Network Maintenance and Capacity Management with Applications in Transportation

by

Dening Peng

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree of
Doctor of Philosophy

Approved November 2017 by the
Graduate Supervisory Committee:

Pitu B. Mirchandani, Chair
Jorge Sefair
Teresa Wu
Xuesong Zhou

ARIZONA STATE UNIVERSITY

December 2017

ABSTRACT

This research develops heuristics to manage both mandatory and optional network capacity reductions to better serve the network flows. The main application discussed relates to transportation networks, and flow cost relates to travel cost of users of the network. Temporary mandatory capacity reductions are required by maintenance activities. The objective of managing maintenance activities and the attendant temporary network capacity reductions is to schedule the required segment closures so that all maintenance work can be completed on time, and the total flow cost over the maintenance period is minimized for different types of flows. The goal of optional network capacity reduction is to selectively reduce the capacity of some links to improve the overall efficiency of user-optimized flows, where each traveler takes the route that minimizes the traveler's trip cost. In this dissertation, both managing mandatory and optional network capacity reductions are addressed with the consideration of network-wide flow diversions due to changed link capacities.

This research first investigates the maintenance scheduling in transportation networks with service vehicles (e.g., truck fleets and passenger transport fleets), where these vehicles are assumed to take the system-optimized routes that minimize the total travel cost of the fleet. This problem is solved with the *randomized fixed-and-optimize* heuristic developed. This research also investigates the maintenance scheduling in networks with multi-modal traffic that consists of (1) regular human-driven cars with user-optimized routing and (2) self-driving vehicles with system-optimized routing. An iterative mixed flow assignment algorithm is developed to obtain the multi-modal traffic assignment resulting from a maintenance schedule. The genetic algorithm with multi-point crossover is applied to obtain a good schedule.

Based on the Braess' paradox that removing some links may alleviate the congestion of user-optimized flows, this research generalizes the Braess' paradox to reduce the capacity of selected links to improve the efficiency of the resultant user-optimized flows. A heuristic is developed to identify links to reduce capacity, and the corresponding capacity reduction amounts, to get more efficient total flows. Experiments on real networks demonstrate the generalized Braess' paradox exists in reality, and the heuristic developed solves real-world test cases even when commercial solvers fail.

ACKNOWLEDGEMENTS

I am deeply grateful for the guidance and mentorship provided by my committee chair, Professor Pitu Mirchandani. From him I have learned much about optimization, transportation science, and how to communicate effectively. I thank him for the collaboration, securing research funding for me, and spending so much of his valuable time working with me. His concern for my well-being was apparent all the way through, for which I am especially grateful.

I am appreciative of the funding from the Institute of Safety and Operations of Large-Area Rural/Urban Intermodal Systems (SOLARIS), and the National Science Foundation (NSF) for partially supporting this research. The research does not necessarily reflect the viewpoints of these funding organizations. Additional fellowships and travel support came from the department, college, and the Graduate Professional and Student Association (GPSA) at Arizona State University, for which I am very grateful.

I am also thankful to my graduate supervisory committee that includes Professor Jorge Sefair, Professor Teresa Wu and Professor Xuesong Zhou for their time and advice.

Finally, I am very grateful to my family for always supporting me, encouraging me, and believing in me. I thank my sister for encouraging me to study abroad and pursue higher education. My parents deserve the utmost appreciation. Their constant and unwavering support helped me overcome all the difficulties I had in my life so far. They have blessed me in more ways than I can tell. The best thing that has ever happened to me is my family.

TABLE OF CONTENTS

CHAPTER Page

APPENDIX Page

LIST OF TABLES

TABLE Page

TABLE Page

LIST OF FIGURES

FIGURE Page

Chapter 1

INTRODUCTION

1.1 Overview

A network is a collection of connected nodes and arcs, which are used to store, distribute and convey various kinds of entities. These nodes, arcs and entities represent disparate things in various applications. For example, in power transmission networks, nodes are power plants, substations, households and factories; arcs are power lines; and entity transmitted is power. In transportation networks, nodes are origins and destinations, arcs are the roads, and entities transported can be vehicles, people, commodities etc. Although the flow of entities in different networks obey different physical rules, normally the basic demand-supply relation among nodes, the flow conservation conditions and the capacity constraints on nodes and/or arcs are common.

Network maintenance is the activity conducted on nodes and/or arcs to restore or improve flow-related attributes like capacity, surface roughness (in transportation networks), outage duration (in power transmission networks), etc. so as to elevate the overall network performance. Just like decision problems of other large systems, the planning of network infrastructure maintenance can be categorized as strategic, tactical or operational.

Strategic planning of network maintenance mostly focuses on network-wide design to maintain the overall performance of the network over the long term. At this high level of planning, the impact of network capacity reduction caused by maintenance activity is negligible, because the maintenance activity usually takes place over a very small portion of the planning horizon.

Tactical planning of network maintenance usually is the medium-term scheduling of maintenance work on the nodes and/or arcs with a network-wide perspective. Since the length of time period when the network is under maintenance is comparable to the tactical planning time horizon, network capacity reduction caused by maintenance work is an important factor to consider for maintenance scheduling.

As to operational planning of network maintenance, it considers short-term repair effects on a node and/or an arc when a network component is under repair during the maintenance operations. At this level of maintenance planning, the dynamics and specific maintenance procedures have substantial impact on the network entities. For example, barriers, traffic cones and heavy vehicles (i.e., pavers) will occupy a segment of road in transportation network for resurfacing work. Plans on the length of the sub-segments for the resurface work and the time to start each sub-segment directly impact the traffic flow during the resurfacing.

This dissertation specifically investigates the network maintenance planning for arcs at the tactical level, where the arc capacity reduction caused by maintenance activity is considered. Since scheduling arc repairs is essentially scheduling the arc capacity reductions, the tactical planning of network maintenance is a network capacity management problem, which manages mandatory network capacity changes to optimally fulfill flow demand. The type of network considered in this research are transportation networks which have straightforward flow diversions in reaction to arc capacity reductions. The optimal scheduling of work zones for arc maintenance is one problem addressed in this research. Another type of network capacity management problem studied is to selectively reduce the capacity of some arcs so as to reach better user equilibrium (Wardrop, 1952) states. This type of network capacity management problem is also studied in the research presented.

In transportation literature, term “link” is used more frequently to represent the actual road segments, while in classic mathematics literature on networks, term “arc” is used for the connection between nodes. In this proposal, terms arc(s) and link(s) are used interchangeably.

1.2 Background and Research Focus

Network maintenance planning can be formulated as multi-objective network design problems, with complex constraints based on the spatial and temporal scope of the maintenance planning. Despite the various factors, such as link/node downtime, congestion, and budget, that need to be considered in these problems, the ultimate goal of network maintenance is to improve the overall capability of the network so as to better serve the flows from the origins (O) to their destinations (D). Hence, the major concern in the research conducted is the performance of the network on fulfilling the flow demand during the maintenance, which can be translated into minimizing the temporal or monetary costs (such as total flow cost, total travel time, total time delay), by scheduling the network capacity changes during the maintenance period.

Maintenance work on the network can cause network topology changes (e.g., link capacity change, closed link, and/or disconnected node). For a feasible schedule of the maintenance projects within the planning time horizon, the network topology changes every time the status of an individual maintenance project is changed (for example, maintenance of a lane segment is started or completed). And each time when the network topology changes, the routing of the flows change accordingly so as to minimize the individual/total flow cost. Hence, there is a total flow cost over the planning time horizon associated with each feasible schedule. In summary, maintenance work zones interact with flows; the optimal scheduling of the maintenance work zones means deciding the optimal sequence to carry out the projects, so that the network topology change patterns achieve the minimum total flow cost over the planning horizon, among all the feasible schedules.

The primary objectives of the dissertation are (1) to develop optimization models that schedule network maintenance and manage network capacity changes considering the interaction between maintenance work and the flows, and (2) to design efficient solution approaches to solve them. Different network flows models will result in different maintenance schedules that are optimal to the specific network flows model. To give an example, the optimal maintenance schedule for a network with multi-commodity flows that take system optimized routing to minimize the total cost of all flows, will most likely be different from the optimal schedule for the same network but with flows that take user optimized routing to reach user equilibrium (Wardrop, 1952). Hence, this research studies network maintenance schedule for different types of network flows models. Also, it is possible that flows with different routing objectives share the same network. This results in not only the interaction between the flows and the maintenance schedule but also the interaction among flows of different types. And thus, the investigation of scheduling maintenance in networks with various flow types also falls into the scope of this research.

This research uses terminology "directed links" to represent roads, each of which consists one or more lanes. An incident on a link segment blocks one or more lanes, thereby decreasing the flow capacity for some lanes and thus of the link segment. Congestion effects of incidents is well researched (Chung, 2011; Corthout et al., 2009; Jeong et al., 2010; Lund and Pack, 2010; Sheu et al., 2004 and 2001), one focus of minimizing these effects is to detect the incident as quickly as possible (Baiocchi et al., 2015; Cheng et al., 2015; Kinoshita et al., 2015; Li et al., 2013; Liu et al., 2014; Lu et al., 2012a and 2012b; Wang et al., 2015; Xiao et al, 2014; Xiao et al., 2012; Zhang et al., 2015;

and Zheng et al., 2013), and subsequently send response vehicles as fast as possible to clear the incident (Hou et al., 2013; Huang and Pan, 2007; Kim et al., 2014; Lei et al., 2015; Lou et al., 2011; Ma et al., 2014a; Ma et al. 2014b; Pal and Bose, 2009; Zhu et al., 2012; and Zografos et al., 2002) and/or to quickly apply traffic controls like traffic signal phase adjustments, ramp meters activation, and traffic barricades to manage the congestion (Ahmed and Hawas, 2015; Gang and Yong, 2011; Liu et al., 2013; Long et al., 2012; Lu et al., 2015; Shen et al., 2007; Sheu, 2007; Sheu et al., 2003; and Zhang et al., 2011). Well-planned and scheduled maintenance could minimize the congestion impacts of maintenance activities even without the help of additional traffic controls.

The impairment of roads, the installation of new traffic management infrastructures (e.g., high occupancy vehicle lanes, tolled lanes, and ramp meters), and adding/improving links require the scheduling of the corresponding maintenance work. In general, maintenance activities change the topology of the transportation network and change the cost of the routes for origin-destination (OD) demands. Since traffic flows are composed of individual vehicles that make their own routing decisions, and with the extensive usage of navigation systems with real-time traffic information, OD demands are able to reactively re-route based on the changed network topology and the resultant cost of candidate routes. Traffic flows consist of different types of network users (i.e., commercial trucks, commuter cars, and motorcycles). These users, besides interacting with each other, react to network topology changes differently because of their distinct routing objectives and flow cost attributes. This makes the transportation network an ideal real-world application for methodology developed on the maintenance scheduling of flow networks.

Maintenance activities of transportation networks result in *work zones*, where some lane segments of links are out of commission for a predicted period of time until the work is completed. The extent of the congestion impacts of a work zone, induced by the traffic that normally uses the lanes affected by the work zone, depend on the volume and mix of traffic. When a lane is blocked in a link segment, the "capacity", in terms of vehicles per hour, of the link deceases for the duration of the work zone. If the volume of traffic using the work zone is very small, especially if there are many alternatives of equally good routes, then the congestion impacts are negligible. On the other hand, if the traffic volume is moderate to high then congestion impacts would not be negligible. Temporary link capacity reductions because of lane closures can result in significant delays for commuters and transport service vehicles. FHWA (2013) estimate that Americans lose 3.7 billion hours and 2.3 billion gallons of fuel every year sitting in traffic jams. Work zones are estimated to cause about 10% of overall congestion which translates into annual fuel loss of over 700 million US dollars.

The large majority of traffic using a road network consists of (1) commuter traffic, and (2) the traffic of service vehicles that includes trucks and vans delivering goods. The primary effect of a work zone on commuter traffic is a change in traffic equilibrium of the flows, because in a few days after the start of the work zone the traffic flows will equilibrate to a new user equilibrium according to the well-known Wardrop's first principle (Wardrop, 1952). So one main idea of this research is to optimally schedule the planned work zones so that the resulting traffic delays for commuter traffic is minimized. When the network is normally not congested, the commuter traffic equilibrium would change little. But work zones could have significant impacts on the equilibrium pattern if the network is normally congested. On the other hand, traffic of service vehicles will be affected when a link that is used by many shortest delivery routes is impacted by the work zone.

It should be noted that for the work zone operations in practice, road construction companies and transportation management agencies do a reasonable job of coordinating work zone activities after the work zone is initiated through appropriate task scheduling and work staging of day-to-day and week-to-week operations. These companies'/agencies' goal is to contain the overall cost, while safety and traffic congestion is not overly affected during peak periods. In current practice, the state departments of transportation have work zone standards for single maintenance projects on state/local roads. These standards provide detailed guidelines and requirements for contractors to prepare bids, obtain the contract for the maintenance project, and conduct the maintenance work. However, the requirement on traffic control is often very vague. For example, the requirement document on traffic control for New Hampshire focuses more on the traffic safety and traffic control installations, and only briefly discusses about minimizing traffic interruption by avoiding maintenance work during peak hours, and by avoiding frequent and abrupt road capacity changes (e.g., lane narrowing, dropped lanes, lane shifting). (New Hampshire Department of Transportation, 2012). Also it does not discuss about the impact of work zones on the traffic in the neighborhoods, which may not be negligible since the temporary link capacity reduction caused by the work zones on the link being repaired will probably cause some traffic that was originally on the link to divert to other links.

In practice, for a single maintenance project along a highway stretch or a local arterial, the typical project cycle starts with the advertisements by a transportation agency. Contractors interested in the project prepare bid documents and submit the bids to the transportation agency to compete for the project. The agency evaluates the bids received on various criteria, especially on the proposed budget, and awards the contract to the contractor with the most competitive qualifying bid. The winning contractor then works on the maintenance project. In summary, the standards and work scope are only concerned with a single maintenance project on a highway stretch or a local arterial.

Consideration of coordinating multiple maintenance projects that may be located close to each other, is often ad-hoc.

Most past research conducted on maintenance scheduling in transportation networks fall into either the strategic planning of long-term network rehabilitation, or operational level of planning that decides the work zone length and short-term scheduling of activities for a single maintenance project. Little research has been done on the tactical level of planning that coordinates maintenance projects based on a network-level perspective and that considers the impact of maintenance work on traffic flows at the same time. More details on related past research are covered in the literature review in Chapter 2. While a single, or few widely scattered concurrent work zones, will not have a large effect upon daily traffic patterns, several work zones that are spatially and temporally close together, and which affect large flows of traffic, may result in traffic patterns that are both costly to commuters and vehicle-based services.

The maintenance of the transportation network is not the only cause for work zones. Work related to infrastructures (e.g., power transmission cables, street/highway lights, sewage pipes, communication cables/fibers) that are close by or under the roads may also result in work zones. The more the work zones that are spatially close to each other and with partially or entirely overlapping planning time horizon, the more critical it is to coordinate the active periods among the projects. A reduction of negative impacts can be expected through proper scheduling of work zones with respect to the spatial locations in the network and the time periods of the work zones.

Depending on the underlying network flows model adopted, the improvement of network capability to better serve the flow demand does not always mean to increase the capacity of some road segments. As stated in the well-known Braess' paradox (Braess et al., 2005), with the adoption of the user equilibrium (UE) flows model (Wardrop, 1952), where each unit of flow finds its own optimized route, and all the flows eventually reach the equilibrium where no flow cost reduction can be achieved through unilateral route change, so that increasing the capacity on part of the network may cause a redistribution of the flows ending with higher total flow cost, while reducing the capacity might result in a flow redistribution that costs less. A real-world example of the Braess' Paradox is that the closing of 42nd Street in New York City in 1990 decreased the congestion in the area (Kolata, 1990). Therefore, network capacity management does not only include the scheduling of mandatory link capacity reductions that maybe required by maintenance activities, but also encompasses the development of optional link capacity reduction mechanisms to improve the efficiency of traffic flows. In traffic networks, this selective link capacity reduction can be achieved by traffic control methods like variable speed limits, ramp metering, and coordinated traffic light phasing.

Thus, this dissertation addresses the network capacity management problem for the following three cases: (1) scheduling mandatory network capacity changes to minimize the total flow cost of service vehicles (e.g., delivery trucks) from multiple origins to destinations in the case of uncongested networks, (2) designing optional network capacity changes to reduce the total travel time of commuter vehicle flows at equilibrium, and (3) scheduling mandatory network capacity changes to minimize the total travel time for multi-modal traffic flows. The maintenance scheduling and capacity management in transportation networks is just one of the many areas where apply the methodological results of this research maybe applied. With few changes reflecting network dynamics and maintenance activity characteristics, the optimization models formulated can be adopted to the modeling of maintenance scheduling and capacity management of other types of networks.

1.3 Summary of Chapters

Chapter 2 starts with the review of network flows problems, whose optimization models and solution methods can be integrated into the network capacity management problem studied in this dissertation. Maintenance scheduling models for networks other than the transportation network (e.g., power transmission networks, water pipe networks, bridge networks, and railroad networks) are also reviewed, so as to obtain the general understanding on how systematic maintenance planning is approached for different types of flow networks. This is followed by a detailed review on maintenance planning specifically for transportation networks. At the end of Chapter 2, three types of network flows management approaches are reviewed, which include ramp metering, toll imposition, and variable speed limit enforcement. These traffic management mechanisms can be employed to maintain and improve network performance when network flows are characterized by traffic equilibrium models.

Chapter 3 investigates the maintenance scheduling in networks of service vehicles (MS-NSV). In Chapter 3, it is assumed that if there are too many trucks traveling on a link, there will be a qualitative change of the relation between the link travel cost and the number of trucks traveling on the link. This change is captured by modeling the link travel cost as a piece-wise linear function of the number of trucks using the link. The problem studied is formulated as a mixed-integer linear program, and is solved by a randomized fix-and-optimize heuristic (RFO) developed. In contrast to solving the problem solely with a commercial solver (e.g., CPLEX), test results demonstrate a significant reduction in computation times when RFO is applied.

Chapter 4 designs the mechanism that improves the efficiency of commuter traffic in network level by selectively reducing the capacity of some links (OCREC). Since commuter traffic are UE flows, OCREC studies the generalized Braess' paradox where reducing the capacity of some links could improve the efficiency of UE flows. A heuristic is developed to identify the links whose capacity reduction may decrease the total travel time at UE, and find the desired amount of capacity reduction for the links identified. The heuristic developed successfully solves real network test cases and confirms that the generalized Braess' paradox does exist in reality. As a comparison, nonlinear commercial solvers (e.g., MINOS) fail to solve test cases of moderate size.

Chapter 5 extends the research in Chapter 3 to study maintenance scheduling in networks with multi-modal traffic flows (MS-MMN). Two travel modes are considered in MS-MMN and they are regular cars and autonomous vehicles. Every traveler driving a regular car takes the route that minimizes his/her own travel time to reach user equilibrium (UE), and travelers riding self-driving vehicles choose the route that minimizes the total travel time of all travelers to achieve system optimum (SO). The stationary flow assignment of this multi-modal traffic is the flow assignment that has regular car flows at UE and self-driving vehicle flows at SO. This stationary flow assignment is proven to exist and it can be obtained by the iterative UE-SO assignment algorithm developed. Due to the non-convexity of MS-MMN, the genetic algorithm is applied to obtain good maintenance schedules.

Chapter 6 summarizes the research conducted and outlines research opportunities for future work, which include various stochastic extensions to the problems studied in Chapter 3, 4 and 5.

Chapter 2

LITERATURE REVIEW

The network maintenance planning has been studied with two major modeling approaches: network reliability modeling and network flows modeling. In research that adopt network reliability modeling approach, the deterioration process of links/nodes is modeled and the objective is to minimize the overall link/node failures (e.g., Bocchini and Frangopol, 2011; Hu et al., 2015; Marquez et al., 2013). The network flows modeling approach aims at managing the network capacity changes to better fulfil flow demands. This modeling approach uses network flows models (e.g., maximum flows model) to evaluate the networks for a specific maintenance schedule, so as to evaluate their optimality (e.g., Boland et al., 2012; Boland et al., 2015; Tawarmalani and Li, 2011). There also exists research that combines these two modeling approaches by associating the deterioration process with the amount of flows on the link (e.g., Hajibabai et al., 2014), or by modeling the link capacity as a function of the link states in the deterioration process (e.g., Chu and Chen, 2012).

Although research on network maintenance planning with the network reliability modeling approach is covered in the review, it is more focused on previous research that adopted the network flows modeling approach, since the research presented emphasizes the interaction between flows and network capacity changes caused either by maintenance activities or by traffic controls. And thus, the literature review starts with the review of several basic network flow models in Section 2.1, which can be used as the part of the optimization models developed that evaluates the optimality of a maintenance schedule or a traffic control mechanism. Section 2.2 reviews maintenance planning in general networks that can be the abstract of any virtual or physical networks. Research that specifically studies transportation related networks (e.g., traffic networks, logistics distribution networks, and bridge networks) is reviewed in section 2.3. Section 2.4 reviews traffic control mechanisms that selectively reduces the capacity or increase the cost of some links to alleviate congestion and drive traffic flows toward more efficient flow patterns network-wide.

2.1 Related Network Flows Models

Based on the physical types and functions of the networks in application, various network flows models are used to evaluate the network capability of fulfilling flow demand. For example, maximum flow model and traffic equilibrium model are two of the models integrated in studying the impact of maintenance work on flows with a network-wide perspective (Boland et al., 2012; Boland et al., 2015; Lee, 2009; and Zheng et al., 2014). Section 2.1.1 to 2.1.4 review these network flows models and briefly discuss their applications.

2.1.1 Maximum Flow Model. The maximum flow problem tries to send as much flow as possible between two special nodes, the source node s and the sink t, through a capacitated network without exceeding the capacity of any link (Ahuja et al., 1993). In a directed network with node set N and link set E, let u_{ij} be the capacity of link $(i,j) \in E$, the linear programming formulation of this problem is:

$$Maximize\ v \tag{2.1.1a}$$

$subject\ to\ (s.t.)$:

$$\sum_{\{j:(i,j)\in E\}} x_{ij} - \sum_{\{j:(j,i)\in E\}} x_{ji} = \begin{cases} v & i = s \\ 0 & \forall i \in N - \{s,t\} \\ -v & i = t \end{cases} \tag{2.1.1b}$$

$$0 \le x_{ij} \le u_{ij} \qquad \forall (i,j) \in E \tag{2.1.1c}$$

Constraint (2.1.1b) is the flow conservation constraints enforcing all nodes other than the source node and sink node to send out the same amount of flows as they receive, and the sink node to receive the amount of flows sent out by the source node. (2.1.1c) is the set of link capacity constraints that ensure the amount of flow on each link not exceed its capacity. A vector $x = \{x_{ij}\}$ satisfying (2.1.1b) and (2.1.1c) is a feasible flow and the corresponding value of the scalar variable v is the value of the flow.

The maximum flow problem is an easy problem to solve since there exist algorithms that can solve it in polynomial time (e.g., shortest augmenting path algorithm, Dinic's algorithm, and generic preflow-push algorithm). It has been applied to the modeling of both physical networks to maximize the throughput, and virtual networks which are the abstracts of problems in other areas like assignment problems and scheduling problems. It is also a fundamental network flows model that occurs as a subproblem in the solution of more difficult network problems.

2.1.2 Minimum Cost Flow Model. The minimum cost flow problem finds the cheapest way of sending given amount of flow from a node (or a set of nodes) to another node (or another set of nodes) through a network, where each link has its capacity and unit flow cost. Let $G = (N, E)$ be a directed network with a positive cost c_{ij} and a capacity u_{ij} associated with every link $(i,j) \in E$. Each node $i \in N$ is associated with a number $b(i)$ which indicates its supply or demand depending on whether $b(i) > 0$ or $b(i) < 0$. If $b(i) > 0$, then node i is a supply node; and if $b(i) < 0$, then node i is a demand node. Variable x_{ij} is the amount of flow on link (i,j). With these parameters and variables, the minimum cost flows problem can be formulated as (Ahuja et al., 1993):

$$Minimize\ z(\boldsymbol{x}) = \sum_{(i,j)\in E} c_{ij}x_{ij} \tag{2.1.2a}$$

$s.t.$:

$$\sum_{\{j:(i,j)\in E\}} x_{ij} - \sum_{\{j:(j,i)\in E\}} x_{ji} = b(i) \quad \forall i \in N \tag{2.1.2b}$$

$$0 \le x_{ij} \le u_{ij} \qquad \forall (i,j) \in E \tag{2.1.2c}$$

Objective (2.1.2a) calculates the total cost of all the flows on all links. Constraint (2.1.2b) is the set of flow conservation constraints that make sure supply (demand) nodes send (receive) the exact amount it can supply (receive), and all the nodes other than the supply and demand nodes will send out the amount of flows the same as the amount they receive. Constraint (2.1.2c) is the capacity constraints limiting the amount of flow on each link to be less than or equal to the link's capacity.

