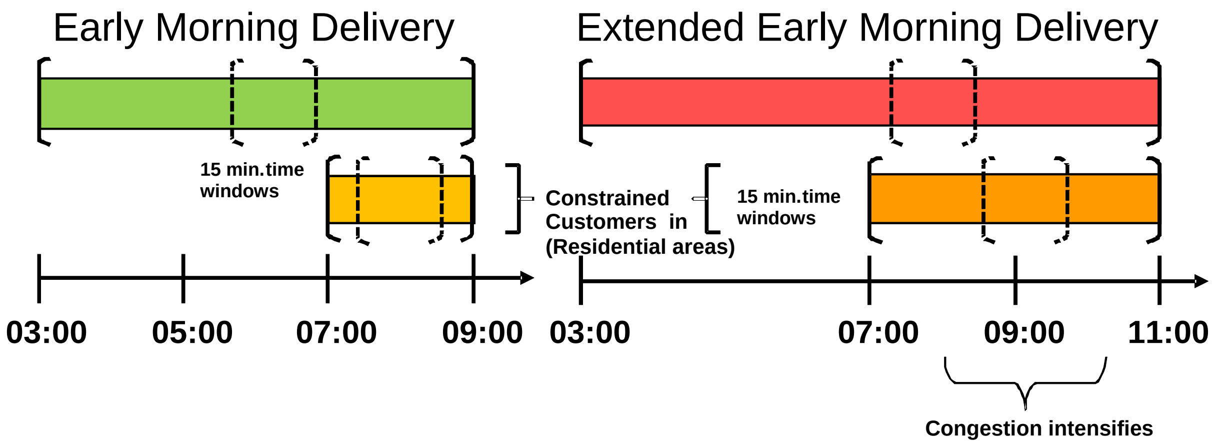

FIGURE 6: Modeled delivery periods, constrained customers, and time window constraints