Polynomial algorithms are also available to solve the minimum cost flow problem. As a category of problems that are pervasive in practice, minimum cost flow problems arise in almost all industries, including agriculture, communications, energy, manufacturing, medicine, retailing, transportation etc. It is also lays the foundation for more complex network flows problems like the multi-commodity flow problem.

2.1.3 Multi-Commodity Flow Model. In many application contexts, several types of entity flows share common network facilities and have their own origins and destinations. For example, in transportation networks vehicles from different origins travel to different destinations using the same transportation infrastructure. And each road has a capacity that restricts the total flow of all the vehicles using that road, regardless of their origins or destinations. To find an optimal flow in these cases, the problem needs to be solved in concert with all types of commodity flows (Ahuja et al., 1993). Thus arises the multi-commodity flow problem.

Let K be the number of commodity types, x_{ij}^k be the amount of flows of commodity k on link (i,j) , and $b^k(i)$ be the supply/demand of commodity k at node i.

With other notations used in Section 2.1.2, the node-link formulation of multi-commodity flow problem is shown below:

$$Minimize\ z(\boldsymbol{x}) = \sum_{(i,j)\in E} c_{ij}\left(\sum_{k\in K} x_{ij}^{k}\right) \tag{2.1.3a}$$

$$s.t.:$$

$$\sum_{\{j:(i,j)\in E\}} x_{ij}^{k} - \sum_{\{j:(j,i)\in E\}} x_{ji}^{k} = b^{k}(i) \qquad \forall i \in N, \forall k \in K \tag{2.1.3b}$$

$$0 \le \sum_{k\in K} x_{ij}^{k} \le u_{ij} \qquad \forall (i,j) \in E \tag{2.1.3c}$$

The formulation above is very similar to the minimum flow cost model in Section 2.1.2, except that the total flow of all commodities on link (i,j) are accounted in the objective (2.1.2a) and the link capacity constraint (2.1.1c), and the flow conservation constraints (2.1.2c) need to be defined for each commodity.

There is a wide variety of application contexts, such as vehicle fleet planning and production planning, which uses the multi-commodity flow problem. Since it is a strongly NP-hard problem (Even et al., 1975), there is no algorithm available that can solve it in polynomial time. But methods like Lagrangian Relaxation, column generation, and Dantzig-Wolfe decomposition can solve it within tolerable amount of time in some cases. In the multi-commodity flow problem discussed in this section, the unit flow cost of each link is a constant that is independent of the amount of flows on the link. In the cases where the link unit flow cost increases as the amount of flows that are using the link increase, the multi-commodity flow problem evolves to the traffic assignment problem.

2.1.4 Traffic Assignment Model. In the modeling of networks with traffic flows (e.g., road networks, fiber networks, and power transmission networks), the congestion effect is commonly considered. And that means the cost of using a link does not only depend on the capacity of the link, but also depends on the amount of flows using the link. The graph below illustrates the cost-flow relationship for a long link:

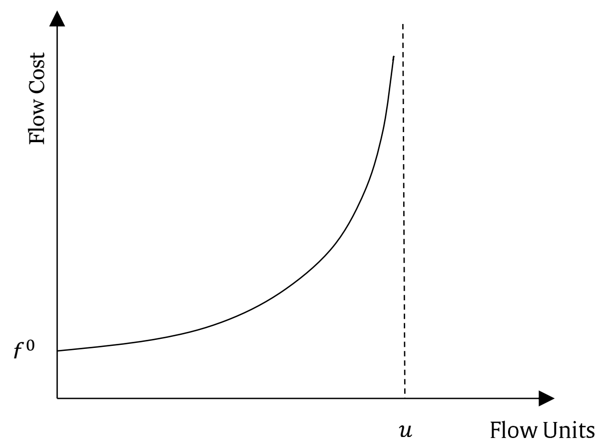


Figure2.1.4-i: Cost-Flow Relationship

The horizontal axis represents the amount of flows using the link, and the vertical axis is the corresponding unit flow cost. f^0 is the base cost for a unit of flow traveling through the link when the link is not used by other flow units, and u is the link capacity.

In the context of traffic flow in transportation networks, Wardrop (1952) postulated two general principles to determine the distribution of traffic flows on the routes between each origin-destination (OD) pair, and they are:

(1) Wardrop's First Principle: The travel time between an origin-destination (OD) pair is the same on all routes used, and it is less than those which would be experienced by a single vehicle on any unused route.

(2) Wardrop's Second Principle: The trips or movements are routed so that the sum of the travel time for all the movements is a minimum.

These two alternative principles are applied widely to the modeling of traffic flows where traffic congestion effect is considered. In research literature on transportation networks, the term "traffic assignment" is used for both system optimal traffic flows problem (multi-commodity flow problem with nonlinear flow-dependent cost) and user optimal traffic flows problem (user equilibrium).

Following the notation in Section 2.1.3, the traffic assignment problem is formulated as:

$$Minimize\ z(\boldsymbol{x}) = \sum_{(i,j)\in E} f_{ij}\left(\sum_{k\in K} x_{ij}^k\right) * \left(\sum_{k\in K} x_{ij}^k\right) \qquad (2.1.4a)$$

$s.t.$:

$$\sum_{\{j:(i,j)\in E\}} x_{ij}^k - \sum_{\{j:(j,i)\in E\}} x_{ji}^k = b^k(i) \qquad \forall i \in N, \forall k \in K \qquad (2.1.4b)$$

$$0 \le \sum_{k\in K} x_{ij}^k \le u_{ij} \qquad \forall (i,j) \in E \qquad (2.1.4c)$$

The traffic assignment model is almost the same as the multi-commodity flow model shown in last section, except that c_{ij} is replaced by the unit flow cost function $f_{ij}(\sum_{k\in K} x_{ij}^k)$ in objective (2.1.4a). In research related to traffic flows, $f_{ij}(\sum_{k\in K} x_{ij}^k)$ is designed to be a convex increasing function of $\sum_{k\in K} x_{ij}^k$, which is the total amount of flows traveling through link (i,j). Branston (1976) reviewed cost-flow functions proposed by researchers at that time, which had been being used in research until today. Among those cost-flow functions the most widely used is:

$$f_{ij}\left(\sum_{k\in K} x_{ij}^k\right) = f_{ij}^0\left(1 + \alpha\left(\frac{\sum_{k\in K} x_{ij}^k}{u_{ij}}\right)^\beta\right)$$

where f_{ij}^0 is the base cost, α and β are parameters that usually take values of 0.15 and 4 respectively.

Sometimes the upper bound of the link capacity constraint (2.1.4c) is removed, since the link capacity information can be integrated into the unit flow cost function, such that the unit flow cost increases to infinity as the amount of flows on the link approaches its capacity. To give an example, Boyce et al. (1981) designed the cost-flow function as:

$$f_{ij}\left(\sum_{k\in K} x_{ij}^{k}\right) = f_{ij}^{0}\left(1 + J\left(\frac{\sum_{k\in K} x_{ij}^{k}}{u_{ij} - \sum_{k\in K} x_{ij}^{k}}\right)\right)$$

where J is a parameter reflecting the delay characteristics along a link.

As a complex nonlinear programming problem, the traffic assignment problem was commonly solved with nonlinear programming solution procedures, which are often combined with some type of decomposition method. Lin et al. (1997) applied the projected Jacobi method for the master problem and a dual Newton-type method to solve the multi-commodity flow quadratic subproblems. Commodity decomposition and arc decomposition were implemented in the dual Newton-type method designed respectively. Goffin et al. (1997) designed a potential reduction algorithm to solve the master problem with column generation technique, which defines a sequence of primal linear programming subproblems. Each subproblem generated finds a minimum cost flow between an origin-destination (OD) pair in a network with infinite link capacities. Lawphongpanich (2000) devised a simplicial decomposition procedure that used Dantzig-Wolfe decomposition for each subproblem. Lotito (2006) developed a disaggregated simplicial decomposition method with a column generation method, which solves a large number of quadratic knapsack subproblems with a Newton-like method. Other nonlinear solution procedures without decomposition include primal-dual interior-point method (Torres et al., 2009), modified analytic center cutting plane method (Babonneau et al., 2009), and alternating linearization bundle method (Kiwiel, 2011) have also been proposed to solve the traffic assignment problem.

Despite the intricacy of the traffic assignment problem, there exists significant research that has studied the problem as a network flows problem, and solve it with available network flows algorithms. Petersen (1975) proposed a primal-dual algorithm which constructed the dual problem for the linear approximation of the primal problem. The solution to the dual problem were the node potentials for each commodity. The node with the largest potential among all commodities is selected and the corresponding minimum cost flow problem for the commodity is solved. The solution obtained for that commodity replaces its solution in the primal problem, and the dual problem based on the updated primal solution is constructed for next iteration. Ouorou et al. (2000) designed a minimum mean cycle cancelling algorithm which made descent steps that involved altering the flow vector of one commodity and the vector of total flows around a cycle. And the cycle was identified with minimum mean directed cycle algorithms in residual networks related to the commodities. These studies, instead of treating the traffic assignment as an application of the nonlinear optimization problem and solving it with generic nonlinear programming solution procedures, focused on analyzing the structure of the traffic assignment problem, and developed algorithms which were evolutions of similar network flows algorithms designed for simpler network flows problems.

The traffic assignment problem discussed so far assumes the origin-destination (OD) demand $b^k(i)$ does not change over time, and thus it is often referred as the static traffic assignment problem. In the cases where time-varying demand and/or the dynamic evolution of network traffic flows are considered, the problem escalates to the dynamic traffic assignment problem, which is studied particularly in the context of transportation networks. Hence in the following part of the review until the end of Section 2.1.4., "unit flow cost" is substituted by "travel time" and "flow units" is replaced with "vehicles".

The **dynamic traffic assignment problem** keeps track of the status of all the links (i.e., number of vehicles currently on a link and the associated travel time) and vehicle flows at each point of time, and routes the vehicles that travel through the network over the planning time horizon so that the total travel time of all the vehicles is minimized. In the problem, routing decision needs to be made every time when a vehicle or a platoon of vehicles exit a link or a link segment, and link travel time is updated accordingly. In previous research, as the essential part of the dynamic traffic assignment modeling, the dynamic evolution of link traffic flow was described by four major types of models:

(1) Lighthill-Whitham-Richards (LWR) model (kinetic wave model)

(2) Point-Queue (PQ) model

(3) Spatial-Queue (SQ) model

(4) Cell transmission (CTM) model

Lighthill and Whitham (1955) and Richard (1956) modeled traffic flow as a compressible fluid of density d and fluid-velocity V (a function of d), and gave the fundamental equation of flow conservation in continuous time as:

$$\frac{\partial d}{\partial t} + \frac{\partial (Vd)}{\partial x} = 0$$

where t was the time point, x was the position along a link, and d was a function of t and x. This kinetic wave model is commonly referred as the Lighthill-Whitham-Richards (LWR) model. It facilitated the modeling of the dynamic traffic assignment problem as optimal control problems, which were solved with augmented Lagrangian method (Wie et al., 1994; Wie, 1998), and heuristics based on marginal delays (Ghali and Smith, 1994).

Point-Queue (PQ) model is a deterministic queuing model. It assumes every link (i, j) consists a free-flow segment with travel time τ_{ij}, and a queuing segment with capacity u_{ij} that restricts the number of vehicles exiting the link. A vehicle entering a link will first travel through the free-flow segment and then join the queue waiting for its turn to exit the link. Denote λ_{ij}^{t} as the total number of vehicles in the queue to leave link (i,j) at the beginning of time period t, l_{ij}^{t} as the number of vehicles leaving link (i,j) at the end of time period t, and e_{ij}^{t} as the number of vehicles entering link (i,j) at the beginning of time period t. With the presumption that there is no vehicle traveling in the network at the beginning of time $t=0$, λ_{ij}^{t} is updated as:

$$\lambda_{ij}^{t} = \begin{cases} 0, & \forall (i,j) \in E, t = 0, \dots, \tau_{ij}-1 \\ \lambda_{ij}^{t-1} + e_{ij}^{t-\tau_{ij}} - l_{ij}^{t-1}, & \forall (i,j) \in E, t = \tau_{ij}, \dots, T \end{cases},$$

and l_{ij}^{t} is updated as:

$$l_{ij}^{t} = min\{u_{ij}, \lambda_{ij}^{t}\}, \quad \forall (i,j) \in E, t = 0, \dots, T.$$

P-Q model limits the number of vehicles leaving link (i,j) to be at most u_{ij} and assumes vehicles stack up vertically so that the queue won't occupy physical length of the link. And thus there is no restriction on the number of vehicles (e_{ij}^{t}) that can enter a link.

Spatial-Queue (SQ) model updates the number vehicles waiting to leave a link (λ_{ij}^{t}) the same as the PQ model, but it is a more realistic model since it considers the fact that vehicle queue will occupy the physical space of the link. If the entire storage space of link (i,j), denoted as H_{ij}, is taken, then no more vehicles can enter the link. Consequently, with the presumption that there is no vehicle traveling in the network at the beginning of time $t=0$, e_{ij}^{t} is updated as:

$$e_{ij}^{t} = \begin{cases} min\{H_{ij}, u_{ij}\}, & \forall (i,j) \in E, t = 0, \dots, \tau_{ij}-1 \\ min\{H_{ij} - (\lambda_{ij}^{t-1} - l_{ij}^{t-1}), u_{ij}\}, & \forall (i,j) \in E, t = \tau_{ij}, \dots, T \end{cases}.$$

Unlike LWR, PQ and SQ which are whole-link models, the cell transmission model (CTM) divides each link (i,j) into M_{ij} cells with equal length of $V_{ij}\psi$, where V_{ij} is the free-flow speed of link (i,j) and ψ is the unit time interval. Daganzo (1994, 1995) showed that if the relationship between traffic flow (q) and density (d) is characterized by equation:

$$q = min\{V_{ij}d, q_{max}, b(d_{jam} - d)\}, \quad \forall 0 \le d \le d_{jam}$$

in which q_{max} is the maximum flow (or capacity), b is the backward propagation speed, and d_{jam} is the jam density, then the LWR model can be approximated by a set of difference equations with current conditions which are updated at every time interval. And the numbers of vehicles entering and leaving a link are updated according to the vehicle flow status of the first and last cells of the link. Hence, CTM is the discrete solution scheme of the LWR model and it captures the congestion evolution within a link as LWR model does.

Let $y^t_{ij(k,k+1)}$ be the number of vehicles transferred from the k^{th} cell to the $k+1^{th}$ cell on link (i,j) during time t, $x^t_{ij(k)}$ be the number of vehicles staying in the k^{th} cell on link (i,j), $H_{ij(k)}$ be the storage space of the k^{th} cell on link (i,j), and δ be the percentage of vehicles in a congested cell that can leave the cell during a unit time interval. The flow dynamics on link (i,j) can be described using the following equations:

$$x^t_{ij(k)} = \begin{cases} 0 & \forall (i,j) \in E, k = 1, \dots, M_{ij}, t = 0 \\ x^{t-1}_{ij(k)} + e^t_{ij} - y^t_{ij(k,k+1)} & \forall (i,j) \in E, k = 1, t = 1, \dots, T \\ x^{t-1}_{ij(k)} + y^t_{ij(k-1,k)} - y^t_{ij(k,k+1)} & \forall (i,j) \in E, k = 2, \dots, M_{ij} - 1, t = 1, \dots, T \\ x^{t-1}_{ij(k)} + y^t_{ij(k-1,k)} - l^{t-1}_{ij} & \forall (i,j) \in E, k = M_{ij}, t = 1, \dots, T \end{cases},$$

$$y^t_{ij(k,k+1)} = min\{x^t_{ij(k)}, u_{ij}, \delta[H_{ij(k+1)} - x^t_{ij(k+1)}]\}, \quad \forall (i,j) \in E, i = 1, \dots, M_{ij} - 1, t = 0, \dots, T$$

$$e^t_{ij} = min\{u_{ij}, \delta(H_{ij(1)} - x^t_{ij(1)})\}, \qquad \forall (i,j) \in E, t = 0, \dots, T$$

$$l^t_{ij} = min\left\{u_{ij}, x^t_{ij(M_{ij})}\right\}, \qquad \forall (i,j) \in E, t = 0, \dots, T$$

Because of its realistic modelling on link traffic flow and relative simple model structure, CTM has facilitated the research on dynamic traffic assignment problem extensively, especially in single destination networks. Ziliaskopoulos (2000) proposed a linear programming model for dynamic traffic assignment problem in single-destination networks, and proved that the necessary and sufficient condition for system optimal dynamic traffic assignment is that every unit of flow follows the time-dependent least marginal cost path to the destination. Based on that research, Zheng and Chiu (2011) developed an augmenting path algorithm to solve the single destination dynamic traffic assignment problem. Shen and Zhang (2008) concluded the PQ, SQ and CTM models gave the same optimal minimal system cost based on the numerical examples tested. And as a step further, Shen and Zhang (2014) mathematically proved the conclusion drawn in Shen and Zhang (2008), and designed a solution procedure that fitted all three models for the dynamic traffic assignment problem in single-destination networks. As to research on dynamic traffic assignment on general networks, Waller et al. (2013) proposed a CTM based model that considered demand uncertainties. Qian and Zhang (2012) designed a path-based model that adopted PQ and LWR for link flows. And a path marginal cost based algorithm was developed to solve the model formulated.

Besides the four models discussed above, there are other dynamic link traffic flow models proposed in previous research, which are discrete-time models that assume the travel time for each link updates at the beginning of every time period, and stays the same until next time period begins. These models also assume that links can accept any amount of vehicles coming in regardless of the vehicles that are already on the link, and links have first-in-first-out vehicle flows. Lafortune et al. (1993) developed a dynamic programming model, in which the link travel time was a step function of the amount of flows in the time period, and the link flow states were propagated with state transition functions, which scheduled future events based on current link flow status. Linear programming models were also developed through approximation schemes for the nonlinear objective function (Nahapetyan and Lawphongpanich, 2007), or through linearization of the link congestion function (Carey and Subrahmanian, 2000), or by modeling the travel time as piece-wise linear functions of the number of vehicles on the link (Kaufman et al., 1998).

Traffic assignment model is commonly applied to the modeling of networks with central controls on the traffic flows like railway networks. However, in networks without central control where flow units can choose their routes based on their individual objectives, a network flows model that adopts Wardrop's first principle is needed, and that is, the traffic equilibrium model.

2.1.5 Traffic Equilibrium Model. If all the users of the network travel to their destinations non-cooperatively, that is, each user chooses the route that minimizes his/her own travel cost, then the equilibrium state in which no single user can reduce his/her travel cost through unilateral route change, will be eventually reached as described in Wardrop's first principle. In traffic assignment problems, it is possible that some travelers are assigned to routes with higher cost than those assigned to others for the same OD pair, so as to achieve lower system wide total cost. This kind of flow pattern will not happen in traffic equilibrium problems.

As far as the literature reviewed, the existing traffic equilibrium models can be categorized with respect to the following aspects:

(1) whether to model the dynamic evolution of link traffic flow or not – dynamic traffic equilibrium vs static traffic equilibrium;

(2) whether to model the elasticity of demand or not – traffic equilibrium with elastic demand vs traffic equilibrium with inelastic demand;

(3) whether to consider users' perception errors on path cost or not – stochastic traffic equilibrium vs deterministic traffic equilibrium;

(4) whether to consider the multi-class composition of traffic flow or not – traffic equilibrium with heterogeneous flows vs traffic equilibrium with homogeneous flows.

The simplest traffic equilibrium model would be the one that does not consider the dynamic evolution of link traffic flow (static), and presumes users have perfect knowledge on the cost of all the routes (deterministic), demand does not change with route cost (inelastic demand), and the traffic flow only contains one class of users (homogeneous flow). Denote OD as the set of origin-destination demand, D_k as the demand of OD pair k, x_i as the total flow on link i from all OD pairs, y_{ik} as the flow from OD pair k on link i, and $f_i(x_i)$ as the flow-dependent unit flow cost (link travel time) function of link i, this **basic traffic equilibrium model** is formulated as:

$$Minimize\ z(\boldsymbol{x}) = \sum_{i \in E} \int_0^{x_i} f_i(\omega) * d\omega \tag{2.1.5a}$$

$s.t.$:

$$D_k = \sum_{\{i:E_i^- = OD_k^-, i \in E\}} y_{ik} - \sum_{\{j:E_j^+ = OD_k^-, j \in E\}} y_{jk} \qquad \forall k \in OD \tag{2.1.5b}$$

$$D_k = \sum_{\{i:E_i^+ = OD_k^+, i \in E\}} y_{ik} - \sum_{\{j:E_j^- = OD_k^+, j \in E\}} y_{jk} \qquad \forall k \in OD \tag{2.1.5c}$$

$$\sum_{\{i:E_i^- = l, i \in E\}} y_{ik} = \sum_{\{j:E_j^+ = l, j \in E\}} y_{jk}, \forall l \in N, \forall k \in \{k: OD_k^- \neq l\} \cap \{k: OD_k^+ \neq l\} \tag{2.1.5d}$$

$$x_i = \sum_{\{k \in OD\}} y_{ik} \qquad \forall i \in E \tag{2.1.5.e}$$

$$0 \leq x_i \leq u_i \qquad \forall (i,j) \in E \tag{2.1.5f}$$

where E_i^- is the head node of link i, E_i^+ is the tail node of link i, and OD_k^- and OD_k^+ are the origin node and destination node of OD pair k respectively. Constraints from (2.1.5b) to (2.1.5d) are flow conservation constraints, and constraints (2.1.5e) ensures the total link flow is the summation of flows from all OD pairs on the link. Like in traffic assignment models, the link capacity constraint (2.1.5f) is usually omitted by modelling $f_i(x_i)$ as a convex function that increases to infinity as x_i approaches u_i.

Comparing to the link-based formulation presented above, a more straightforward formulation of the basic traffic equilibrium problem is the route-based formulation since the equilibrium condition is defined on route cost. Let L_k be the route set of OD pair k, r_{lk} be the flow on route l of OD pair k, and δ_{ilk} be the binary parameter indicating whether link i is part of the route l for OD pair k or not, the route-based model is formulated as:

$$Minimize\ z(\boldsymbol{x}) = \sum_{i \in E} \int_0^{x_i} f_i(\omega) * d\omega \quad (2.1.5a)$$

$s.t.$:

$$D_k = \sum_{\{l \in L_k\}} r_{lk} \qquad \forall k \in OD \quad (2.1.5g)$$

$$x_i = \sum_{\{k \in OD\}} \sum_{\{l \in L_k\}} r_{lk} \delta_{ilk} \qquad \forall i \in E \quad (2.1.5.h)$$

$$0 \le x_i \le u_i \qquad \forall (i,j) \in E \quad (2.1.5f)$$

$$r_{lk} \ge 0 \qquad \forall l \in L_k, \forall k \in OD \quad (2.1.5i)$$

where constraint (2.1.5g) makes sure the demand of each OD pair is satisfied and constraint (2.1.5h) calculates the total amount of flow on a link from all OD pairs. The disadvantage of the route-based formulation is that it requires explicit enumeration of paths between every OD pair to obtain the route set L_k and the binary parameter set δ_{ilk}. With these two parameter sets, the multi-commodity flow problem and traffic assignment problem reviewed in previous two subsections can also be formulated as route-based models.

With the route-based formulation, Sheffi (1984) demonstrated that the first-order conditions of the Lagrangian relaxation with respect to constraint (2.1.5g) were essentially the user equilibrium conditions, and subsequently proved that the user equilibrium conditions were satisfied at the optimal point. Sheffi (1984) also proved the optimal point was unique by showing the feasible region and the objective function were convex.

The link-based traffic equilibrium problem can be efficiently solved with Frank-Wolfe algorithm (1956). Based on an initial set of feasible link flows, the algorithm repeatedly solves a linear programming problem to obtain auxiliary link flows, and performs a line search for the optimal convex combination of the auxiliary flows and the current link flows. Since the traffic equilibrium problem has a unique optimal solution, the convergence of Frank-Wolfe algorithm is assured because all search directions of line search are descent directions and all steps are descent steps. Besides line search, the method of successive average, which assigns weights of $1 - \frac{1}{n}$ and $\frac{1}{n}$ to the current flow and the auxiliary flow respectively, is also used to obtain the convex combination of flows. The convergence of Frank-Wolfe algorithm with successive average method was proven by Powell and Sheffi (1982). Even though the Frank-Wolfe algorithm with either line search or successive average method converges, the converging process is considered slow. To accelerate the convergence, Patriksson (1994) proposed a simplicial decomposition approach which stores all the auxiliary flow vectors generated in previous iterations and obtain the optimal convex combination of all these flow vectors as the resulting flow of current iteration.

If the OD demand is not fixed but considered as a decreasing function of the traveling cost between the OD pair, then the elastic demand is modeled in the traffic equilibrium problem. Let $Q_k^{-1}(\omega)$ be the inverse of the demand function associated with the travel cost of OD pair k, the route-based **traffic equilibrium problem with elastic demand** is formulated as:

$$Minimize\ z(\boldsymbol{x}) = \sum_{i\in E}\int_0^{x_i} f_i(\omega) * d\omega\ - \sum_{k\in OD}\int_0^{D_k} Q_k^{-1}(\omega)d\omega \tag{2.1.5j}$$

$s.t.$:

$$D_k = \sum_{\{l\in L_k\}} r_{lk} \qquad \forall k \in OD \tag{2.1.5k}$$

$$x_i = \sum_{\{k\in OD\}}\sum_{\{l\in L_k\}} r_{lk}\delta_{ilk} \qquad \forall i \in E \tag{2.1.5.h}$$

$$0 \le x_i \le u_i \qquad \forall (i,j) \in E \tag{2.1.5f}$$

$$r_{lk} \ge 0 \qquad \forall l \in L_k, \forall k \in OD \tag{2.1.5i}$$

$$D_k \le \overline{D_k} \qquad \forall k \in OD \tag{2.1.5l}$$

where $\overline{D_k}$ is the upper bound of the demand that can be generated from OD pair k. It should be noted in the formulation above is that D_k now is a variable instead of a parameter, and that is also why constraint (2.1.5l) is included to define the value range of D_k. Sheffi (1984) constructed the Lagrangian of the problem with respect to constraint (2.1.5k), and proved the route-based formulation had unique optimal solution, and the optimal solution satisfies the user equilibrium condition with elastic demand.

With initial link traveling cost based on the presumption that there is no flow, and through iterative calculation of the path cost, corresponding demand, auxiliary link flows, and link traveling cost, method of successive averages can be adapted to solve the traffic equilibrium problem with elastic demand (Bell and Lida, 1997). Simple changes in the representation of the problem, such as the zero-cost overflow formulation and the excess-demand formulation, can also make the problem amenable for solution with fixed-demand equilibration algorithms (Sheffi, 1984).

The basic traffic equilibrium model and the model with elastic demand discussed above assume users have perfect information on route travel cost (e.g., travel time) over the entire network, and thus are referred as the deterministic models. In contrast to that, the **stochastic traffic equilibrium models** assume travelers do not know the actual cost of routes, and their perceived route cost is the actual route cost plus a random error term. Travelers choose the routes with the minimum perceived travel cost and eventually will reach the stochastic user equilibrium state, which is described as: no travelers can improve his or her perceived travel cost by unilaterally changing routes.

Denote P_{lk} as the probability that route l of OD pair k is chosen among all the routes connecting this OD pair, C_{lk} as the random variable representing the perceived travel cost on route l of OD pair k, and $\boldsymbol{f}$ as the given set of measured travel costs (actual travel cost for each route), in the case that demand is inelastic, the stochastic user equilibrium (SUE) conditions can be characterized by the following equations:

$$r_{lk} = D_k P_{lk} \qquad \forall k \in OD, \forall l \in L_k$$

$$P_{lk} = P_{lk}(\boldsymbol{f}) = P(C_{lk} \leq C_{l'k}, \forall l' \neq l, l' \in L_k, l \in L_k | \boldsymbol{f}) \qquad \forall k \in OD, \forall l \in L_k$$

The route choice probability is interpreted as the probability of perceived travel cost of the chosen route being the least among all the routes between the OD pair. Therefore, at stochastic user equilibrium, the cost on all used paths is not going to be equal but will conform the SUE conditions listed above.

To describe the route choice probability function $P_{lk}(\boldsymbol{f})$, various route choice models were proposed in previous research, and among them the multinomial logit (MNL) and multinomial probit (MNP) were the two earliest models. The multinomial logit model assumes the random error terms of the perceived travel cost are independently and identically distributed Gumbel variables, and derives the route choice probability as:

$$P_{lk} = \frac{\exp\left(f_{r_{lk}}\right)}{\sum_{l' \in L_k} \exp\left(f_{r_{l'k}}\right)}$$

where $f_{r_{lk}}$ is the measured travel cost of route r_{lk}. Even though the multinomial logit model gives the route choice probability in a nice closed form, it has two major deficiencies (Sheffi, 1984). First, it lacks sensitivity to network topology and this results in assigning too much flow to partially overlapped routes. Second, it calculates route choice probabilities solely based on route cost differences, and does not consider the dependence of the perception variance on the measured route cost. Many extensions of the multinomial logit model, such as the C-logit, implicit availability/perception logit, path-size logit, paired combinatorial logit, cross-nested logit, generalized nested logit, and logit kernel (mixed logit), were developed to fix the deficiencies while preserving the analytical tractability of the logit-type model. Prashker and Bekhor (2004) gave a comprehensive review on these models and integrated them into the modeling of stochastic traffic equilibrium problem.

The multinomial probit model assumes the random error terms are normal random variables with zero mean, and consequently the joint density function of the error terms is a multivariate normal function. The variance-covariance matrix usually is constructed based on the measured route cost and the cost of overlapped part of two routes (Sheffi, 1984; Yai et al., 1997). The multinomial probit model does not have the two deficiencies as the logit model and thus generates flow patterns that are more reasonable. However, it requires high computational cost when there are more than two alternative routes, because the route choice probability function, which is the cumulative distribution function of a multinomial random variable, does not have a closed form. To evaluate the route choice probability, analytical approximation methods like numerical integration algorithms and successive approximation method, and Monte Carlo simulation were adopted in previous research, which were reviewed by Sheffi (1985) and Rosa and Maher (2002).

More recently, Castillo et al. (2008) used Weibull distribution to model the random perception error terms, and proposed a multinomial weibit (MNW) route choice model to capture the route-specific perception variance. The MNW model has advantages over the MNL and MNP models because it has a closed-form route choice probability function, and it is able to model perception variance as an increasing function of the measured route cost. Based on this, Kitthamkesorn and Chen (2013) designed a path-size weibit model which resolved the route overlapping issue with the introduction of a path-size factor. This path-size factor adjusts choice probabilities for routes with strong couplings so as to prevent too much flow being assigned to overlapping routes.

Without the integration of specific route choice models, Sheffi (1984) formulated the general stochastic traffic equilibrium problem as an optimization problem with the objective:

$$\min_{\boldsymbol{x}} z(\boldsymbol{x}) = -\sum_{k \in OD} D_k \left(E\left[\min_{l \in L_k}\{C_{lk}\} | \boldsymbol{c_k}(\boldsymbol{x})\right]\right) + \sum_{i \in E} x_i f_i(x_i) - \sum_{i \in E} \int_0^{x_i} f_i(\omega) d\omega$$

where $\boldsymbol{x}$ is the set of route flows for all the OD pairs, $\boldsymbol{c_k}(\boldsymbol{x})$ is the actual cost of the routes connecting OD pair k, and $E\left[\min_{l \in L_k}\{C_{lk}\} | \boldsymbol{c_k}(\boldsymbol{x})\right]$ is the expected perceived travel cost for OD pair k. Represent the expected perceived travel cost function $E\left[\min_{l \in L_k}\{C_{lk}\} | \boldsymbol{c_k}(\boldsymbol{x})\right]$ by $S_k[\boldsymbol{c_k}(\boldsymbol{x})]$, since $\frac{\partial S_k(\boldsymbol{c_k})}{\partial c_{lk}} = P_{lk}$ and $\frac{\partial^2 S_k(\boldsymbol{c_k})}{\partial c_{lk}^2} = \frac{\partial P_{lk}(c_{lk})}{\partial c_{lk}} \leq 0$ because routes with higher actual cost should have smaller probability of being perceived as the route with least perceived cost, $S_k[\boldsymbol{c_k}(\boldsymbol{x})]$ is concave with respect to $\boldsymbol{c_k}(\boldsymbol{x})$**.** With the properties of $S_k[\boldsymbol{c_k}(\boldsymbol{x})]$ regarding its first and second partial derivatives on $\boldsymbol{c_k}(\boldsymbol{x})$, Sheffi (1984) showed the first-order conditions of the optimization problem coincided with the SUE conditions and proved the optimal solution was the stochastic user equilibrium. Since $f_i(x_i)$ is monotonic, the inverse $x_i(f_i)$ exists. And thus the objective $z(\boldsymbol{x})$ can be transformed as a function of link traveling cost $\boldsymbol{f}$ (which is $z(\boldsymbol{f})$) rather than link flows $\boldsymbol{x}$ (which is$z(\boldsymbol{x})$). This means $z(\boldsymbol{f})$ and $z(\boldsymbol{x})$ are monotonic transformation to each other, and each point of $z(\boldsymbol{x})$ corresponds to one and only one point of $z(\boldsymbol{f})$. With this property of $z(\boldsymbol{f})$ and $z(\boldsymbol{x})$, Sheffi (1984) proved $z(\boldsymbol{f})$ had a unique minimum by showing its Hessian matrix was positive definite, and proved $z(\boldsymbol{x})$ also had a unique minimum which was the stochastic user equilibrium.

Based on the route choice models adopted in the stochastic traffic equilibrium problem, various solution approaches have been developed. Stochastic traffic equilibrium with logit-type route choice models can be solved with Powell-Sheffi algorithm (Powell and Sheffi, 1982), modified Frank-Wolfe algorithm (Akamastu, 1996), path-based partial linearization method (Chen et al., 2012), and self-adaptive gradient projection algorithm (Zhou et al., 2012). For stochastic equilibrium models based on MNP, the most commonly used approaches are based on Monte Carlo simulation (Sheffi, 1984; Clark et al., 2002). As to weibit stochastic user equilibrium models, Kitthamkesorn (2014) developed a link-based solution algorithm which obtained a search direction by solving a convex auxiliary problem (i.e., the first-order approximation of the objective function), and performed line search based on the search direction to calculate the step size and solution of current iteration.

Recent research also studied the modeling and solution methods for stochastic traffic equilibrium with elastic demand. Most of the research reviewed adopted logit-type route choice models (Ryu et al., 2014; Sun et al., 2015; Xu et al., 2013; Yu et al., 2014); only Meng et al. (2012) studied the problem with multinomial probit route choice model. Solution approaches proposed have been quite similar to those developed for the problem with inelastic demands. But there were also new solution methods like the predictor-corrector interior point algorithm designed by Yu et al. (2014).

Most of the research on stochastic traffic equilibrium assumes the actual link and route travel costs at free-flow conditions are deterministic. However, this assumption is not realistic since the free-flow travel cost will be different in different weather and road conditions, and will be affected by non-routine traffic delays. Mirchandani and Soroush (1987) relaxed that assumption and proposed a generalized stochastic traffic equilibrium model where the free-flow travel cost on a link is probabilistic, introducing another level of randomness besides the random perception errors on travel cost. They studied the problem with linear, exponential and quadratic disutility functions, and solved it with a generalized incremental loading assignment technique.

Like the dynamic traffic assignment problem, in the cases where time-varying demand and the dynamic evolution of link traffic flows are considered in the traffic equilibrium study, the **dynamic traffic equilibrium problem** arises. To model the dynamic evolution of link traffic flows, research on dynamic traffic equilibrium has used LWR model (Bellei et al., 2005; Kachroo and Ozbay, 1998;), point-queue model (Gawron, 1998; Han, 2003; Tong and Wong, 2010; Iryo, 2015), spatial queue model (Balijepalli et al., 2014), cell-transmission model (Balijepalli et al., 2014; Golani et al., 2004; Levin et al., 2015a; Meng and Khoo, 2012; Qian and Zhang, 2013; Waller and Ziliaskopoulos, 2006), and various other models with combinations of link performance functions and flow conservation functions (Carey, 2009; Kachroo and Ozbay, 2005; Li et al., 2013a; Papageorgiou, 1990; Varia and Dhing 2004; Wie et al., 1990; Yang et al., 2012).

Similar to the original version of Wardrop's first principle that describes the static traffic equilibrium, the dynamic generalization of Wardrop's first principle is stated as:

"If, at each instant in time, for each origin-destination pair, the instantaneous expected unit travel costs for all the paths that are being used are identical and equal to the minimum instantaneous expected unit path cost, the corresponding time-varying flow pattern is said to be user optimized." (Wie et al., 1990)

The generalized Wardrop's first principle applies to the **dynamic deterministic traffic equilibrium problem**, which assumes every user has perfect knowledge on the path cost throughout the time horizon.

Based on the link traffic flow models adopted, the dynamic deterministic traffic equilibrium problem may have different solution properties. Szeto et al. (2006) gave a detailed comparison between point-queue models and spatial-queue models on route cost properties and solution properties. They showed that dynamic user equilibrium existed in point-queue models but might not exist in spatial-queue models, and both of these two types of models might have multiple equilibria. For point-queue models, the existence of dynamic equilibrium was mathematically proven by Mounce (2007), and multiple equilibria was shown by Iryo (2011). However, the solution properties of dynamic equilibrium solutions with the prevalent cell-transmission model have not been thoroughly investigated.

The dynamic deterministic traffic equilibrium problem has been studied with solution approaches from three disciplines: control theory, nonlinear programming, and simulation. Research that studied the dynamic user equilibrium as control problems commonly applied nonlinear optimal control methods (Papageorgiou, 1990) or feedback methodologies (Papageorgiou, 1990; Kachroo et al., 1998; Kachroo et al., 2005). In literature where dynamic user equilibrium was formulated as nonlinear programming problems, and combinatorial solution procedures have been proposed to solve the problem (Golani et al., 2004; Janson, 1991; Waller et al., 2006). Due to the convenience of describing the dynamic evolution of traffic flows, simulation methods have been the most popular approach to the dynamic equilibrium problem. It either is used as a platform to develop new and efficient traffic equilibrium assignment algorithms (Gawron, 1998; Levin et al., 2015; Varia et al., 2004; Yang et al., 2012) and mechanisms that improve the efficiency of existing algorithms (Balijepalli et al, 2015; Levin et al., 2015; Tian et al., 2014), or provided results for solution procedures developed to compare with (Li et al., 2013). Besides solution approaches from those three disciplines, Carey (2009) proposed a bi-level dynamic user equilibrium framework, which separated the loading of flows on the time-space network from the modeling of flows and trip times of individual links.

The stochastic version of the dynamic traffic equilibrium problem relaxes the presumption that every user has perfect knowledge about route cost, and assumes users perceive route cost with a random perception error and choose the route with the minimum perceived cost at each time instant. Hence, at dynamic stochastic traffic equilibrium, for each OD pair and at each instant in time, no user can reduce his or her perceived route travel cost by unilaterally changing routes. Iryo (2015) showed the existence and uniqueness of dynamic stochastic equilibrium in a simple loop network with point-queue model for link traffic flows. Solution properties of dynamic stochastic equilibrium with other link flow models and route choice models have not been investigated yet.

The handful of papers found on the dynamic stochastic traffic equilibrium adopted either the basic multinomial logit model (Bellei et al., 2005; Han, 2003; Qian et al., 2013) or the multinomial probit model (Meng et al., 2012; Zhang et al., 2008) for the route choice probability function. The solution methods proposed include method of successive average (Han, 2003; Meng et al., 2012; Zhang et al., 2008), pure network loading (Qian et al., 2013; Han, 2003), diagonalization method (Han, 2003), quadratic interpolation (Han, 2003), Bather's method (Bellei et al. 2005) and Ishikawa algorithm (Meng et al., 2012). Chong et al. (2014) modeled the dynamic route choice as the conditional joint distribution of route traffic given that the network was in dynamic stochastic equilibrium, and developed a Metropolis-Hastings sampling scheme to solve the dynamic stochastic equilibrium problem.

Little research is available on models and solution approaches for dynamic traffic equilibrium with elastic demand (Guo et al., 2015). Because the dynamic traffic equilibrium problem has an additional temporal dimension than the static traffic equilibrium problem, it is natural to include more flexibility in demand modelling than merely accounting for the demand elasticity. Research has studied the demand variability by defining departure times as variables to be optimized, and to minimize route travel times at equilibrium (Han et al., 2011; Heydecker et al., 2005; Huang et al., 2002; Huang et al., 2002; Li et al., 2008; Lim et al., 2005; Long et al., 2015; Mahmassani et al., 1984; Mun, 2011). These research formulated the dynamic traffic equilibrium problem with departure time choices as nonlinear optimization problems, and proposed various heuristics and meta-heuristics (e.g., genetic algorithm) to solve the models developed.

Traffic equilibrium models discussed so far assume traffic flow is homogeneous. In transportation networks, flow homogeneity means all the vehicles or travelers are the same in all aspects (e.g., vehicle type, link travel time function, route choice behavior, etc.) except for their origins and destinations. However, it is common sense that traffic flow is composed of vehicles in different physical sizes and drivers with different driving behaviors. Hence, to model traffic equilibrium more realistically, it is necessary to consider the **heterogeneity of traffic flow.**

Numerous research studied the modelling of traffic equilibrium with heterogeneous flows in transportation networks. To deal with flow heterogeneity, these research divided travelers/vehicles into a number of classes, and assigned each class of users with different utility functions (Konishi, 2004), or value of time (Han and Yang, 2008; Huang and Li, 2007; Jiang et al., 2011; Lu and Mahmassani, 2008; Lu and Mahmassani, 2009), or link travel cost/time (Bliemer and Bovy, 2003; Mahmassani and Mouskos, 1988; Scrimali, 2014; Wu et al., 2006), or toll amounts (Ye, 2010).

Numerous research studied the modelling of traffic equilibrium with heterogeneous flows in transportation networks. To deal with flow heterogeneity, these research divided travelers/vehicles into a number of classes, and assigned each class of users with different utility functions (Konishi, 2004), or value of time (Han et al., 2008; Huang et al., 2007; Jiang et al., 2011; Lu et al., 2008; Lu et al., 2009), or link travel cost/time (Bliemer et al., 2003; Mahmassani et al., 1988; Scrimali, 2014; Wu et al., 2006), or toll amounts (Ye et al., 2010).

In stochastic equilibrium problems, flow heterogeneity was also captured in route choice models, so that the routing behaviors of users in different classes were described by route choice models with different parameter values. For example, for logit-based route choice models, different classes of travelers have different dispersion parameters (Miwa et al., 2010) or different variances for route cost perception errors (Jaber et al., 2009). And for probit-based route choice models, travelers in different classes have different variance-covariance matrices (Connors et al., 2007; Lee, 2008; Zhang et al., 2013). Di et al. (2008) proposed a travel time budget model that differentiated travelers based on their risk-taking preferences. In that paper, travelers were categorized into three classes (i.e., risk averse, risk prone and risk neutral) and each class was assigned with a distinct travel time risk, which was the probability that a trip could not be completed within a certain amount of time given the probability density function of the trip time. The risk-based route disutility was calculated as the summation of the expected perceived trip time and a risk-factored term, which was the product of normalized quantile of completing the trip with class-specific risk value, the weight for route travel time variance, and the variance of the perceived route travel time. Based on the model proposed by Di et al. (2008), Nie (2011) modeled the perceived trip travel time as the convolution of flow-dependent perceived link travel time and proposed a link-based model. Wu et al. (2013) devised an efficient gradient projection algorithm to solve the model proposed in Nie (2011), which avoided path enumeration through a column generation procedure based on a reliable shortest path algorithm. With the same classification of travelers based on the risk-taking preference, Xu et al. (2014) designed a mean-excess travel time model that did not only consider travel budget but also accounted for demand elasticity.

The multi-class traffic equilibrium problem has been studied in dynamic settings as well (Bliemer et al., 2003; Lee, 2008; Lu et al., 2008; Lu et al., 2009; Scrimali, 2014; Zhang et al., 2013). Compared to the static models, the dynamic models proposed described the traffic flow with more details. These models assumed overtaking behaviors could happen among vehicles in different classes, and vehicles in the same class still obeyed the First-In-First-Out rule while they were traveling in a link. The class-specific link flow status was updated and link travel cost was calculated based on the aggregated flow on the link for each class.

The equilibrium states of various models with heterogeneous traffic flows (i.e., static or dynamic, deterministic or stochastic, and elastic demand or inelastic demand) can be described similarly to the counterpart models with homogeneous flows. The solution approaches developed are also quite similar to the homogeneous flow cases except for specific considerations for class-specific travel cost calculation and route assignment.

As a conclusion for this subsection, the traffic equilibrium problem is a big topic with a broad scope. Traffic equilibrium is not only studied in the context of traffic flow modeling in transportation networks, but also in other subjects like the power transmission in power distribution networks and packets routing in fiber networks. This subsection only reviewed fundamental and major equilibrium models that have been extensively studied in previous research. Other types of traffic equilibrium models, such as the model considering link interactions in which travel cost of a link also depends on the flows on other links, and the equilibrium modeling of modal split where travel demand can split and take different modes of transportation (e.g., cars, buses, and light rails), are not covered in this review.

2.2 General Network Maintenance Planning

Network maintenance planning has been studied with applications in various industries. Among the rich literature found, some researches have investigated this problem with a network-wide perspective. They schedule the maintenance of network components to achieve maximum overall network performance or minimum total maintenance cost. Criteria that evaluate the maintenance plan on its impact on system-wide network performance, such as network reliability, network operating cost, and network flows disruption, have been adopted in previous research. This section reviews the maintenance planning for networks other than the transportation network, emphasizing the general modeling approaches adopted in literature.

The reliability modeling approach has been widely applied in the research of maintenance planning for bridge networks (Bocchini and Frangopol, 2011; Bocchini and

Frangopol., 2013; Frangopol and Liu, 2007; Hu et al., 2015; Liu et al., 2005; Liu et al., 2006; Morcous et al., 2005), power generation and transmission networks (Marquez et al., 2013; Usberti et al., 2015), water distribution pipe networks (Luong et al., 2005), and railroad networks (Zhang et al., 2013). With Markovian models (Luong et al., 2005; Morcous et al., 2005; Orcesi et al., 2010) or reliability index profiles which are functions of time and repair effectiveness (Bocchini et al., 2011; Bocchini et al., 2013; Hu et al., 2015; Liu and Frangopol, 2005; Liu and Frangopol, 2006; Marquez et al., 2013; Usberti et al., 2012; Zhang et al., 2013b), the reliability modeling approach models the deterioration process and condition improvements after maintenance for each network component. The long-term network level reliability then is evaluated by objective functions that aggregate network components' condition throughout the planning horizon.

The objective functions used in the literature reviewed can be categorized into three major types. The first type of objective functions calculate the weighted average based on the reliability indicators of individual network components. Exemplary objective functions in this type include the weighted average bridge condition (Morcous and Lounis, 2005), the total weighted long-run availability of all the pipes (Luong and Nagarur, 2005) and the expected number of power failures per year for each customer (Usberti et al., 2012). The second type of objective functions minimize the total maintenance cost over the planning horizon, which are constrained by required level of network reliability like the connectivity requirements in bridge networks (Bocchini and Frangopol, 2013; Liu and Frangopol, 2005; Liu and Frangopol, 2006). The third type of objective functions minimize the summation of total network usage cost and maintenance cost over the period of time under consideration. In the models where the third type of objective functions are applied, the unit cost of using the network components (e.g., links) depends on the condition of the component. And the objective function requires maintenance to be scheduled so that the maintenance cost is minimized, and the resulting condition of network components gives the minimum total users' cost over the planning horizon (Hu et al., 2015; Orcesi et al., 2010).