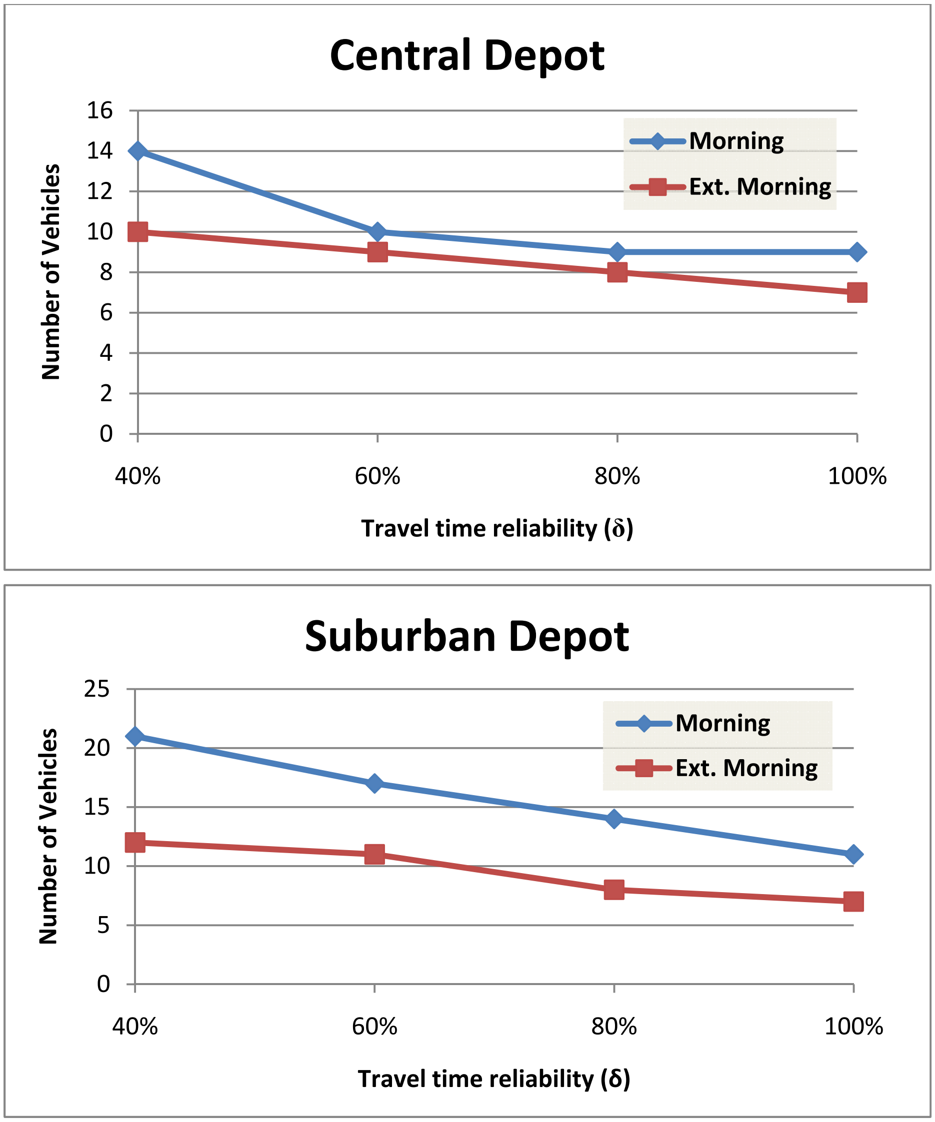

FIGURE 7: Effects of congestion on fleet size

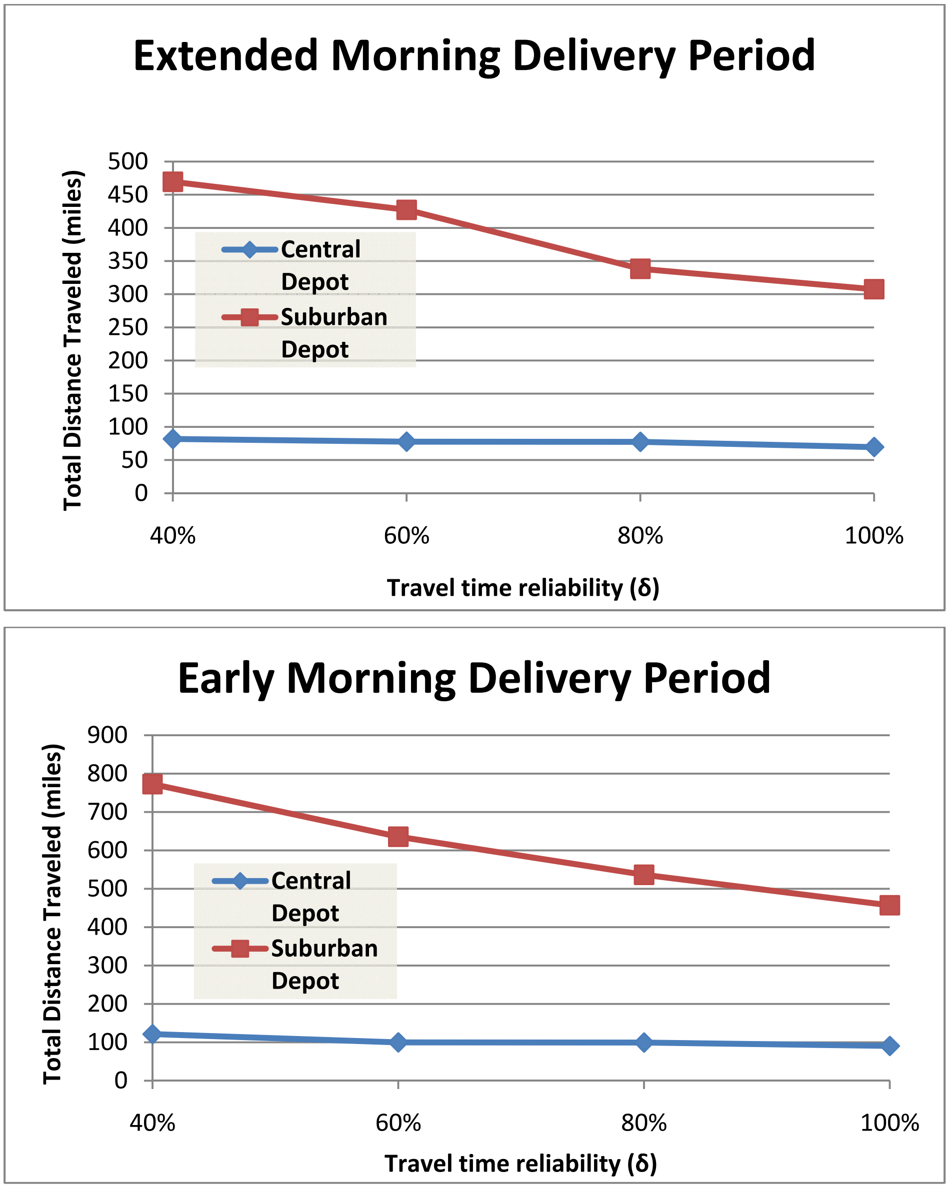

FIGURE 8: Effects of congestion on total VMT