In network operating cost modeling, the optimality of a maintenance plan is evaluated more directly. For bridge networks, Bocchini and Frangopol (2011) evaluated the maintenance schedule by the total flow cost at users' equilibrium. For power generation and transmission networks, based on the fact that the unit costs of power generation for different generators were different, Marwali and Shahidehpour (1998), Marwali and Shahidehpour (1999), and Niazi et al. (2015) developed models that minimized the total energy production cost during the maintenance period.

Among literature reviewed on bridge network maintenance planning, only Orcesi and Cremona (2010) considered the impact of bridge capacity reduction caused by maintenance activities on network flows. The rest of the literature assumed the bridge would not be closed or have capacity reduction during the maintenance, which could be a reasonable presumption if the planning time horizon for the entire network is much longer than the time period when the bridge is under maintenance. In power generation and transmission networks, more research was conducted on short-term maintenance scheduling. For safety reasons, generators or transmission lines have to be physically disconnected from the network for maintenance activities. To deal with the temporal unavailability of generators and transmission lines, Gomes et al. (2007) proposed a model to minimize the number of critical power transmission branches. In graph theory, the critical branch is defined as the only branch connected to the vertex point, the removal of which will disconnect the network. Goel et al. (2013) developed a workforce routing and scheduling model to minimize the total down time of transmission lines and the travel effort of maintenance crews. Efficient workforce routing is an important factor to consider in power transmission line maintenance planning since maintenance crews have to travel along the long stretches of transmission lines to maintain them. Similar types of workforce routing and scheduling models were proposed in literature on railroad network maintenance scheduling as well (Peng and Ouyang, 2012; Zhang et al., 2013b).

In research that adopted network flows modeling approach, the temporal capacity reduction or unavailability of network components, and its impact on network flows were studied. Tawarmalani and Li (2011) proposed a mixed-integer programming model that scheduled link maintenance in abstract tree networks to minimize the total flow disruptions, which was the difference between the flow patterns before and during the maintenance. Boland et al. (2014) studied the network maintenance scheduling with the objective of maximizing the total flow over the planning time horizon, and investigated the problem as a maximum total flow problem with flexible link outages. Based on Boland et al., (2014), Boland et al. (2015) extended the research and developed continuous-time models that considered storage nodes. In that research, integer programming models based on time discretization were developed to provide primal bounds and dual bounds for the continuous time problem. Both Boland et al. (2014) and Boland et al. (2015) applied the models developed to the maintenance scheduling of a coal mine production network.

Research reviewed in this section studied maintenance planning in networks that had relatively simple network flows attributes (e.g., single OD demand, single commodity), and few research explicitly considered or modeled these attributes. In research on maintenance planning and scheduling for transportation networks, the flow demand constraints, flow conservation constraints, and equilibrium conditions were more commonly considered in models developed. And those studies are reviewed in next section.

2.3 Maintenance Planning in Transportation Networks

The repair and maintenance of road network results in "work zones", where some lane segments of a link are closed for a predicted period of time until the work is completed. Work zone planning is a challenging task since there are multiple parties involved and more than many factors need to be taken into consideration. Bayraktar and Hastak (2009) reviewed the factors impacting the success of work zone projects. They modeled the relationships between the goals of the project stakeholders and public satisfaction of the project using Bayesian belief networks. The model was aimed to assist highway agencies in developing suitable contracting strategies considering 52 interrelated factors impacting the success of work zone projects, which were grouped into four categories (contract characteristics, motorist issues, public issues, and resource issues). Despite the comprehensive list of factors taken into account, the model can only help prepare bids and not help to actually schedule the work zones.

Most of the literature related to the maintenance planning in transportation networks can be grouped into four categories. The first category includes research that investigated the long-term network rehabilitation planning problem with the objective of maintaining the roads in good condition with least cost in different aspects. For example, Smilowitz and Madanat (2000) proposed a linear programming model to determine the optimal maintenance activities for each link at each time interval that minimized the total maintenance cost and user cost over the planning time horizon. Both user cost and maintenance cost of a specific maintenance type were functions of the link states. And the link states were modeled as a Markovian process to capture the quality deterioration and maintenance effectiveness. To give another example, Chu and Chen (2012) developed a bi-level hybrid dynamic model in which the upper level problem decides the optimal threshold for each road that triggers maintenance action and the lower level problem solves the user equilibrium problem. These two levels of problems are connected by the road deterioration function which models the effects of traffic loads on a road and the impacts of road roughness on users' traveling cost. This type of research considers network-wide maintenance planning over a relatively long period of time (a year or longer). By assuming the project period is much shorter than the planning horizon, they omitted the impact of temporary link capacity reductions on traffic flow caused by the maintenance work. However, this assumption is not always reasonable especially for the maintenance work like resurfacing sets of links which would take months or longer. When the length of project period is comparable to the planning horizon, it is necessary to consider the effect of temporary link capacity reductions and to schedule the work zones in the way that minimizes the negative impacts on traffic flows.

Research in the second category focused on developing operational strategies for work zone scheduling on a highway segment or a local arterial. Some research in this category has studied the short-term work zone scheduling with time horizons less than a day. This research focuses on optimizing the work zone planning of a single link but does not consider the impact of possible diverting traffic resulted from work zones to other links that are connected to or close to the work zone; see e.g., works of Meng and Weng (2013), Tang and Chien (2008) and Jiang and Adeli (2003). However, in reality, as long as traffic congestion exists and there are alternative routes available, some portion of the traffic will divert to other routes which will affect the traffic on those alternative routes. Chien and Tang (2014) proposed a genetic algorithm to optimize the work zone length and start time in a day of the maintenance work on a highway stretch. The optimal schedule minimizes the total cost to the agencies conducting the maintenance plus the cost to the road users. Even though the temporary link capacity reductions, and resulting increased road user cost, and possible traffic diversion, were modeled, only one alternative route for the diverted traffic was considered. Often there are more than two lanes for some segments of highway, but Chien and Tang (2014) did not explicitly explore different lane closing scenarios. Schroeder and Rouphail (2010) compared different lane closure scenarios and discussed the operational impacts of freeway work zones on traffic. Their approach can only compare every limited number of scenarios since each scenario requires extensive analysis. Summarizing, the research in this category focuses on scheduling work zones on single links and has very limited or no consideration on the impact of traffic diversion resulting from multiple link capacity reductions.

The third category consists research that studied the scheduling of network expansion projects. This type of research specifically considered the flow pattern changes caused by the increase of link capacities or the addition of new links over the planning time horizon. This research topic is closely related to the network design problem, which selects among a set of candidate links to be added to a network with budget constraints, so as to achieve lowest total cost at users' equilibrium state or system optimum. It is an extension of the network design problem since the addition of the chosen links need to be scheduled, and possible traffic flow pattern changes need to be evaluated after the addition of each link. Fontaine and Minner (2014) developed a mixed-integer programming model to select and schedule network expansion projects with minimum total project cost and system optimum flow cost, and solved it using Bender's decomposition. Bagloee and Asadi (2015) presumed the set of network expansion projects were given and only one of these projects could be worked on at a time, and studied the network expansion scheduling problem as a traveling sales man problem to determine the optimal sequence of the expansion projects. The inter-dependency of the expansion projects was evaluated using the artificial neural network model, so that the "cost" of "moving" from one expansion project to another could be computed. Gao et al. (2011) combined the problems of road maintenance and road expansion planning, and developed a mixed-integer, nonlinear, bi-level model that scheduled the repair or expansion of every road with budget constraints. In the model proposed, the road capacity increase after maintenance and expansion were considered, and the road degradation process was modeled. General Bender's decomposition method was applied to obtain the optimal maintenance and expansion schedule that gave the minimum total users' cost at equilibrium state. Although literature reviewed in this category modeled the capacity increase after the maintenance or expansion, they did not consider the link capacity reductions during the time period when these activities were being performed.

Only a handful of works considered the impact on traffic over the network due to multiple work zones and they comprised the fourth category. Orabi and El-Rayes (2012) developed a complex model with three genetic algorithm based modules – scheduling, network performance, and user savings, to select and prioritize rehabilitation projects, subject to budget constraints. Lee (2009) proposed a work zone scheduling model which considered the routing-changing behavior of road users. The schedule was optimized with an ant colony algorithm, where the users' equilibrium under each schedule scenario was obtained through simulations using VISSIM software. Hosseininasab and Shetab-Boushehri (2015) studied the work zone scheduling problem as a time-dependent network design problem. They formulated the problem as bi-level programming models, and used genetic algorithm to obtain the link maintenance schedule that gave the minimum total traveling cost at equilibria over the planning time horizon. All the three of Orabi and El-Rayes (2012), Lee (2009) and Hosseininasab and Shetab-Boushehri (2015) did not explicitly discuss partial link capacity reductions resulting from work zones. Zheng et al. (2014) assumed the link capacity would reduce by 50% in their decision model developed.

However, a link might have more than two lanes and it is not always true or optimal to close half of the lanes at a time for maintenance. Ma et al. (2004) developed a hybrid simulation methodology with genetic algorithm to schedule multiple lane closures with minimum total traffic delay of the network. However, the flexible lane-level maintenance scheduling required high computation effort for the solution approach proposed in Ma et al. (2004). For a problem instance of scheduling the maintenance of 20 lanes, it took more than 120 hours.

In maintenance planning with network flows modeling approach, the network capacity reductions are mandatory since the maintenance work has to be completed before the due date. In cases when budget is not the major concern, optimal maintenance scheduling is essentially managing mandatory network capacity reductions so that the negative impacts on flows is minimized. Due to the existence of the well-known Braess' Paradox when the user equilibrium principle is adopted, and link capacity drops when congestion occurs, network capacity management methods that intentionally reduce the capacity of some links, such as imposing link tolls and ramp metering, could also improve the overall performance of the network if the objective is to minimize total travel cost at users' equilibrium. Hence, next section reviews research that studied the design of these network capacity management mechanisms, and how they help improve the overall network performance.

2.4 Traffic Flow Control Mechanisms

To improve safety, alleviate congestion, and eliminate chaos at intersections, traffic flow control mechanisms, such as traffic lights and link speed limits, have been implemented in local transportation networks since a hundred years ago. Freeways were originally expected to provide unlimited mobility. However, because of the quick increase of traffic demand over time, demand peaks during rush hours, and link capacity reduction caused by incidents, freeways without traffic controls result in similar situations as the local networks prior to the introduction of traffic lights: blocked segments and reduced safety (Papageorgiou and Kotsialos, 2002). To restore and maintain the maximum utilization of freeways, traffic flow management strategies such as tolled imposition, ramp metering and speed limits, have been studied in numerous researches. Unlike most research developing traffic management strategies specifically for highway networks, this section and the research presented in later chapters do not differentiate between local and freeway networks, trying to generalize these traffic management strategies and apply them to transportation networks in general.

2.4.1 Manage Network Flows through Ramp Metering. Ramp meters are the two-phase signal lights installed at the entrance ramps of freeways. Upon activation, ramp meters will turn on the red light to enforce each vehicle entering the freeway to wait for a period of time, and then switch to the green light to let the vehicle enter the mainline. Papageorgiou and Kotsialos (2002) showed that ramp meters could effectively ameliorate local traffic conditions through restricting the amount of traffic flowing into the mainline, and by increasing traffic flows exiting the mainline. They illustrated their arguments with graphs shown in Figure 2.4 – i and Figure 2.4 – ii below, where the shaded areas are the congested zones:

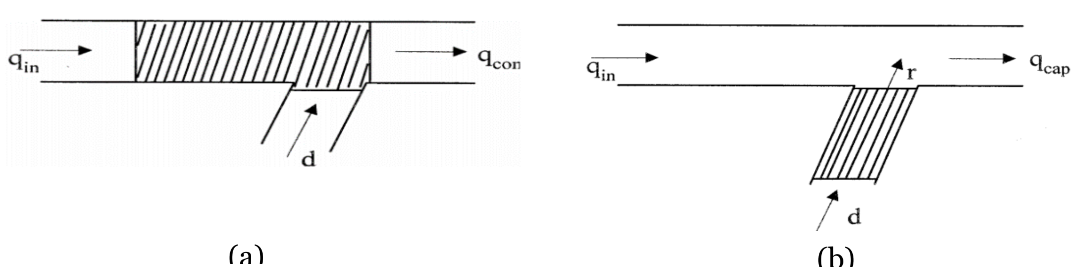
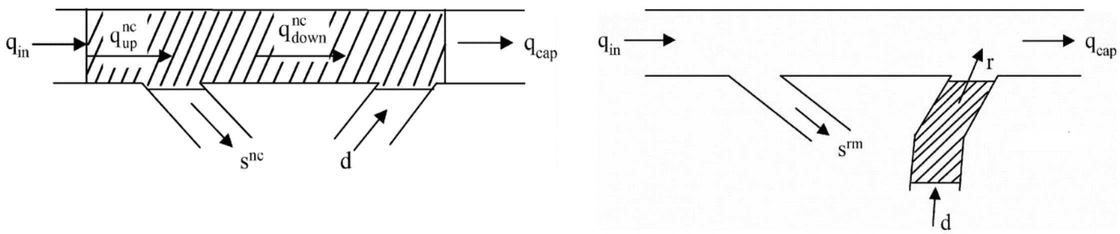


Figure2.4.1-i: (a) without and (b) with Ramp Metering (Papageorgiou and Kotsialos, 2002)



(c) (d)

Figure2.4.1-ii: (c) without and (d) with Ramp Metering (Papageorgiou and Kotsialos, 2002)

Graph (a) and (b) depict that by ramp metering, even though vehicles will wait in queues on ramps to enter the freeway, traffic condition in the mainline ameliorates and that benefits a larger group of travelers. Graph (c) and (d) illustrate that the improved mainline traffic resulted from ramp metering can increase the traffic flow that needs to exit the mainline, leaving more mainline capacity for traffic downstream.

The proven effectiveness of local traffic control by ramp metering led to the rich literature produced on individual/isolated ramp metering algorithms (Abdel-Aty et al., 2007; Chi et al., 2013; Chow and Li, 2014; Elefteriadou et al., 2014; Jin et al., 2014; Perrine et al., 2015; Rezaee et al., 2013; Wang et al., 2010; Wang et al., 2014; Zhao et al., 2011). Various ramp metering algorithms that coordinate among different ramp meters have also been developed in previous research (Bhouri et al., 2013; Chai et al., 2015; Dominguez and Fernandez, 2012; Geroliminis et al., 2011; Gomes and Horowitz, 2006; Jiang and Chung, 2015; Kotsialos et al., 2004a; Kotsialos et al., 2004b; Landman et al., 2016; Li et al. 2014; Li and Chow, 2015; Meng and Khoo, 2010; Meshkat et al., 2015; Papamichail et al., 2010; Reilly et al., 2015; Shen and Zhang, 2010; Zhang and Wang, 2013). To enhance the traffic control effect, there exist research that integrated ramp metering with variable speed limits for the mainline traffic (Carlson et al., 2010; Carlson et al., 2014; Li et al., 2014; Lu et al., 2011). However, all of these researches were only concerned with the traffic condition along a highway stretch or inside the freeway loop specifically studied, but did not consider possible traffic diversions onto other freeways or local road networks, which could result in new network-wide traffic equilibrium states. It is true that ramp meters have limited capability to impact the network-wide equilibrium because of the limited queuing space for vehicles waiting on ramps, and the attempt to avoid the vehicle queue spilling back to the local roads. Howbeit, the deployment of ramp meters and coordinated ramp metering still would cause traffic diversions and change the equilibrium, because travel times on ramps do change.

To evaluate the improvement of network-wide traffic condition through ramp metering, a number of empirical studies have been conducted by comparing travel time data between the time periods when there was ramp metering and when there was not (Faulkner et al., 2014; Levinson and Zhang, 2006; Osman et al., 2015; Xie et al., 2012; Zhang and Levinson, 2010). Results showed that ramp metering might not necessarily lead to travel time reductions in every case, but it did reduce travel time variation. Besides these empirical studies, Zhang (2007) used an unconventional positive approach to model how travelers adapted their routing to the deployment of ramp meters, and explored traffic diversion as an emergent process on a large network with travelers' routing processes individually traced. By characterizing the behavior of the cell transmission model for a freeway with on-ramps and off-ramps, Gomes et al. (2008) investigated the traffic equilibria with and without ramp metering, and showed that congestion could be eliminated by ramp metering. But that research still only focused on a freeway stretch and did not consider network-wide OD flow diversions.

2.4.2 Manage Network Flows through Toll Imposition. Compared to ramp metering, imposing tolls on some or all of the links, has been more commonly studied as an efficacious way to drive user optimized flow pattern towards system optimum in literature. In the case of tolls being charged for trips made, the traffic equilibrium is a bi-criterion equilibrium since users have to select among paths based on not only the travel time, but also the tolled amount of the path. Value of time (VOT) has been introduced to describe travelers' tradeoff between monetary cost and travel time in response to toll charges. It combines the monetary and temporal cost of a path as the generalized path cost. According to the well-established first-best congestion pricing theory, the system optimum flow pattern can be achieved by charging a toll on every link of the network, the amount of which equals to the difference between the marginal social cost and the marginal private cost (Dafermos and Sparrow, 1971). In reality, charging tolls on all of the links are not applicable since the cost of toll collection over the entire network is prohibitively expensive. To manage network flows with tolls in a more practical way, some research developed models and solution approaches based on the second-best pricing scenario (Verhoef, 2005; Yang and Zhang, 2002; Yang and Zhang, 2003), where only a subset of links were subject to toll charges.

Under the presumption of homogeneous users, VOTs are identical for all travelers. With a single VOT, the minimization of total travel time and the minimization of total amount tolled result in the same traffic flow pattern. If link travel time function is separable and monotonic (Yin and Yang, 2004), the marginal-cost optimal toll can be simply obtained as $\beta v_i t_i'(v_i)$, where β is the VOT, v_i is the flow on link i, and $t_i'(v_i)$ is the first derivative of travel time function evaluated at v_i. However, due to travelers' demographic differences, they must have different VOTs. To incorporate the VOT differences among travelers, the commonly used approach in previous research is to group travelers into different classes, and assign each class a different VOT.

In the cases of multiple user classes with distinct VOTs, the system optimum flow pattern distinguishes between the minimization of travel time and the minimization of tolled amount (Yin and Yang, 2004). In spite of that, Yang and Huang (2004) showed that the same multi-class network equilibrium flow could be obtained, whenever the generalized travel cost was measured in cost or time. They demonstrated that the uniform link toll pattern, which supported a multi-class user equilibrium as a cost-based system optimum, could be obtained by multiplying the user externality of travel time by the arithmetic mean of the VOTs of all the users traversing that link. The user externality is the additional travel time that a marginal user imposes on others already traveling on link. They also showed that the uniform link toll, which supported a multi-class user equilibrium as a time-based system optimum, could be determined from the solution of a linear dual problem, and the toll could be either a charge or a subsidy to link users. Other research on this topic handled the bi-criteria optimization either through devising mechanisms to integrate the travel time and travel cost (Marcotte and Zhu, 2009; Wang and Ehrgott, 2013; Yang and Zhang, 2008; Zhang et al., 2008b), or by investigating the Pareto-optimality with respect to tolls and travel time (Song et al., 2009).

Besides the case of multi-class users with different VOTs, the network equilibrium with tolls has been studied in other scenarios, such as stochastic user equilibrium (Liu et al., 2014; Meng and Wang, 2008; Yang, 1999), joint route and departure time choice in dynamic traffic network (Joksimovic et al., 2005), and tolls being imposed step-by-step (Chen et al., 2015; Guo, 2013). Although rich literature is found regarding various extensions of the equilibrium problems in networks with tolls, research based on data collected from real world, which estimates VOTs for users of different classes and investigates the impact of charging tolls on travelers' routing decisions, is scarce. In many real-world networks with tolled links/lanes, the usage of the tolled links/lanes are often overestimated. Bao et al. (2015) introduced the concepts of mental account and mental budgeting into travelers' rout choice process, and studied the reason of the overestimated usage of tolled roads. They found that travelers with low and moderate out-of-pocket travel budget perceived a much higher generalized travel cost than the actual cost on the tolled roads, and that caused the usage being overestimated by conventional equilibrium models for networks with tolls.

Although theoretically network flow management through toll imposition can drive user equilibrium flow pattern towards the system optimum, how to establish VOTs for different road users and how to design the tolls (i.e., which road to impose tolls and what amount to be tolled), are still research problems to be investigated.

2.4.3 Manage Network Flows through Link Speed Limit Imposition. Speed limits are usually imposed on roads to enhance safety and sometimes to reduce fuel consumption, emissions and noise (Yang et al., 2012). When there is no speed limit imposed, the speed-flow relationship and travel time-flow relationship can be described using the two graphs in Figure 2.4.3-i, where C is the link capacity:

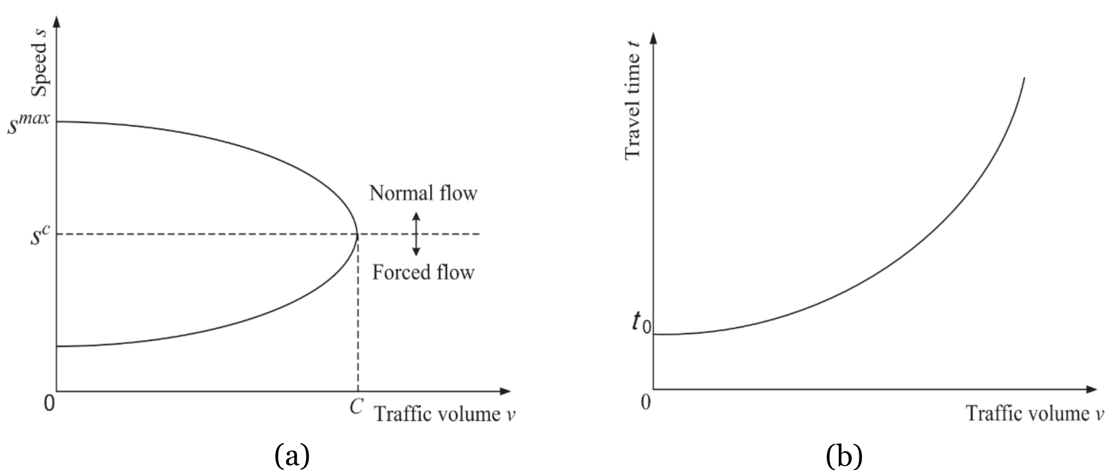
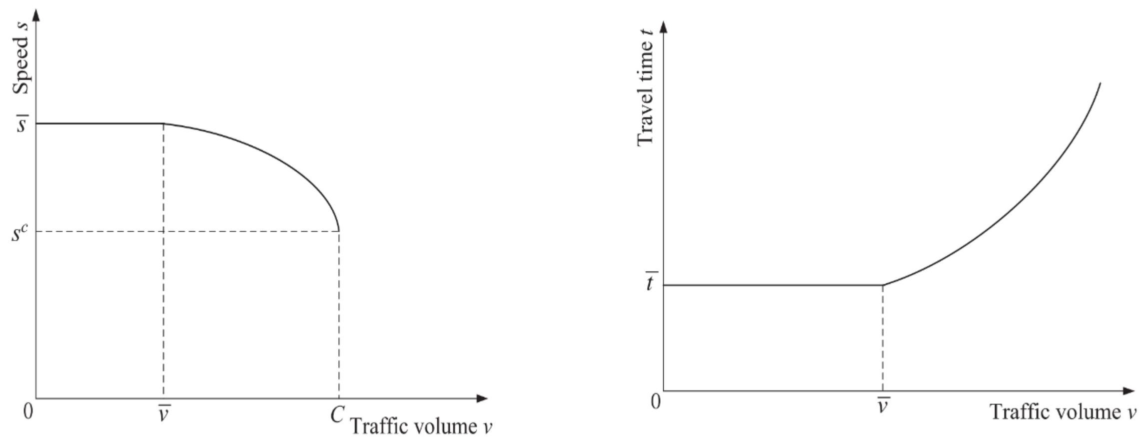


Figure 2.4.3-i: Speed-Flow (a) and Travel Time-Flow (b) Relationship without Speed Limits (Yang et al., 2012)

In the case when speed limit is enforced on the link, the speed-flow and travel time-flow relationships are illustrated in Figure 2.4.3-ii, where $\bar{s}$ is the imposed link speed limit:



(c) (d)

Figure 2.4.3-ii: Speed-Flow (c) and Travel Time-Flow (d) Relationship with Speed Limits (Yang et al., 2012)

Imposing speed limits will inevitably impact travel times and mobility, and eventually would result in the network-wide reallocation of traffic flows. Yang et al. (2012) studied the uniqueness of link travel times and flows at user equilibrium with link-specific speed limits, and investigated how link speed limits impact network level traffic equilibrium macroscopically. They also compared the capability of speed limits with toll charges on traffic reallocation effects, and concluded that a speed limit law could regulate traffic flows as well as a toll charge scheme and performed better than some rebate toll schemes under certain conditions. Yang et al. (2015) extended the research by modeling the speed choices of heterogeneous travelers, which were determined by subjective travel time cost, the perceived crash risk and the perceived ticket risk on each link in uncongested condition. In their research, different user classes interact with each other and choose their own optimal speed on particular roads, resulting in a Nash equilibrium speed pattern. And then based on the speed choices, travelers make route choices and eventually reach the user equilibrium.

2.5 Conclusion

In tactical level of maintenance planning, the length of the time period when maintenance projects are being worked on is comparable to the length of the planning horizon. And the temporal network capacity reductions caused by maintenance activities and its impact on network-wide traffic diversions have to be considered. This induces the network capacity management problem of scheduling the maintenance so that flows are not overly affected by the mandatory temporal network capacity changes. It is a problem that has been investigated in very few literatures and will be addressed in the research presented.

In the cases when the total flow cost at user equilibrium (UE) is used to evaluate the network on its capability of fulfilling flow demands, increasing network capacity might not always be the solution to alleviating congestions. As an application of the well-known Braess' Paradox, network capacity management strategies can be developed to selectively reduce the capacity or increase the generalized travel cost for some of the links, so as to drive the traffic flow pattern towards more efficient equilibrium states. Most previous research that studied the network capacity management mechanisms considered isolated link capacity controls, and only analyzed the impacts of traffic controls on local traffic flows. They lacked the systematic perspective to consider the coordinated network capacity control and network-wide traffic diversions. Hence, this dissertation will make the first attempt to resolve this problem in the network level.

In the reviewed literature that studied managing mandatory network capacity changes, network maintenance strategies were evaluated by a single type of network flows model. However, due to the heterogeneity of multi-modal traffic in urban transportation networks, travelers choosing different travel modes may require disparate network flows models to evaluate a maintenance plan. To give an example, regular cars are the major users of the city road network, and their user-optimized routing pattern requires traffic equilibrium models to evaluate the impact of maintenance activities. Compared to regular cars, autonomous vehicles are equipped with the technology to decide its route without the interference of riders, and is a new travel mode that will be available in the near future. And thus, this new travel mode is expected to play an important role in reducing traffic congestion by taking routes that minimize the total travel time of all travelers with some incentives. Hence the autonomous vehicle flows can be modeled as the system optimum (SO) flows. Enlightened by this vision, investigating the optimal maintenance planning for a mixture of traffic flows with different routing objectives is another aspiration of this dissertation.

Chapter 3

MAINTENANCE SCHEDULING IN NETWORKS OF SERVICE VEHICLES (MS-NSV)

3.1 Introduction

Although service vehicles (i.e., commercial trucks) are not the major users of the city transportation network, they are always one of the travelers' and city planners' major concerns because of their large sizes, heavy weights, and enormous fuel consumption and emission. Besides service vehicles, temporal changes on the transportation network, which are resulted from work zones, also induce negative impacts on traffic flows. Since work zones reduce visibility and mobility, they reduce road capacity and safety significantly. Hence, it is not surprising to see that the combination of service vehicles and work zones exacerbates the traffic condition -- although large trucks accounted for only 4% of all registered vehicles in the United States, 27% of work zone fatal crashes involved at least one large truck (FWHA, 2013).

In the presence of several work zones that are spatially close to each other, traveling through work zones one after another is stressful. These work zones cause extensive traffic delays and compound safety concerns, especially for service vehicles because of their large sizes and heavy weights. It would be ideal if work zones could be scheduled one after another so that only one work zone is active at any point of time. However, due to the budget and resource limitations, a common completion deadline is usually imposed on a group of work zones. And thus, the investigation of how to schedule multiple work zones, subject to a common due date, and with considerations of network-wide origin-destination (OD) flow routing of service vehicles, is of great benefit to all the road users.

The research presented in this chapter treats the traveling cost of a link as the cost in general sense, which can be interpreted as combinations of travel time, monetary cost, and road unsafety. The total link traveling cost is designed to be piece-wise linear with respect to the number of service vehicles using that link, so that the expensive extra flow cost will be incurred if the available link capacity is exceeded. The piecewise linear cost function approximates the nonlinear relation between the traffic delay and unsafety, and the number of service vehicles traveling on that road. A mixed integer linear programming model is formulated to schedule work zones subject to a common deadline and OD demand of service vehicles. A randomized fix-and-optimize heuristic is developed to solve the model efficiently and tested with different networks.

3.2 MS-NSV Model

3.2.1 Piecewise Linear Cost Structure. In networks with service vehicle flows, linear flow cost structure is commonly used, where the cost of travelling on a link is set linear with respect to the total flow amount on that link when the amount of flows is smaller than or equal to the available capacity of the link. In applications where the demand on a link is more than the available capacity, the excess flow is either detoured or given a very high cost for using the link thereby circumventing the hard capacity constraint. In this chapter we will use the latter approach by modeling the link cost function piece-wise linear, so as to approximate the traffic condition aggravation effect in service networks. With the piece-wise linear cost functions, the work zone scheduling model developed later can be solved by commercial solvers like CPLEX, the performance of which can be used to compare with the new heuristic developed later in the chapter.

In the work zone scheduling model, it is assumed that there are Origin-Destination (OD) flow demands of service vehicles (e.g. trucks) every time period (e.g., peak period of a day). Each service vehicle can choose its own route to minimize its travel cost, and is treated as a unit of flow. In this chapter, we assume the minimum scheduling unit of a work zone is a lane of a link regardless of its length. When a link is under maintenance, one or more lanes are closed and this leads to the temporary link capacity reductions. That is likely to cause the current flow on the link to exceed the available link capacity, incurring the expensive extra flow cost. The available link capacity can be interpreted as the threshold of the traffic condition degradation effect. When the number of service vehicles on the link is smaller than the available link capacity, the traffic condition worsens at a relatively slow rate. However, if the number of service vehicles traveling on the link exceeds the available link capacity, the traffic condition degradation effect will have a qualitative change, and each additional service vehicle on that link will worsen the traffic condition much more severely. The threshold (available link capacity) is designed to be positively related to the number of lanes open to serve the traffic flows. For example, for a link with multiple lanes, if the threshold is u when a link only has one lane open, then the threshold becomes $2u$ when two lanes of the link are open.

Suppose a link has three lanes and all three lanes have the same "flow capacity" u, Figure 3.2.1-i on the next page illustrates the relation between the flow units traveling on the link in a time period and the total flow cost in different lane closure scenarios. When two lanes are closed for maintenance, the available capacity of the link is u. If the units of flows using the link are more than u during the time period, then the extra flow cost will be incurred. This is why the slope of the cost curve is much steeper when the flow units are more than u for the two-lane closure case. Similar, cost curve pattern can be observed in the cases of no-lane closure and one-lane closure. When some of the lanes in a link is closed for maintenance, some of the flows that are originally on this link may divert to other links to reach the destination with lower total cost, and that means the network flows are *reactive* to the maintenance schedules.

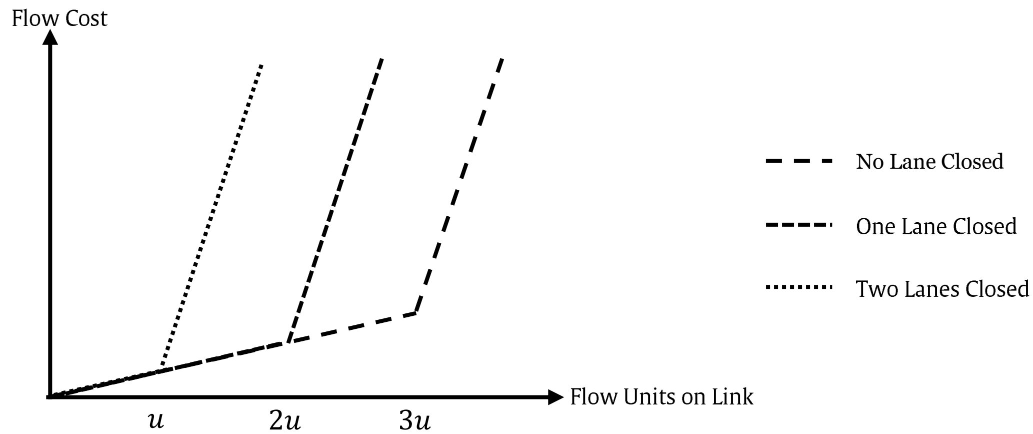


Figure 3.2.1-i: Three-Lane Link Flow Cost Curve

3.2.2 Model Formulation. The MS-NSV model possesses the features of both scheduling models and multi-commodity flows models. The objective of the model is to schedule the lane closures so that all links that need maintenance are repaired before a given completion date for the whole network, while the total flow cost for all the OD pairs, which includes regular flow cost and extra flow cost, is minimized over the project period. This section describes the MS-NSV model in detail.

Denote c_i as the regular unit flow cost of link i, y_{ikt} as the flow units of OD pair k that flow through link i on day t, and z_{it} as the difference between flow units of all the OD pairs that flow through link i and the available capacity of link i on day t, the objective function is formulated as $min \sum_{i \in E}\{\sum_{t=1}^{t=T}[c_i * (\sum_{k \in OD} y_{ikt}) + z_{it}\rho c_i]\}$, where E is the set of links, OD is the set of OD demand, and T is the common completion date of all the maintenance work. ρ is the congestion flow cost multiplier which makes the extra unit flow cost ρc_i much larger than the regular unit flow cost c_i. The first part $\sum_{i \in E}\sum_{t=1}^{t=T}[c_i *$

$(\sum_{k\in OD} y_{ikt})]$ calculates the total regular flow cost for all the OD pairs on all the links over the project period, and the second part $\sum_{i\in E}\sum_{t=1}^{t=T} z_{it}\rho c_i$ calculates the total congestion flow cost for all the links over the project period. Both y_{ikt} and z_{it} are non-negative continuous variables. Note that z_{it} is non-negative in the sense that it will have positive value only when the total flow units on link i exceed the available capacity and it will be zero otherwise.

Binary variables s_{imt} are introduced as the flag variables indicating whether the repair of the m^{th} lane of link i starts on day t, and $s_{imt} = 1$ if it is. The MS-NSV model assumes once a lane is closed for repair, it cannot open to serve the flows until its repair is completed. Hence we have the constraints $\sum_{t=1}^{t=T} s_{imt} = 1$ for $\forall i \in R$ and $\forall m \in [1, n_i]$, where R is the set of links that need repair and n_i is the number of lanes in link i. This set of constraints force every lane of all the links that need repair to have one and only one repair start date.

To indicate whether m^{th} lane of link i is closed for maintenance on day t, binary variables x_{imt} are added to the model. x_{imt} equal to 1 if the m^{th} lane of link i is closed for maintenance on day t. Let p_i be the number of days needed to repair a lane of link i, we formulate the constraints $\sum_{t=1}^{t=T} x_{imt} = p_i$ for $\forall i \in R$ and $\forall m \in [1, n_i]$ to ensure the repair on all the links be completed by the common completion date T. Since each lane of the links needing maintenance have one and only one repair start date and the number of days needed to repair a lane is given, whether a lane is closed or not on a day is determined once the repair start date of that lane is determined. And thus, we develop the set of constraints $x_{imt} = \sum_{a=max(t-p_i+1,1)}^{a=t} s_{ima}$ for $\forall i \in R, \forall t \in T$ and $\forall m \in [1, n_i]$ to make sure that once a lane is closed for repair, it will not open to serve the flows until the repair work on this lane is finished and that it will be open on other dates. Constraints $\sum_{t=1}^{t=T} s_{imt} = 0$

for $\forall i \notin R, \forall m \in [1, n_i]$ and $\sum_{t=1}^{t=T} x_{imt} = 0$ for $\forall i \notin R$ and $\forall m \in [1, n_i]$ are added to the model so that all the lanes of links that do not need repair will not have maintenance start date and will be open to serve the flows throughout the project period.

For each OD pair on each day, flow conservation constraints, consisting of three groups, are needed. The first group of constraints makes sure the total incoming flow units minus the total outgoing flow units equal to the OD demand for the origin node of the OD pair. Let D_k be the demand of OD pair k, the first part is formulated as $D_k = \sum_{\{i:E_i^- = OD_k^-, i \in E\}} y_{ikt} - \sum_{\{j:E_j^+ = OD_k^-, j \in E\}} y_{jkt}$ for $\forall k \in OD, \forall t \in [1,\ T]$, where OD_K^- is the origin node of OD pair k, E_i^- is the head node of link i and E_j^+ is the tail node of link j. The second group ensures the total outgoing flow units minus the total incoming flow units equal to the demand of OD pair k for its destination node and is formulated as $D_k = \sum_{\{i:E_i^+ = OD_k^+, i \in E\}} y_{ikt} - \sum_{\{j:E_j^- = OD_k^+, j \in E\}} y_{jkt}$ for $\forall k \in OD, \forall t \in [1,\ T]$, where OD_K^+ is the destination node of OD pair k, E_i^+ is the tail node of link i and E_j^- is the head node of link j. For the rest of the nodes, other than origin and destination nodes of OD pair k, the total incoming flows on the node from the origin of OD pair k should equal to the total outgoing flows from the node to the destination of the OD pair k. This is the third group of the flow conservation constraints and it is formulated as $\sum_{\{i:E_i^- = l, i \in E\}} y_{ikt} = \sum_{\{j:E_j^+ = l, j \in E\}} y_{jkt}$ for $\forall l \in N, \forall t \in [1, T], \forall k \in \{k: OD_k^- \neq l\} \cap \{k: OD_k^+ \neq l\}$, where N is the set of nodes in the network.

In addition, binary variables v_{imt} are introduced to calculate the increased lane capacities and v_{imt} equals to 1 if lane m of link i is repaired before day t, since it is obvious that when a segment of road is repaired, the road condition should be improved and the capacity should increase. Constraints $v_{imt} = \sum_{a=1}^{a=t-p_i} s_{ima}$, for $\forall i \in R, \forall m \in [1, n_i]$ and

$\forall t \in [p_i + 1, T]$ determine the values of v_{imt} by values of s_{imt}. In the constraints, the date ranges from $p_i + 1$ to T since the lane will be repaired and open to serve the flows on day $p_i + 1$ the earliest, because even if the maintenance starts on day 1, it would take p_i days to complete the repair work for this lane. Constraints $v_{imt} = 0$, for $\forall i \in R, \forall m \in [1, n_i]$ and $\forall t \in [1, p_i]$ make sure each lane of the links that need maintenance stay in the status of not repaired in the first p_i days. And constraints $v_{imt} = 0$, for $\forall i \notin R, \forall m \in [1, n_i]$ and $\forall t \in [1, T]$ force lanes of links that do not need repair stay in the status of not repaired throughout the project period.

Let θ be the percentage increase in lane capacity after the lane is repaired, and let u_i be the capacity of a lane of link i, the available capacity of link i on day t is $\left(n_i - \sum_{m=1}^{n_i} x_{imt} + \sum_{m=1}^{n_i} \theta v_{imt}\right)u_i$. Hence the values of z_{it} are determined by constraints $\sum_{k \in OD} y_{ikt} - \left(n_i - \sum_{m=1}^{n_i} x_{imt} + \sum_{m=1}^{n_i} \theta v_{imt}\right)u_i \leq z_{it}$ and $z_{it} \geq 0$ for $\forall i \in E$ and $\forall t \in [1,\ T]$, where $\sum_{k \in OD} y_{ikt}$ are the total flow units from all OD pairs on link i on day t. Because of the introduction of z_{it}, flows can exceed the available capacity. Hence it is needed to make sure there won't be flows on links with all lanes closed for maintenance, that is, entirely closed links cannot serve any flow. For this reason, the set of variables w_{it} are added into the model, the values of which equal to 1 if all the lanes of link i are closed on day t. Constraints $\sum_{k \in OD} y_{ikt} \leq \sum_{k \in OD} D_k \left(n_i - \sum_{m=1}^{n_i} x_{imt}\right)$ for $\forall i \in R$ and $\forall t \in [1,\ T]$ make sure when all the lanes of link i are closed on day t, link i does not serve any flows. $\sum_{k \in OD} D_k$ serves as a large number in this constraint and ensures flows from all OD pairs can use link i as long as it has at least one lane open. The sets, parameters and variables of the MS-NSV model are presented in Table 3.2.2-i :

Table 3.2.2-i: MS-NSV Notations

Term	Definition
Sets	
N	Node set of the network
E	The set of existing links in the network
R	The set of existing links that need to be repaired in the network, $R \subseteq E$
OD	The set of Origin-Destination pairs of flows
Parameters	
T	Completion date for all the maintenance work (the earliest start date of a work zone is Day 1)
n_i	Number of lanes of link i
u_i	Capacity of a lane of link i
c_i	The regular flow cost incurred by one unit flow on link i per day
p_i	The number of days needed to repair a lane of link i
ρ	Extra flow cost multiplier, ρc_i is the extra flow cost incurred by the available link capacity being one unit less than the flow on link i
θ	Percentage of lane capacity increased after maintenance
D_k	Flow demand of OD pair k
Variables	
s_{imt}	Binary variable indicating whether to repair the m^{th} lane of link i starts on day t. If repair work starts on day t, $s_{imt} = 1$; otherwise, $s_{imt} = 0$
x_{imt}	Binary variable indicating whether the m^{th} lane of link i is closed for maintenance on day t, if it is closed, $x_{imt} = 1$; otherwise $x_{imt} = 0$
y_{ikt}	The flow units incurred by the Origin-Destination (OD) flow of OD pair k on link i on day t
z_{it}	Flow units on link i exceeding the available capacity of the link on day t. If the available capacity of link i on day t is less than the total flow units on link i, z_{it} equals to the difference between the available capacity and total flow on link i; otherwise $z_{it} = 0$

Term	Definition
Variables	
v_{imt}	Binary variable indicating whether the m^{th} lane of link i is repaired before day t, if it is, $v_{imt} = 1$, otherwise 0; for all the links that don't need maintenance, $v_{imt} = 0$ all the time

The complete model of scheduling work zones in networks of service vehicles (MS-NSV) can now be written as:

MS-NSV: $min \sum_{i\in E}\{\sum_{t=1}^{t=T}[c_i * (\sum_{k\in OD} y_{ikt}) + z_{it}\rho c_i]\}$ (1)

$\sum_{t=1}^{t=T} s_{imt} = 1, \quad \forall i \in R, \forall m \in [1, n_i]$ (2)

$\sum_{t=1}^{t=T} x_{imt} = p_i, \quad \forall i \in R, \forall m \in [1, n_i]$ (3)

$x_{imt} = \sum_{a=max(t-p_i+1,1)}^{a=t} s_{ima}, \quad \forall i \in R, \forall t \in T, \forall m \in [1, n_i]$ (4)

$\sum_{t=1}^{t=T} s_{imt} = 0, \quad \forall i \notin R, \forall m \in [1, n_i]$ (5)

$\sum_{t=1}^{t=T} x_{imt} = 0, \quad \forall i \notin R, \forall m \in [1, n_i]$ (6)

$D_k = \sum_{\{i:E_i^- = OD_k^-, i\in E\}} y_{ikt} - \sum_{\{j:E_j^+ = OD_k^-, j\in E\}} y_{jkt}, \forall k \in OD, \forall t \in [1,\ T]$ (7)

$D_k = \sum_{\{i:E_i^+ = OD_k^+, i\in E\}} y_{ikt} - \sum_{\{j:E_j^- = OD_k^+, j\in E\}} y_{jkt}, \forall k \in OD, \forall t \in [1,\ T]$ (8)

$\sum_{\{i:E_i^- = l, i\in E\}} y_{ikt} = \sum_{\{j:E_j^+ = l, j\in E\}} y_{jkt}, \quad \forall l \in N, \forall t \in [1, T],$
$\forall k \in \{k: OD_k^- \neq l\} \cap \{k: OD_k^+ \neq l\}$ (9)

$v_{imt} = \sum_{a=1}^{a=t-p_i} s_{ima}, \quad \forall i \in R, \forall m \in [1, n_i], \forall t \in [p_i + 1, T]$ (10)

$v_{imt} = 0, \quad \forall i \in R, \forall m \in [1, n_i], \forall t \in [1, p_i]$ (11)

$v_{imt} = 0, \quad \forall i \notin R, \forall m \in [1, n_i], \forall t \in [1, T]$ (12)

$\sum_{k\in OD} y_{ikt} - (n_i - \sum_{m=1}^{n_i} x_{imt} + \sum_{m=1}^{n_i} \theta v_{imt})u_i \leq z_{it}, \quad \forall i \in E, \forall t \in [1,\ T]$ (13)

$\sum_{k\in OD} y_{ikt} \leq \sum_{k\in OD} D_k (n_i - \sum_{m=1}^{n_i} x_{imt}), \quad \forall i \in R, \forall t \in [1,\ T]$ (15)

$s_{imt}, x_{imt}, v_{imt} \in \{0, 1\}, \quad \forall i \in E, \forall m \in [1, n_i], \forall t \in [1,\ T]$ (16)

$z_{it} \geq 0, \quad \forall i \in E, \forall t \in [1, T]$ (17)

$y_{ikt} \geq 0, \quad \forall i \in E, \forall k \in OD, \forall t \in [1, T]$ (18)

3.3 Computational Implementation

The MS-NSV model is programmed in C++ with IBM® ILOG® CPLEX® Concert Technology. As a mixed-integer program that does not have unimodular coefficient matrix for the constraints that involve scheduling variables, the MS-NSV is unlikely to be polynomially solvable and cannot be solved by CPLEX within a tolerable amount of time. Using a computer of 3.7 GHz quad-core CPU and 24.0 GB memory for the computation work of a small problem instance with 16 nodes, 48 links, 108 lanes, 16 OD pairs, and 27 days to repair 50% of the links, CPLEX still has a 32% optimality gap after 14 hours of computation. Therefore, it is clear an efficient heuristic to solve the problem quickly with satisfactory accuracy is needed.

3.4 Solution Approach

3.4.1 Randomized Fix-and-Optimize (RFO) Heuristic. There are two levels of problems that constitute the problem of work zone scheduling in networks of service vehicles. The upper level is the scheduling problem which decides the repair start date for each lane of the links that need maintenance. The lower level is a series of multi-commodity flow problems based on the available capacities of links on each day, which is determined by the current lane closures. Once the schedule is set, solving the multi-commodity flow problems for each day is a relatively easy problem since the flow variables are all continuous variables. And thus the solution approach proposed in this chapter focuses on the upper level of obtaining good work zone schedules.

To motivate the heuristic, suppose at a point in the algorithmic process we obtain a feasible schedule that has some aspects similar to the optimal schedule. For example, Figure 3.4.1-i gives a comparison between the Gantt charts of the optimal schedule and one of the feasible schedules obtained for a small test network of 4 nodes, 12 links and 12 OD pairs. The vertical axis shows the lanes of links that need maintenance and the horizontal axis shows the date during the project period. Each bar represents the time period when a lane is closed for maintenance and cannot be used to serve the OD flows. For example, in the optimal schedule, Lane 1 of Link 2 is closed on Day 1 and will be reopen on Day 8, and Lane 2 of Link 2 will be closed from Day 7 to Day 13. Hence this two-lane link will have one lane available from Day 1 to Day 6 and from Day 8 to Day 13. On Day 7 Link 2 is not available to serve any flows since both of the two lanes are closed.

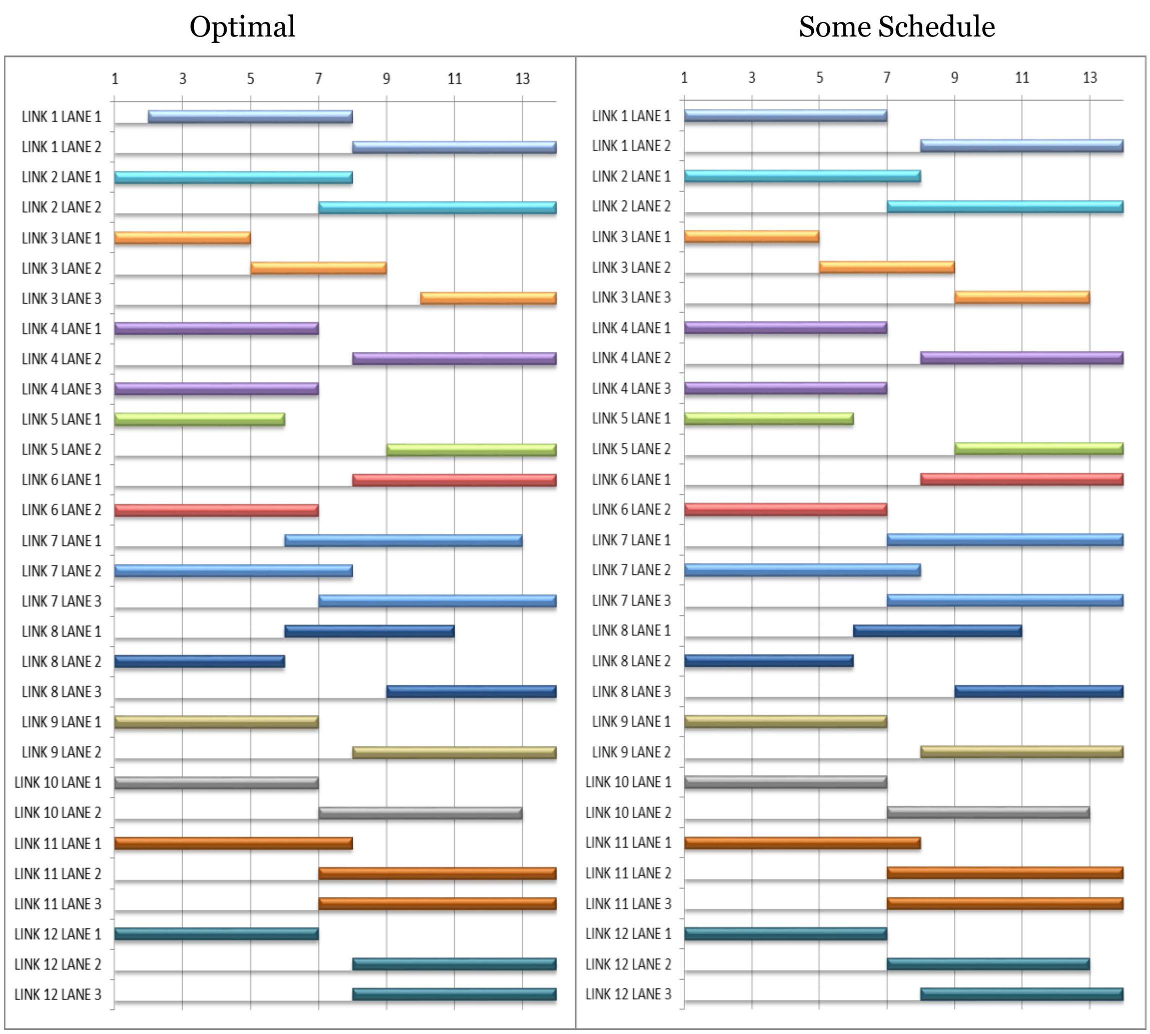


Figure 3.4.1-i: Schedule Comparison

From the Gantt chart we can see that the feasible schedule has lane closures of Link 1, 3, 7, and 12 different from the optimal schedule. If we only optimize the lane closure schedules of these four links and fix the schedules of all the other links, the problem size will be much smaller and the time needed to solve the problem instance will reduce dramatically since there are much fewer integer variables to go through in the branch-and-bound process performed by solvers like CPLEX. This observation leads to the adoption of the *fix-and-optimize* heuristic as the core of the solution approach.

The fix-and-optimize heuristic was first introduced by Helber and Sahling (2010). It is an iterative optimization-based heuristic developed to solve the multi-level capacitated lot sizing problem which is a mixed-integer program. The basic process of the fix-and-optimize heuristic is to partition the integer variables into subsets, based on an initial solution, and then optimize the values of a subset of integer variables together with all continuous variables while the values of the other integer variables in other subsets are fixed (this is called a subproblem of the fix-and-optimize procedure). If the new objective function value is better than current best objective value, then the current candidate optimal values are updated; iterate this process for other subsets of variables until a specified stopping criteria is met. The percentage of integer variables in each subset of all the integer variables ranged from 0.5% to 10% based on the difficulty and size of the problem instances tested in Helber's paper. For each specific problem instance, the number of integer variables in a subset was fixed. Also, the integer variables were decomposed into subsets based on the descending order on cost of each product in the lot-sizing problem, since usually a quite reasonable schedule was found after the first round of the product-oriented decomposition.

In the problem of scheduling work zones in networks of service vehicles, the relation among work zones is more complex than that among products in the capacitated lot-sizing problem. Products just compete with each other for resources (machine hours) in the capacitated lot-sizing problem. On the other hand, in the MS-NSV problem there are no resource constraints that work zones compete for, but instead the work zones affect the capacity of the network to serve the OD demands which in turn compete for this capacity. Therefore, only the schedules that consider all or many work zones will have the lowest increase in total flow cost, because OD demands happen over the whole network and each OD pair has network-wide minimum cost routing. This means applying fix-and-optimize heuristic with small subsets of work zones (one or two links) will hardly find satisfactory schedules since it is only considering the maintenance of a few links at a time.

However, if the size of the work zone subsets is large, the size of each fix-and-optimize subproblem will also be large and it would take long time to solve. To mitigate the conflict between solution quality and solving time length, we develop the fix-and-optimize procedure with varying subset sizes and use a truncated branch-and-bound method.

Initially, CPLEX tries to solve the entire problem within a given time limit (e.g. 60 seconds). If the problem is solved optimally, then the optimal schedule will be output and the program will terminate. If the problem is not solved optimally, the best feasible schedule obtained so far will be stored and used as the initial feasible solution for the fix-and-optimize procedure. A feasible schedule should be able to both complete all the maintenance work before the specified completion date and make sure each OD pair won't be disconnected because of possible entire-link closures throughout the project period. This situation of disconnecting an OD pair is likely to happen when large portion of links need to repair within a very short project period. To meet the maintenance completion deadline, the time windows of many work zones may overlap which could lead to many links being entirely closed at the same time, and this may result in no path can be found for one or more OD pairs. If no schedule can meet the completion deadline and the OD flows requirements at the same time, then the preset project completion deadline is too tight and needs to be extended to obtain feasible schedules.

The randomized fix-and-optimize (RFO) iteration starts with randomly dividing links that need maintenance into two subsets and solving each fix-and-optimize subproblem (FO subproblem) with a specified time limit. A RFO iteration is finished when the schedules of all the generated subsets of links are optimized. The RFO will be performed for a preset number of iterations and if any of the FO subproblems is not solved within the time limit in the last iteration, the RFO will enter a new stage where the number of subsets which the links to repair are randomly divide into is three. The RFO proceeds similarly in stages with more subsets of links and each RFO iteration is performed the same way as it is in the initial stage when there are only two subsets.

The reason of randomly grouping links that need maintenance into subsets is because we do not know the set of links with schedules that are different from the optimal schedule since we do not have the optimal schedule. Also, consideration of various OD demand patterns, and flows being reactive to network capacity changes, makes it formidable to pin-point the links that can have better schedule through classical network flows optimization models. Hence random grouping is applied to explore various combinations of links for better schedules. Both the decomposition of the links based on the required number of days to repair and decomposition based on links' unit flow cost are tested, but both of them have inferior performance compared to the random grouping approach. Through the iterative randomized fix-and-optimize process, the work zone schedule change gradually towards the optimal schedule.

The detailed procedure of RFO is summarized on the next page:

Randomized Fix-and-optimize Heuristic

---

1. Solve the entire problem with time limit $timeLimitSV$

   If optimal solution obtained, proceed to Step 4

   Otherwise store the best feasible schedule and objective value, and go to Step 2

2. Set number of subsets $numBat = 2$

3. Randomly divide links to repair into $numBat$ groups

   3.1. Fix (v, s, x, w) for links in $numBat - 1$ groups, $LonSolTime = 0$, set iteration number $iterNum = 1$

   3.2. Solve the FO subproblem with time limit $timeLimitFO$ for the subset (n) of links the (v, s, x, w) values of which are not fixed

   If optimal solution is not obtained in $timeLimitFO$ proceed to Step 3.2.1

   3.2.1. Store the current best feasible schedule and objective, and set $LonSolTime = 1$

   Otherwise directly proceed to Step 3.3.

   3.3. If the objective obtained in current FO subproblem is lower than the best objective of the FO subproblems obtained so far ($TotalCostFO$), update the $TotalCostFO$ and the schedule of links in subset n

   Otherwise directly proceed to Step 3.4

   3.4. Check whether there are subsets of links of which the FO subproblems are not solved

   If there are, proceed to Step 3.4.1.

   3.4.1. Choose one of the subsets to be subset n and go back to Step 3.1

   Otherwise proceed to Step 3.4.2

   3.4.2. If $TotalCostFO < TotalCost$ (best objective overall), proceed to Step 3.4.2.1

   3.4.2.1. Update the value of $TotalCost$ with the value of $TotalCostFO$, increase $iterNum$ by 1, go back to Step 3

   Otherwise proceed to Step 3.4.2.2

   3.4.2.2. If $iterNum < iterLimit$, proceed to Step 3.4.2.2.1

   3.4.2.2.1. Increase $iterNum$ by 1, go back to Step 3

   Otherwise proceed to Step 3.4.2.2.2.

   3.4.2.2.2. If $LonSolTime = 1$, proceed to Step 3.4.2.2.2.1

   3.4.2.2.2.1. If $numLinpBat > 3$ proceed to Step 3.4.2.2.1.1

   3.4.2.2.1.1. Increase subsets number $numBat$ by 1, set iteration number 1, go back to Step 3

   Otherwise proceed to Step 4.

   Otherwise proceed to Step 4.

4. Output the best schedule and flows obtained

The flow chart of the RFO is displayed blow:

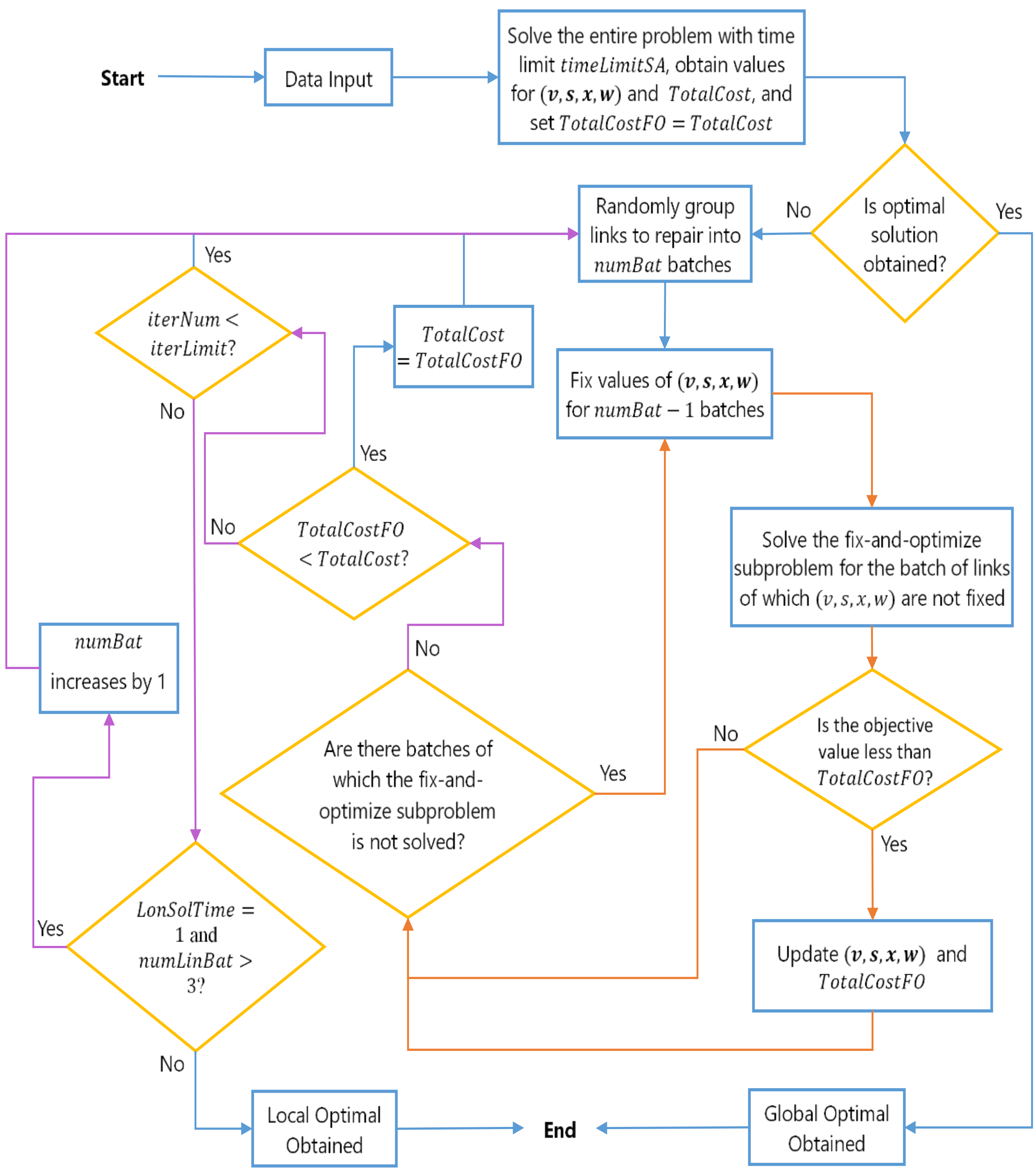


Figure 3.4.1-ii: Flow Chart of RFO

3.4.2 Parameters Affecting the Performance of RFO. The randomized fix-and-optimize heuristic has two levels of computation procedures. The first level randomly decomposes the links that need maintenance into a specific number of subsets and the second level optimizes the repair schedules of each link subset with the schedules of links in other subsets fixed (FO subproblem) within a specified time limit. Hence the efficiency of RFO heuristic is mostly determined by two parameters: the number of iterations RFO performs for a specific number of groups which the links to repair are randomly partitioned, and the time limits for the initial attempt on solving the entire problem and for the attempts on each FO subproblem.

More RFO iterations means that the heuristic can solve FO subproblems for more combinations of links to repair for a specific subset size and is more likely to obtain better feasible solutions with objectives that are closer to the optimal solution. However, after a considerable amount of experimentation, we found that increasing the number of iterations does not effectively improve the solution quality. This is because there are too many possible combinations of links to repair for any specific subset size, and the chance is little that the links, which have schedules different from the optimal schedule, are in the same subset through random decomposition. Fewer subsets with more links in each subset can increase the chance of grouping together the links with repair schedules different from the optimal schedule. However, the time needed to find better schedules for each FO subproblem will be longer since now the FO subproblem has large number of integer variables. Thus, performing large number of iterations with fewer subsets with many links in one group will either result in poor solution quality with low time limit for each FO subproblem, or result in very long solving time with high time limit for each FO subproblems. As default values, we set the number of iterations the same as the specified number of link groups (e.g. perform 2 RFO iterations when the number of groups is 2), and the Computational Experiments in next section will show the RFO gives good feasible solutions within reasonable amount of time.

We also need the time limits for the initial attempt on solving the entire problem and for attempts on each FO subproblem. Problem instances with a few work zones have less integer variables, and is more likely to obtain a feasible solution that is close to the optimal solution (solution with less than 5% relative optimality gap) in a short time during the initial attempt to solve the entire problem. For each FO subproblem, if there is a feasible schedule that is better than the current best feasible schedule, the solver should be able to find it very quickly since the FO subproblem has even less integer variables. As long as a feasible schedule is found that is better than the current best feasible schedule, it can be used as the initial schedule for the next RFO iteration. Increasing the time limit in this case is pointless since a better schedule is already found and increased time will be wasted on improving the lower bound to prove the solution is optimal for the FO subproblem or the entire problem.

As the number of work zones increases, the dramatic increases in the number of combinations of integer variables complicates the branch-and-bound process substantially. This makes it nearly impossible to quickly obtain a feasible solution that is close to the optimal solution in the initial attempt on the entire problem. Improving the quality of initial feasible solution through increasing the time limit is not wise since it is very likely that the relative optimality gap is still larger than 5% after hours of calculation. With an initial feasible solution which is not close to the optimal solution to start the RFO process, it would also be challenging for the solver to find feasible solutions that are much better than the current best feasible solution found in a short time in the FO subproblem. Therefore, increasing the time limit on solving the FO subproblem will be much more effective in finding better solutions since the FO subproblem has much fewer integer variables. And thus, both the time limits on the initial attempt on the entire problem and on the attempts on each FO subproblem should be relatively higher to allow the solver to spend more time on searching for better feasible solutions.

3.4.3 Computational Experiments. The randomized fix-and-optimize heuristic is tested on three representative networks: a radial network, a grid network, and the Sioux Falls network. For each network, the links that need maintenance are randomly selected based on the preset percentage of links to repair. For each network with the set of links to repair selected, test cases vary by the parameter T, which is the completion date for all the maintenance work. The extra flow cost multiplier ρ is set to 10000 and the percentage of lane capacity increase after repair θ is set to 20% for all the test cases. The computer used to run these tests cases is the same computer mentioned in Section 3.3.

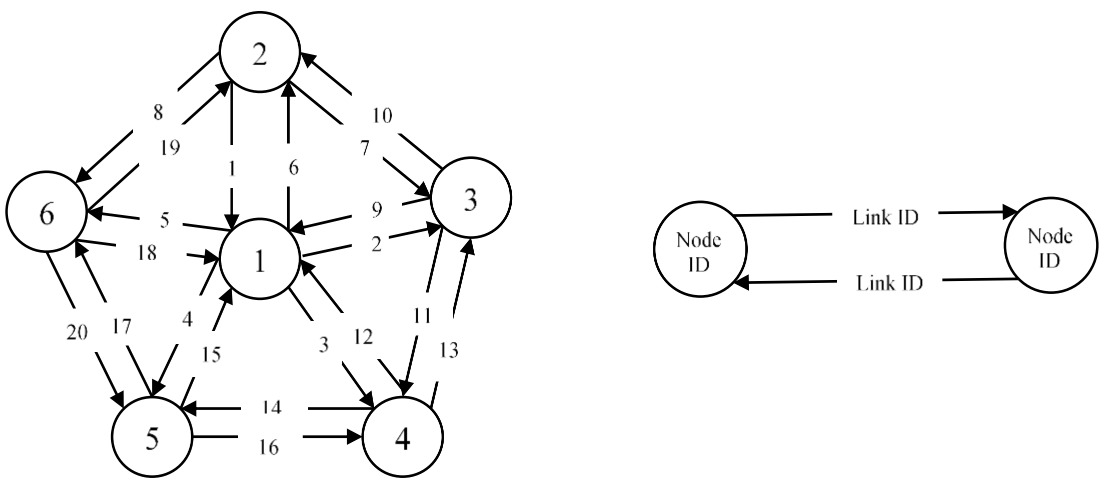


Figure 3.4.3-i: Radial Network

We begin the test on the heuristic designed with a radial network. Radial transportation network structure is commonly found in large cities with long history like London and Paris. The radial network tested is a small network with 6 nodes, 20 links and 20 OD pairs (network is shown in Figure 3.4.3-i). Among the 20 links, 10 are randomly selected as the links that need maintenance resulting in a total number of 30 work zones to be scheduled (since a link has multiple lanes and each lane is an independent work zone). The time limits for solving the entire problem initially and for each FO subproblem are both 60 seconds. The performance comparison between solving the test cases by randomized fix-and-optimize heuristic (RFO) and solely by CPLEX is shown in Table 3.4.3-i.

Table 3.4.3-i: RFO VS CPLEX on Radial Network

Completion Date *(T)*	Solving Time		Objective Value		Objective Value Difference
	RFO	MIP	RFO	MIP	
12	1.89 sec	1.89 sec	489892	489892	0.00%
13	4.37 sec	4.37 sec	404316	404316	0.00%
14	10.70 sec	10.70 sec	318741	318741	0.00%
15	1.53 min	29.75 min	233166	233166	0.00%
16	3.69 min	>14.87 hr.	170591	170591 (UB) 167322 (LB)	0.00%(UB Gap)
17	6.13 min	>40.82 hr.	101516	101516 (UB) 92039 (LB)	0.00%(UB Gap)
18	6.12 min	>2.69 hr.	25833	25645 (UB) 573 (LB)	0.73%(UB Gap)
19	7.03 min	>2.54 hr.	19188	19264 (UB) 6762 (LB)	0.40%(UB Gap)
20	7.39 min	> 15.73 hr.	10189	9790 (UB) 3320 (LB)	4.07%(UB Gap)
26	4.62 sec	4.62 sec	623.34	623.34	0.00%
36	49.79 sec	49.79 sec	856.62	856.62	0.00%
46	1.88 min	1.07 hr.	1090.17	1090.17	0.00%

For the solving time of CPLEX that has ">", it means CPLEX is not able to solve the test case optimally after a long time and the solving process is terminated manually with the best upper bound and lower bound obtained recorded. The upper bound is the objective value of the best feasible solution obtained at the time of terminating the solving process. The optimality gap is calculated as the objective obtained by RFO minus the objective (or upper bound if solving process is terminated manually) obtained by CPLEX and divide the difference by the objective (or upper bound) obtained by CPLEX. These result display formats are the same for the illustration on the experiments on the grid network and Sioux Fall network later.

The solving time of RFO and CPLEX for some test cases are the same because CPLEX was able to solve the entire problem in 60 seconds and the randomized fix-and-optimize procedure did not start. Since the grouping of links that need maintenance is random for each RFO iteration, the time needed to solve the same test case for each run will be different and the best solution obtained in each run may also be different from each other. We run RFO to solve each test case that are not solved optimally by CPLEX in 60 seconds for five times, take the average of the solving times and the objective values from the five runs, and compare them with the objective and solving time of CPLEX. The objective values and solving times of five runs of each test case are listed in Appendix A.

From Table 3.4.3-i we can see that even for a 20-link radial network with 50% of the links need maintenance, CPLEX is not able to solve some of the test cases in tolerable amount of time. Also, the RFO heuristic is able to obtain optimal or near-optimal solutions within little amount of time compared to CPLEX. Notice that for the test case when $T = 19$, the objective value from RFO is better than the best feasible solution obtained by CPLEX. To obtain the best feasible solution of this test case, RFO takes less than 7 minutes and the solution dominates the best feasible solution from CPLEX after nearly 3 hours of computation.

A larger network tested is a grid network with 16 nodes, 48 links and 24 OD pairs (network is shown in Figure 3.4.3-ii). Grid transportation network structure is frequently found in large modern cities like Phoenix and Vancouver, and their central business districts. The grid network tested also has 50% of links randomly selected as the links to be repaired and the total number of work zones to be scheduled is 52. The time limits set for solving the entire problem initially and for the FO subproblems are both 60 seconds. RFO is used to solve each test case for five times and the objective values and solving times for each solution run for each test case are listed in Appendix A. The comparison between the average performance of RFO and the performance of CPLEX is displayed in Table 3.4.3-ii below:

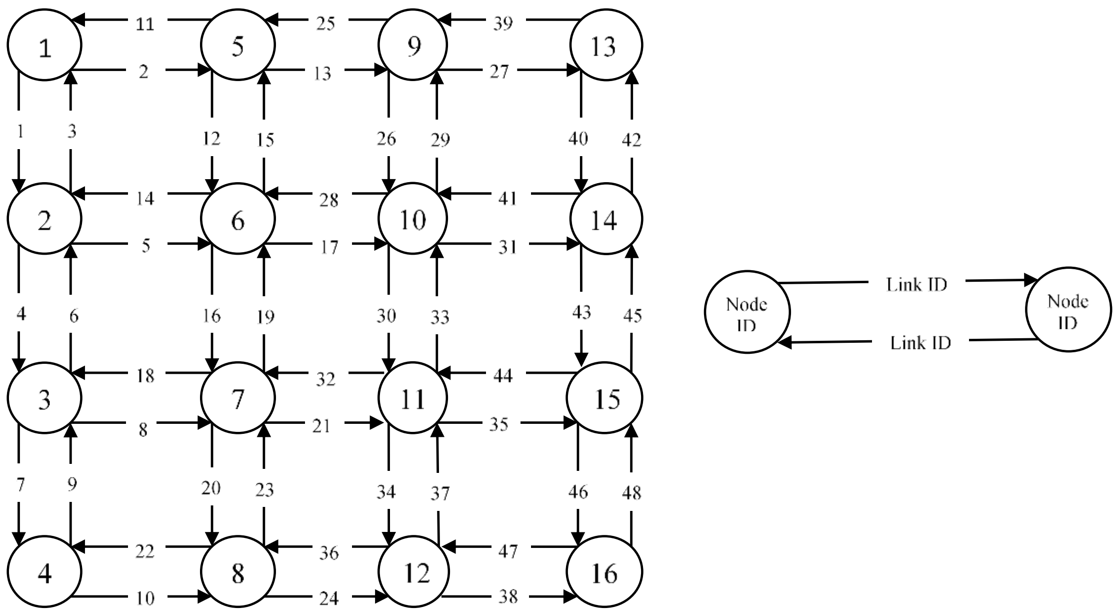


Figure 3.4.3-ii: Grid Network

Table 3.4.3-ii shows that RFO is much more efficient than CPLEX on solving the test cases of the grid network, especially when the test case is difficult to solve. And the solution quality of RFO is also quite good. Usually the percentage of links that need maintenance in a network won't be as much as 50%. The reason we set the percentage of links to repair 50% for the radial network and grid network tested is because we would like to show how difficult the MS-NSV problem can be and how efficient the RFO is compared to solving the test cases solely by CPLEX.

Table 3.4.3-ii: RFO VS CPLEX on Grid Network

Completion Date *(T)*	Solving Time		Objective Value		Objective Value Difference
	RFO	MIP	RFO	MIP	
12	52.21 sec	52.21 sec	255576	255576	0.00%
13	2.35 min	1.24 min	186740	186740	0.00%
14	4.488 min	25.78 min	143429	142525	0.67%
15	4.492 min	37.03 min	105997	105502	0.47%
16	3.596 min	27.36 min	67711.7	66209.3	2.06%
17	6.67 min	>14.28 hr.	51773.7	51771(UB) 37692(LB)	0.00%(UB Gap)
18	7.882 min	>1.23 hr.	37350	37344.6(UB) 25990.6(LB)	0.68%(UB Gap)
19	6.848 min	>13.39 hr.	26672.5	26666.25(UB) 19660.61(LB)	0.41%(UB Gap)
20	5.154 min	>3.98 hr.	15988.9	15988.21(UB) 12611.43(LB)	0.01%(UB Gap)
21	5.3 min	>2.98 hr.	7810.32	7807.98(UB) 5806.61(LB)	0.02%(UB Gap)
22	48.72 sec	48.72 sec	1630.4	1630.4	0.00%
23	57.45 sec	57.45 sec	1701.99	1701.99	0.00%
26	2.09 min	2.75 min	1915.75	1915.534	-0.01%
36	2.67 min	2.61 min	2631.65	2630.874	0.02%
46	57.73 sec	57.73 sec	3347.04	3347.04	0.00%
56	31.64 sec	31.64 sec	4066.15	4066.15	0.00%
66	1.32 min	1.75 min	4786.15	4785.52	0.06%

We also test the randomized fix-and-optimize heuristic on the Sioux Falls network which is a real network with 24 nodes, 76 links and 87 OD pairs. There are two sets of problem instances created for the Sioux Falls network, the first set of test cases are based on the scenario that 10% of the links are randomly selected as the links that need maintenance which results in a total number of 16 work zones need to be scheduled. The percentage of links to repair in the second set of test cases is 20% and the total number of work zones to be scheduled is 25. The time limits on solving the entire problem initially and on solving each FO subproblem are both 40 seconds for first set of test cases, and both are 120 seconds for the second set of test cases.

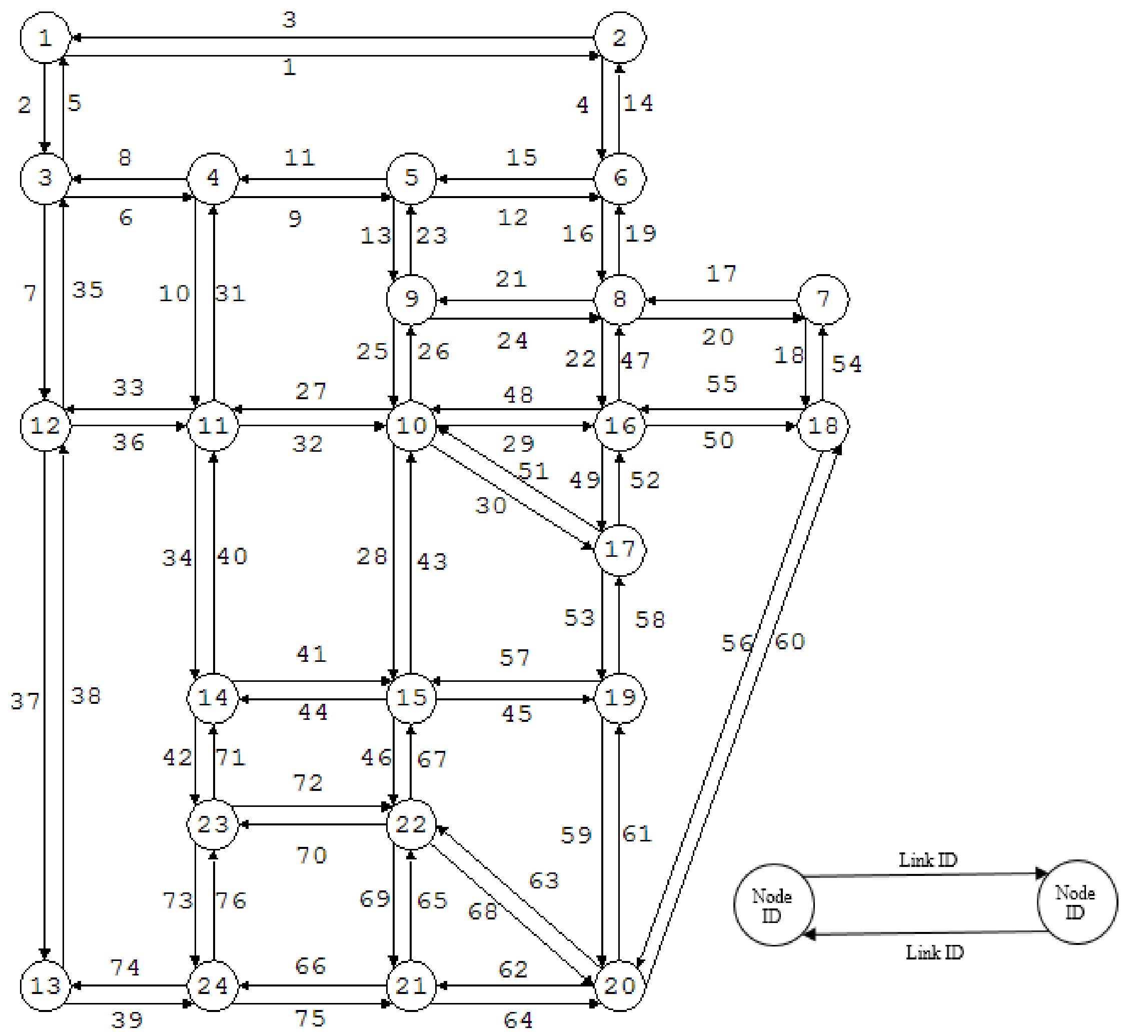


Figure 3.4.3-iii: Sioux Falls Network

Table 3.4.3-iii and Table 3.4.3-iv on the next two pages give the performance comparison between RFO and CPLEX on the first and second set of test cases respectively. Again, RFO solves each test case five times, and the objective values and solving time of each run are listed in Appendix A.

Table 3.4.3-iii: RFO VS CPLEX on Sioux Falls Network with 10% of Links to Repair

Completion Date (*T*)	Solving Time		Objective Value		Objective Value Difference
	RFO	MIP	RFO	MIP	
18	33 sec	33 sec	232233.88	232233.88	0.00%
19	3.33 min	1.92 min	237499.4	237458.8	0.02%
20	3.384 min	2.22 min	242533.2	242531.8	0.00%
21	2.44 min	1.1 min	247323.4	247342.39	-0.01%
22	2.788 min	57.67 sec	252177	252203.14	-0.01%
23	3.464 min	1.12 min	260322	260489.83	-0.06%
24	3.73 min	2.11 min	268560.4	268666.57	-0.04%
25	5.544 min	3.85 min	277241.2	277160.17	0.03%
26	5.176 min	6.21 min	285831.4	285930.8	-0.03%
27	6.168 min	3.16 min	294679.2	294426.42	0.09%
28	7.308 min	16.39 min	303283	302816.79	0.15%
29	7.512 min	12.8 min	311629.4	311690.72	-0.02%
30	7.89 min	13.27 min	320744.6	320326.55	0.13%
31	9.556 min	18.85 min	329453.2	329038.46	0.13%
32	10.066 min	11.79 min	338659.2	338241.54	0.12%
33	5.598 min	12.82 min	348635.4	347560.79	0.31%
34	9.208 min	17.7 min	357030.2	356870.16	0.04%
35	10.23 min	23.64 min	366264.4	366126.06	0.04%
36	9.654 min	16.4 min	375561.2	375420.78	0.04%
37	10.74 min	16.12 min	380249	385436.84	-1.35%
38	10.99 min	24.18 min	395848.2	395675.45	0.04%

From Table 3.4.3-iii we see that when the completion date is small the RFO takes more time to give the final solution than CPLEX does. This is because the problem instance of Sioux Falls network with 10% of links to repair is relatively easy to solve especially when the completion date is small, since the number of integer variables are not large. As the completion date gets larger, the problem instance has more integer variables and gets harder to solve. As a result, the solving times of test cases with larger completion dates are much longer for CPLEX. As a comparison, the solving times for RFO on these test cases increase slightly and the objectives obtained are close to the optimal objectives given by CPLEX.

Table 3.4.3-iv: RFO VS CPLEX on Sioux Falls Network with 20% of Links to Repair

Completion Date (*T*)	Solving Time		Objective Value		Objective Value Difference
	RFO	MIP	RFO	MIP	
28	44.086 min	2.15 hr.	446506.6	443226.27	0.74%
29	48.692 min	2.37 hr.	451906.4	451307.57	0.13%
30	41.016 min	3.22 hr.	462594.8	459098.29	0.76%
31	1.0915 hr.	2.39 hr.	468561.2	466737.54	0.39%
32	1.0075 hr.	3.68 hr.	475069.6	474657.98	0.09%
33	51.396 min	3.14 hr.	486382.2	483550.96	0.59%
34	1.258 hr.	4.37 hr.	495743.8	492508.96	0.66%
35	1.398 hr.	>1.29 hr.	502681.4	502912.96 (UB) 445782.53 (LB)	-0.05%(UB Gap)
36	1.234 hr.	>1.37 hr.	513386.2	511092.08 (UB) 463690.32 (LB)	0.45%(UB Gap)
37	1.29 hr.	>1.38 hr.	522632.2	521498.54 (UB) 459461.32 (LB)	0.22%(UB Gap)
38	36.59 min	>1.4 hr.	549474.2	529503.64 (UB) 464731.92 (LB)	3.77%(UB Gap)
39	36.994 min	10.4 hr.	548756.8	537251.06	2.14%
40	42.258 min	>1.42 hr.	562862	547592.55 (UB) 469568.44 (LB)	2.79%(UB Gap)
41	43.08 min	>1.4 hr.	568607	555430.09 (UB) 52061.60 (LB)	2.37%(UB Gap)
42	50.53 min	>1.43 hr.	585013	566841.84 (UB) 482454.42 (LB)	3.21%(UB Gap)

Data in Table 3.4.3-iv shows that when 20% of links need maintenance, solving time of CPLEX increase significantly. RFO has pretty good performance in solving most of the problem instances because it gives near-optimal solutions with much less time compared to CPLEX. For problem instances with completion dates of 38, 40, and 42, the optimality gaps are relatively large compared to those of other problem instances. This means the parameters of RFO are not appropriately set for these problem instances, and adjustments like increasing the time limits of the FO subproblems and/or changing the RFO iterations to be performed can improve the performance of RFO.

Notice that in Table 3.4.3-iii and 3.4.3-iv the objective obtained by RFO for some test cases is better than the optimal objective obtained by CPLEX. For example, in Table 3.4.3-iii for the test case when $T = 23$, the objective obtained by RFO is 260302, which is less than the optimal objective 260489.83 from CPLEX. This is because the relative MIP gap tolerance is set to 0.5% for the CPLEX and FO subproblems. CPLEX stops solving process as soon as the relative optimality gap (which is calculated as upper bound minus lower bound and then divide the difference by the upper bound) is under 0.5% and uses the best feasible solution obtained as the optimal solution, which is same for FO subproblems. But because of the randomized grouping of links that need repair, it is possible for a FO subproblem start with a branching node that leads to a better upper bound when the 0.5% relative optimality gap is reached, and this node is not selected or reached by CPLEX in the regular branch-and-bound process. So when the 0.5% relative optimality gap is reached, the upper bound obtained by CPLEX is not as good as the one obtained by RFO. If we reduce the relative MIP gap tolerance to 0.1% or smaller for CPLEX, CPLEX should be able to obtain the same final solution but certainly with much more time spent on the branch-and-bound process.

3.5 Conclusion

In this chapter, a mixed-integer linear programming model is formulated to schedule work zones in networks of service vehicles (MS-NSV). The model schedules work zones with network-wide perspective to achieve minimum total flow cost of all OD demands throughout the project period. The MS-NSV problem is very challenging and CPLEX cannot solve it efficiently. To give an example, CPLEX is not able to obtain the optimal solution for a small network with 20 links after hours of computation on a personal computer. The randomized fix-and-optimize heuristic (RFO) is developed to solve the problem efficiently, which can obtain optimal or near-optimal solutions with much less time compared to solving the MS-NSV problem solely with CPLEX. The performance of RFO and CPLEX are compared on various tests cases to illustrate the advantage that RFO has over CPLEX.

Since to schedule the work zones (lane closures) is essentially to manage the mandatory network capacity changes to achieve the minimum negative impacts on service vehicle flows, the MS-NSV problem is a network capacity management problem. The network flows model used in the MS-NSV problem is the multi-commodity flow model with system optimum as the objective, where link capacity reductions absolutely cannot reduce the total flow cost. The next chapter will briefly introduce the proposed research aimed at addressing the network capacity management problem in networks with user-optimized flows, where selective link capacity reductions may reduce the total flow cost. It also briefly discusses the proposed research that studies the maintenance planning in networks with both the flow type with system optimum as the objective, and the flow type conforming the user equilibrium principle.

Chapter 4

NETWORK-LEVEL TRAFFIC MANAGEMENT THROUGH SELECTIVE LINK CAPACITY REDUCTIONS (OCREC)

4.1 Introduction

In 2015, people in the 52 metropolitan areas in US experience peak-hour travel times that are 37% longer than the off-peak travel times on average (FHWA, 2015). Besides travel delays, traffic congestion also leads to higher fuel consumption and pollution, unsafe travel conditions and longer response time for emergency vehicles.

Although there are various non-recurring events that cause traffic congestion, such as incidents (25% of overall congestion), work zones (10%), and weather (15%), half of all congestion happens day after day at the same time and location (FHWA, 2016). And this recurring congestion is imputed to the basic imbalance between traffic supply and demand. To reverse this traffic supply-demand imbalance, abundant research has been conducted on how to increase traffic supply (i.e., network capacity expansion) smartly (e.g., Ewing and Proffitt, 2016; Fan and Gurmu, 2014; Gan et al. 2013; Mathew and Sharma, 2009; Msigwa et al. 2015). However, because of the expensive road construction cost and quick saturation of the newly built roads due to growing travel demand, network capacity expansion turns out to have very limited effect on alleviating traffic congestion. Another caveat about network capacity expansion is the counter-intuitive situation described by the Braess Paradox (Braess et al. 2005), where adding a road to a congested road network could increase the overall travel time.

What's more, as pointed out by FHWA (2017), traffic bottlenecks (e.g., freeway entrance/exit, lane drop, weaving areas, freeway-to-freeway interchanges) are increasingly the issue that cause congestion. To alleviate congestion at traffic bottlenecks, traffic flow control mechanisms like ramp metering (e.g., Chai et al., 2015; Jiang and Chung, 2015; Landman et al., 2016; Li and Chow, 2015; Meshkat et al., 2015; Osman et al., 2015; Perrine et al., 2015; and Reilly et al., 2015) and variable speed limit (Carlson et al., 2014; Carlson et al., 2010; Li et al., 2014; and Lu et al., 2011) have also been studied and implemented in practice. Researchers have investigated ways to limit the number of vehicles traveling through the bottlenecks so that traffic flow can be smoother and moving, instead of stop-and-go or completely stagnant. However, these mechanisms only focus on local traffic conditions and often are not able to improve the network-wide travel time much for all the travelers (Levinson and Zhang, 2006).

Another approach to reduce traffic congestion is travel demand management. With a network-wide perspective, this approach deals with the imbalance between traffic supply and demand focusing on the demand side, and aims at reducing the travel demand for some time periods (typically the rush hours) and travel modes (mostly private cars) by effectively influencing people's travel activities (e.g., departure time, route selection, travel mode selection) though monetary pricing or incentives.

Although user equilibrium (UE) flow (Wardrop, 1952) routing guarantees the fairness among all travelers, it is not an efficient flow pattern compared to the system optimum (SO) flow (Wardrop, 1952), where some travelers may need to take routes with longer travel times so that the total travel time of all the travelers can be minimized. Hence, mechanisms that can drive UE flows towards more efficient flow patterns, such as imposing tolls on some or all the links (e.g., Bao et al., 2015; Chen et al., 2015; Liu et al., 2014; Guo, 2013; Marcotte and Zhu, 2009; Wang and Ehrgott, 2013), have continuously drawn researchers' attention. However, tolls may sometimes be considered not very practical due to the high cost of toll collection and the socio-economic differences on value of time.

Besides the vast literature on the toll imposition strategies, there exists research that use incentives to guide the routing of traffic flows. For example, Hu et al. (2017) developed an integrated system called Metropia that can calculate the optimal and near-optimal routes and estimate the corresponding travel times based on real-time traffic data and departure time options. The system influences people's travel habits by assigning tradable credits to departure times and route options that are better for the entire traffic system, and work with well-known stores or chains to let travelers use the tradable credits to buy products or gift cards. This system has been implemented in large cities like Los Angeles and such implementations demonstrate its effectiveness in changing people's travel habits and alleviating congestion. One drawback of this approach is that travel demand is too rigid to allow such systems to reduce traffic congestion substantially, because most people must travel during rush hours to arrive at work on time.

Based on the advantages and limitations of the aforementioned congestion alleviation approaches, this chapter studies a solution to traffic congestion with a network-wide perspective but without physically expanding the network capacity. Enlightened by the situations described by Braess Paradox - that building a road to a congested road network could increase the overall travel time and blocking a road in a congested network may decrease the overall travel time – this research explores ways to selectively reduce the capacity of some roads to improve the overall efficiency of the traffic flow at UE. It is a strategy that attempts to drive user equilibrium flows toward more efficient flow patterns without the introduction of monetary pricing or incentive schemes.

4.2 Related Work

Research that has studied control mechanisms to improve the efficiency of traffic flows without monetary penalties or incentives are handful. Jahn et al. (2005) developed a route guidance system that solved a constrained system optimum problem with user constraints. Based on route travel times at user equilibrium, their model generated routes with travel times that were within a certain range of the UE travel times for each origin-destination (OD) pair, and solved the system optimum using these routes generated. The "fairness" of the routes recommended to the travelers is related to the width of the range. If the range is very small, the traffic flow will be closer to the UE flow; and if the range is large, the traffic flow will resemble the SO flow and some travelers might be recommended to take lengthy detours. Schulz and Stier-Moses (2006) showed mathematically that the route guidance system developed by Jahn et al. (2005) results in a traffic assignment that is provably efficient and close to fair, the efficiency and fairness of the resultant traffic pattern still depends on the range parameter. An important presumption in such a system is that every traveler uses the same route guidance system and follows the recommended route, which is not practical.

The numerous literature on Braess paradox in transportation networks can be categorized into two major groups. The first group studies methods to detect whether Braess paradox could occur (Chen et al., 2016; Di et al., 2014; Hwang and Cho, 2016; Pas and Principio, 1997; Steinberg and Zangwill, 1983; Valiant and Roughgarden, 2010; Zverovich and Avineri, 2015). The methods proposed in these studies only works well for single OD demand and linear link travel time functions (i.e., link travel time is a linear function of the flow on the link). Such approaches have difficulties in detecting Braess paradox in networks of moderate size or with multiple OD pairs, or with general continuous non-decreasing link travel time functions.

The second group have studied the bounds of the Braess ratio (Lin et al., 2005; Lin et al., 2011; Roughgarden, 2006), which is obtained by dividing the total travel time at UE for the original network by the solution after the removal of some links. Only two papers are found to apply Braess paradox to improve the efficiency of UE flows by entirely closing some links in the network. Askoura et al. (2011), developed a path-based approach to find the sub-network which reduced the total travel time at UE. They analyzed the total OD flow cost with all paths enumerated, and removed some of the links based on the travel demand volume to obtain the desired sub-network. However, their approach did not work well in test cases with multiple OD pairs in networks of moderate size. Bagloee et al. (2014) proposed a method to obtain a pool of candidate links by comparing the total travel time at UE before and after the closure of the link, and then used a genetic algorithm to find a good combination of the links to close to reduce the total UE travel time. The disadvantage of their approach is that for large networks with many links, both obtaining the pool of candidate links and searching for the combination of links to close is computationally unwieldly.

In light of the aforementioned literature, our research studies network-level traffic management through selective link capacity reductions, which essentially is the optimal capacity reduction with equilibrium constraints (OCREC). First, it develops different optimization models to investigate the existence of Braess paradox when links are not entirely removed but the capacity may be reduced. Second, it introduces a new way to identify links whose capacity reduction may reduce the total travel time at UE by comparing the link flows at UE and SO. Since the link travel time function used here is the Bureau of Public Roads (BPR) function, both the UE and SO problems are nonlinear problems. We used the "Traffic Assignment by Paired Alternative Segments" (Bar-Gera, 2010) approach to solve the UE problem, and developed a Frank-Wolfe type algorithm to solve the SO problem efficiently. Third, we develop a heuristic to find a good combination of links and the desired capacity reduction that results in more efficient UE flows.

4.3 OCREC Models

In the problem of OCREC, it is assumed that there are fixed origin-destination (OD) flow demands routed through the network as per Wardrop's first principle (Wardrop, 1952), where every traveler has perfect knowledge of path travel times, and will rationally choose the path that minimizes his/her own travel time. Because of this selfish routing, the static and deterministic equilibrium flow will be reached where no single traveler can reduce his/her travel time by changing the route unilaterally. All (or a subset of) the links in the network may reduce their capacities by some amount. If a link has capacity reduction, its link travel time will change for the same link flow. As a result, the flow patterns at the equilibrium before and after link capacity reductions most likely will be different, and so will the total travel times. The goal of OCREC is to find the optimal set of links and the optimal amount of capacity reductions on these links so that the new UE has the lowest total travel time.

OCREC is a two-level problem. The upper level problem is to find the links on which reduce capacity and the amount to reduce. With the decreased link capacities, the lower level problem is a traffic equilibrium problem which obtains the user equilibrium (UE) flows and the associated total travel time. Based on how UE condition is enforced, the traffic equilibrium problem can be formulated as a path-based model or a link-based model. In the path-based model, specific variables are defined for the path cost and path flows, and the UE condition is enforced by complementarity constraints on these path variables. In the link-based model where path cost and path flows are not calculated, the UE condition is ensured by the Beckmann's objective function:

$$minimize \ \sum_{i\in E} \int^{x_i} t_i(\omega)d\omega \ ,$$

where x_i is the flow on link i, E is the link set and $t_i(x_i)$ is the link travel time function of link i evaluated at x_i. Sheffi (1984) proved that the flow pattern obtained by the Beckmann's objective function satisfies the UE condition.

Based on whether or not link capacity constraints are explicitly defined, traffic equilibrium models can be grouped into two categories: the capacitated traffic equilibrium models and the uncapacitated traffic equilibrium models. In capacitated traffic equilibrium models, hard link capacity constraints are modeled to ensure that the total amount of link flow do not exceed the link capacity. The resultant traffic flow in this case is a constrained equilibrium since OD flows will choose the path with the second lowest cost if at least one link on the path with the lowest cost has reached its capacity, or choose the path with the third lowest cost if at least one link on the path with the lowest cost and at least one link on the path with the second lowest cost have reached their capacities, and so on.

In uncapacitated traffic equilibrium models, hard link capacity constraints are not explicitly defined; "link capacity" usually is a parameter in link travel time function so that link travel time increases rapidly once the link flow exceeds the nominal link capacity. For example, Bureau of Public Roads (BPR) link travel time function is:

$$t_i(x_i) = t_i^0 * \left(1 + \alpha \left(\frac{x_i}{C_i}\right)^{\beta}\right),$$

where t_i^0 is the free-flow travel time on link i, C_i is the nominal link capacity, and α and β (greater than 1) are parameters. Link travel time will increase nonlinearly when link flow exceeds the nominal link capacity. Thus, with the objective of minimizing the total travel time at UE for the path-based model, or the Beckmann's function for the link-based model, these link travel time functions work as "soft capacity constraints" to prevent link flows from exceeding nominal link capacities too much. Because no hard link capacity constraints are imposed, traffic flow pattern obtained in this case satisfies the UE condition precisely.

Since many network test cases are uncapacitated traffic equilibrium models with BPR link travel time functions, and they are widely used by transportation researchers and practitioners, the research presented in this chapter also assumes the BPR function and adopts the uncapacitated model for the lower level traffic equilibrium problem. The next two sub-sections respectively discuss the path-based model and link-based model for the OCREC problem.

4.3.1 Path-based Model. The path-based model is a single level optimization model with the objective to minimize the total flow cost and with complementarity constraints to ensure user equilibrium condition. Denote variables $\boldsymbol{c_p^k}$ and $\boldsymbol{f_p^k}$ as the travel time and the amount of flows on path $\boldsymbol{p}$ of OD pair $\boldsymbol{k}$ respectively. The objective function is formulated as $\mathbf{min} \sum_{\boldsymbol{k \in OD}} \sum_{\boldsymbol{p \in P_k}} \boldsymbol{c_p^k f_p^k}$, where $\boldsymbol{E}$ is the set of links, $\boldsymbol{OD}$ is the set of OD demand pairs, and $\boldsymbol{P_k}$ is the path set of OD pair $\boldsymbol{k}$. To ensure all the OD demands are satisfied, parameter $\boldsymbol{D_k}$ is introduced as the demand of OD pair $\boldsymbol{k}$, and constraint $\sum_{\boldsymbol{p \in P_k}} \boldsymbol{f_p^k} = \boldsymbol{D_k}$ is formulated for each OD pair so that the total amount of flows on all the paths connecting the OD pair equals to the demand for that OD pair. Let variable $\boldsymbol{x_i}$ be the total amount of flows on link $\boldsymbol{i}$ and $\boldsymbol{\delta_{i,p}^k}$ be the binary parameter indicating whether link $\boldsymbol{i}$ is on path $\boldsymbol{p}$ of OD pair $\boldsymbol{k}$, then constraint $\boldsymbol{x_i} = \sum_{\boldsymbol{k \in OD}} \sum_{\boldsymbol{p \in P_k}} \boldsymbol{f_p^k \delta_{i,p}^k}$ is added for each link to make sure all the OD flows that using the link are accounted in the link flow.

Introducing $t_i(x_i)$ as the link travel time function, constraint $c_p^k = \sum_{i \in E} \delta_{i,p}^k t_i(x_i)$ is formulated for each path to calculate the path travel time. Define variable r_i as the capacity reduction on link i, then $t_i(x_i)$ is calculated as $t_i^0 * \left(1 + \alpha \left(\frac{x_i}{C_i - r_i}\right)^\beta\right)$ for links that can have capacity reductions, where t_i^0 is the free-flow travel time on link i, C_i is the link capacity, and α and β (greater than 1) are parameters. For links where capacity reductions

are not allowed, the link travel time function remains as $t_i(x_i) = t_i^0 * \left(1 + \alpha \left(\frac{x_i}{C_i}\right)^\beta\right)$. In order to restrict the capacity reduction on a link, parameter r_i^{max}, which is smaller than C_i, and constraints $r_i \le r_i^{max}$ are added to the model. Let variable c_{min}^k be the minimum travel time between OD pair k, then constraint $c_{min}^k \le c_p^k$ assures that c_{min}^k is not larger than the travel time of any path of OD pair k. This user equilibrium condition is then ensured by the complementarity constraints $0 \le \left(c_p^k - c_{min}^k\right) \perp f_p^k \ge 0$ formulated for each path of all the OD pairs, which basically means if $c_p^k - c_{min}^k > 0$, $f_p^k = 0$; and if $f_p^k > 0$, $c_p^k - c_{min}^k = 0$. The sets, parameters, variables and functions used in the OCREC path-based model are given in Table 4.3.1-1:

Table 4.3.1-i: Notations for Path-based OCREC

Term	Definition
Sets	
E	The set of existing links in the network
R	The set of existing links that allows capacity reduction $R \subseteq E$
OD	The set of Origin-Destination pairs of flows
P_k	The set of paths for OD pair k

Term	Definition
Parameters	
C_i	Traffic flow capacity of link i
t_i^0	Travel time on link i when there is no traffic flow on the link
r_i^{max}	Maximum capacity reduction can be achieved on link i
$\delta_{i,p}^k$	Binary variable indicating whether link i is on path p of OD pair k. If it is, then $\delta_{i,p}^k = 1$; otherwise $\delta_{i,p}^k = 0$
D_k	Flow demand of OD pair k
Variables	
x_i	Traffic flow on link i
r_i	Capacity reduction caused on link i
f_p^k	Traffic flow of OD pair k on path p of the OD pair, $p \in P_k$
c_p^k	Travel time on path p of the OD pair k, $p \in P_k$
c_{min}^k	Minimum travel time of all the paths of OD pair k
Functions	
$t_i(x_i)$	Travel time on link i when the flow on the link is x_i. For $\forall i \in E \backslash R, t_i(x_i) = t_i^0 \left[1 + \alpha \left(\frac{x_i}{C_i}\right)^\beta\right]$; for $\forall i \in R, t_i(x_i, r_i) = t_i^0 \left[1 + \alpha \left(\frac{2x_i}{C_i - r_i}\right)^\beta\right]$.

The complete path-based model of optimal capacity reduction with equilibrium constraints (OCREC) can now be written as:

Path-based OCREC:

$$minimize\ z(\boldsymbol{x}, \boldsymbol{r}) = \sum_{k \in OD} \sum_{p \in P_k} c_p^k f_p^k \quad (1)$$

$$\sum_{p \in P_k} f_p^k = D_k \qquad \forall k \in OD \quad (2)$$

$$x_i = \sum_{k \in OD} \sum_{p \in P_k} f_p^k \delta_{i,p}^k \qquad \forall i \in E \quad (3)$$

$$c_p^k = \sum_{i \in E} \delta_{i,p}^k t_i(x_i) \qquad \forall k \in OD, \forall p \in P_k \quad (4)$$

$$c_{min}^k \leq c_p^k \qquad \forall k \in OD, \forall p \in P_k \quad (5)$$

$0 \leq \left(c_p^k - c_{min}^k\right) \perp f_p^k \geq 0$	$\forall k \in OD, \forall p \in P_k$	(6)
$0 \leq r_i \leq r_i^{max}$	$\forall i \in R$	(7)
$f_p^k \geq 0$	$\forall k \in OD, \forall p \in P_k$	(8)
$c_p^k \geq 0$	$\forall k \in OD, \forall p \in P_k$	(9)
$c_{min}^k \geq 0$	$\forall k \in OD$	(10)
$x_i \geq 0$	$\forall i \in E$	(11)

Theoretically, with all paths of each OD pair enumerated, the path-based OCREC can be solved in one shot by nonlinear commercial solvers that can handle complementarity constraints (e.g., Knitro®). For example, for a test case based on the four-node network displayed in Figure 4.3-1. There are 40 units of travel demand from node 1 to node 4 and 20 units from node 3 to node 4. All the five links can have capacity reductions and the capacity reduction limit is 0.0001 unit less than the original link capacity. Detailed network data is given in Appendix B. Given all the paths connecting the two OD pairs found, Knitro solves this test case in less than one second. The optimal solution is to reduce the capacity of link 5 by 59.999 units, and the total travel time at UE after the capacity reduction is 3042.555. As a comparison, the total travel time at UE before the capacity reduction is 3066.637.

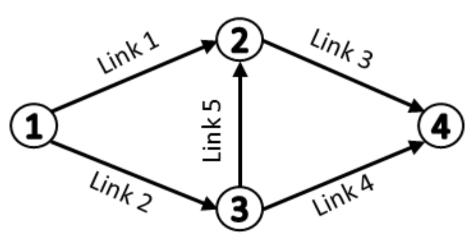


Figure 4.3.1-i: Four-Node Network

It is commonsense that enumerating all the paths for each OD pair is impractical even for networks of moderate size. Thus, a more reasonable approach to solve the path-based model is to generate a set of paths based on current link travel times, solve the restricted OCREC problem with the paths found, update the link travel times based on the new flow pattern, and find more paths for the restricted problem in the next iteration. The computation procedure will end when the total travel time between two iterations are close enough.

The path-based OCREC belongs to a category of optimization models referred as "mathematical programing with equilibrium constraints" (MPEC) (Luo et al., 1996), which are extremely hard to solve. This renders the path-based model not a viable approach because the restricted OCREC problem cannot be solved for larger test cases. For example, for the test case created based on the square network shown in Figure 4.3.1-ii, which has 56 links and 14 OD pairs, Knitro computes for 3 hours but still returns an infeasible solution for the restricted problem with 12 paths found for each OD pair. The detailed data of the network and OD demand of the test case given in Appendix B.

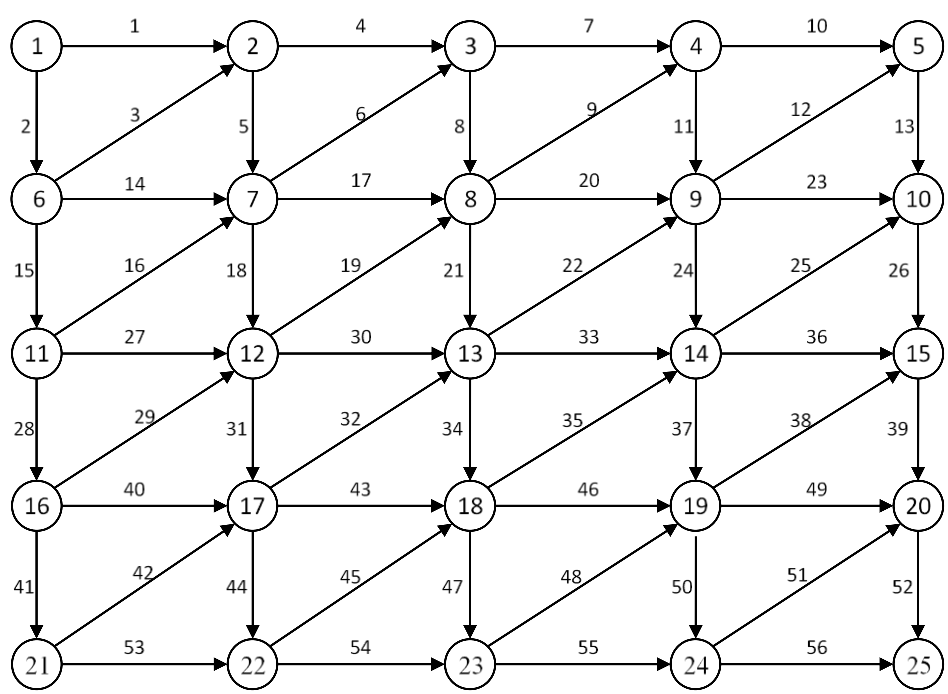


Figure 4.3.1-ii: Square Network

Since the difficulty of solving the restricted OCREC problem is mostly due to the complementarity constraints, we reformulated the path-based OCREC without these complementarity constraints. Introducing binary variable w_p^k for $\forall k \in OD, \forall p \in P_k$, the complementarity constraint (constraint 6) can be reformulated as:

$$c_p^k - c_{min}^k \leq M w_p^k \qquad \forall k \in OD, \forall p \in P_k \qquad \text{(a)}$$

$$f_p^k \leq M\left(1 - w_p^k\right) \qquad \forall k \in OD, \forall p \in P_k \qquad \text{(b)}$$

$$w_p^k \in \{0, 1\} \qquad \forall k \in OD, \forall p \in P_k \qquad \text{(c)}$$

where M is a big number. We can examine the effectiveness of these three constraints on ensuring the UE flow pattern in two scenarios. The first scenario is if the path travel time equals to the minimum travel time for the OD pair, we need to verify whether the path flow can be positive. In this scenario, $c_p^k - c_{min}^k = 0$ and the binary variable w_p^k can take value 0 in constraint (a), enabling path flow variable f_p^k to take positive values in constraint (b). The second scenario is if the path travel time is longer than the minimum travel time for the OD pair, we need to verify whether the path flow is 0. In this scenario, $c_p^k - c_{min}^k > 0$, forcing w_p^k to take value 1 in constraint (a). And $w_p^k = 1$ forces f_p^k to be 0 in constraint (b). Hence constraints (a), (b) and (c) ensure that flow pattern obtained is UE flow.

This reformulation is easier to solve since Knitro can find a feasible solution for the square network with 12 paths found for each OD pair, whereas Knitro returns an infeasible solution for the same test case for the original model with complementarity constraints. However, it takes intolerably long time for Knitro to solve the restricted problem with more paths generated for the square network, let alone networks in larger sizes. Therefore, the path-based OCREC is only used for the validation of solution methods developed for the link-based approach for small networks, which are presented in the following sections.

4.3.2 Link-based Model. The link-based model is a bi-level optimization model. The upper level obtains the link capacity reductions, and the lower level computes the UE flows associated with the link capacity reductions obtained from the upper level. Denote $U(\boldsymbol{x}, \boldsymbol{r})$ as the total travel time at UE under the capacity reduction scenario $\boldsymbol{r}$, where $\boldsymbol{r}$ is the vector of link capacity reductions and $\boldsymbol{x}$ is the vector of link flows. The objective of the upper level model is $minimize\ z(\boldsymbol{r}) = U(\boldsymbol{x}, \boldsymbol{r})$, which is to find the link capacity reductions that give the lowest total travel time at UE. Constraint $0 \le r_i \le r_i^{max}$ is added for each link to restrict the amount of capacity that can be reduced on the link.

The lower level is the classic traffic equilibrium model. Define variable x_i^k as the flow from OD pair k on link i, constraint $x_i = \sum_{k \in OD} x_i^k$ is formulated to ensure flow consistency for each link so that the flows from all OD pairs using the link are accounted for in the total link flow. For each OD pair, flow conservation constraints, consisting of three groups, are needed. The first group of constraints makes sure the total incoming flow units minus the total outgoing flow units equal to the OD demand for the origin node of the OD pair. Let D_k be the demand of OD pair k, the first group is formulated as $D_k = \sum_{\{i: E_i^- = OD_k^-, i \in E\}} y_{ik} - \sum_{\{j: E_j^+ = OD_k^-, j \in E\}} y_{jk}$ for $\forall k \in OD$, where OD_K^- is the origin node of OD pair k, E_i^- is the head node of link i and E_j^+ is the tail node of link j. The second group ensures the total outgoing flow units minus the total incoming flow units equal to the demand of OD pair k for its destination node and is formulated as $D_k = \sum_{\{i: E_i^+ = OD_k^+, i \in E\}} y_{ik} - \sum_{\{j: E_j^- = OD_k^+, j \in E\}} y_{jk}$ for $\forall k \in OD$, where OD_K^+ is the destination node of OD pair k, E_i^+ is the tail node of link i and E_j^- is the head node of link j. For the rest of the nodes, other than origin and destination nodes of OD pair k, the total incoming flows on the node from the origin of OD pair k should equal to the total outgoing flows from the node to the destination of the OD pair k. This is the third group of the flow conservation constraints and it is formulated as $\sum_{\{i:E_i^- =l, i\in E\}} y_{ik} = \sum_{\{j:E_j^+ =l, j\in E\}} y_{jk}$ for $\forall l \in N, \forall k \in \{k: OD_k^- \neq l\} \cap \{k: OD_k^+ \neq l\}$, where N is the set of nodes in the network.

The sets, parameters, variables and functions used in the link-based OCREC model are presented in Table 4.3.2-i:

Table 4.3.2-i: Notations for Link-based OCREC

Term	Definition
Sets	
N	Node set of the network
E	The set of existing links in the network
R	The set of existing links that allow link capacity reductions $R \subseteq E$
OD	The set of Origin-Destination pairs of flows
Parameters	
C_i	Traffic flow capacity of link i
t_i^0	Travel time on link i when there is no traffic flow on the link
r_i^{max}	Maximum capacity reduction can be achieved on link i
E_i^-	Head node of link i
E_i^+	Tail node of link i
OD_k^-	Origin node of OD pair k
OD_k^+	Destination node of OD pair k
D_k	Flow demand of OD pair k
Variables	
x_i^k	Traffic flow on link i from OD pair k
x_i	Traffic flow on link i from all OD pairs
r_i	Capacity reduction on link i

Functions	
$t_i(x_i)$	Travel time on link i when the flow on the link is x_i. If BPR function is used, for $\forall i \in E\backslash R$, $t_i(x_i) = t_i^0\left[1+\alpha\left(\frac{x_i}{C_i}\right)^\beta\right]$; for $\forall i \in R$, $t_i(x_i) = t_i^0\left[1+\alpha\left(\frac{x_i}{C_i - r_i}\right)^\beta\right]$.

The complete link-based model of optimal capacity reduction with equilibrium constraints (OCREC) is shown below:

Upper Level of Link-based OCREC:

$$minimize\ z(\boldsymbol{r}) = \sum_{i \in E} t_i(x_i) * x_i \quad (1)$$

$s.t.$:

$$0 \le r_i \le r_i^{max} \qquad \forall i \in R \quad (2)$$

Lower Level of Link-based OCREC:

$$minimize\ z(\boldsymbol{x}) = \sum_{i \in E} \int_0^{x_i} t_i\,(\omega) d\omega \quad (3)$$

$s.t.$:

$$x_i = \sum_{k \in OD} x_i^k \qquad \forall i \in E \quad (4)$$

$$D_k = \sum_{\{i: E_i^- = OD_k^-, i \in E\}} x_i^k - \sum_{\{j: E_j^+ = OD_k^-, j \in E\}} x_j^k \quad \forall k \in OD \quad (5)$$

$$D_k = \sum_{\{i: E_i^+ = OD_k^+, i \in E\}} x_i^k - \sum_{\{j: E_j^- = OD_k^+, j \in E\}} x_j^k \quad \forall k \in OD \quad (6)$$

$$\sum_{\{i: E_i^- = l, i \in E\}} x_i^k = \sum_{\{j: E_j^+ = l, j \in E\}} x_j^k \qquad \forall l \in N, \forall k \in \{k: OD_k^- \neq l\} \cap \{k: OD_k^+ \neq l\} \quad (7)$$

$$x_i \ge 0 \qquad \forall i \in E \quad (8)$$

This link-based OCREC is not convex. To give an example, we create a problem instance based on the four-node network in Figure 4.3.1-i, but with a different link capacities and free-flow travel times, and with 6 units of travel demand going from node 1 to node 4. The detailed network information can be found in Appendix B. When there is no capacity reduction, the total travel time at UE is $z(\boldsymbol{r}) = 178.528$. After 2 units of capacity is reduced on link 5, the total travel time at UE is $z(\boldsymbol{r}') = 249.462$. Let $\lambda = 0.5$, the convex combination of these two capacity reduction scenarios is to reduce 1 unit of capacity on link 5, and the total travel time at the corresponding UE is $z(\boldsymbol{r}'') =$

$z(\lambda \boldsymbol{r} + (1-\lambda)\boldsymbol{r}') = 244.8$. Since $\lambda z(\boldsymbol{r}) + (1-\lambda)z(\boldsymbol{r}') = 213.995 < 244.8 = z(\lambda \boldsymbol{r} + (1-\lambda)\boldsymbol{r}')$, we produce a case where the link-based OCREC is not convex because convexity requires $\lambda z(\boldsymbol{r}) + (1-\lambda)z(\boldsymbol{r}') \geq z(\lambda \boldsymbol{r} + (1-\lambda)\boldsymbol{r}')$ for $\lambda \in (0,1)$.

We can also show the link-based OCREC is not always cave using the original problem instance of the four-node network in Figure 4.3.1-i. The total travel time at UE is $z(\boldsymbol{r}) = 3066.635$ when there is no capacity reduction, and the total travel time at UE is $z(\boldsymbol{r}') = 3049.805$ when 30 units of capacity is reduced on link 5. Again, let $\lambda = 0.5$, the convex combination of these two capacity reduction scenarios is to reduce 15 units of capacity on link 5; the associated total travel time at UE is $z(\boldsymbol{r}'') = z(\lambda \boldsymbol{r} + (1-\lambda)\boldsymbol{r}') = 3056.738$. This demonstrates that the link-based OCREC is not concave because $\lambda z(\boldsymbol{r}) + (1-\lambda)z(\boldsymbol{r}') = 3058.221 > 3056.738 = z(\lambda \boldsymbol{r} + (1-\lambda)\boldsymbol{r}')$, and concave problems must have $z(\boldsymbol{r}) + (1-\lambda)z(\boldsymbol{r}') < z(\lambda \boldsymbol{r} + (1-\lambda)\boldsymbol{r}')$ for $\lambda \in (0,1)$.

Therefore, the link-based OCREC is a challenging bi-level nonlinear optimization problem that is neither convex nor concave. Hence, we develop a heuristic for the link-based model which systematically identifies candidate links for capacity reduction and the desired amount of reduction.

4.4 Approximate Solution Approach for the Link-based Model

4.4.1 Structure of the Heuristic. Because the system optimum (SO) is the most cost efficient flow pattern, the primary purpose of the heuristic developed is to move the user equilibrium (UE) flows toward SO flow by reducing the link capacities. To find candidate links for capacity reduction, the UE problem and the SO problem are first solved for the original network without capacity reductions. Based on the UE and SO flows obtained, links are sorted in descending order with respect to the link flow difference between UE and SO. We consider links with positive flow difference between UE and SO are overly used by the UE flows. Hence, to drive the UE flow towards the SO flow, the capacity of these links may be reduced and the link travel times will increase, and so will the travel time of the paths using these links. The UE flows then will divert to other paths with less travel time to reach a new UE which may be closer to the SO flow.

Let x_i^{UE} denote the flow on link i at UE and x_i^{SO} denote the flow on link i at SO. To find the link whose capacity reduction can improve the efficiency of UE flows, and the desired amount of capacity to decrease on the link, the heuristic starts with the link with the largest $x_i^{UE} - x_i^{SO}$ and reduces the link capacity by $0.618 * \left(x_i^{UE} - x_i^{SO}\right)$. The numeric multiplier 0.618 can be replaced by any number between 0 and 1. 0.618 is used in our heuristic because it results in the best heuristic performance for the problem instances tested. We note coincidentally that 0.618 is the ratio used in golden section line search method. Let $this\_CapRed$ represent the additional amount of capacity reduction added to the capacity reduction already accepted, initially $this\_CapReduc = 0.618 * \left(x_i^{UE} - x_i^{SO}\right)$. The UE problem will be solved to check whether the capacity reduction results in UE with less total travel time. If it does, the capacity reduction will be accepted and additional capacity reduction with the amount of $\frac{1}{0.618} * this\_CapReduc$ will be attempted on the same link. The reason to set the multiplier $\frac{1}{0.618}$ is to have more aggressive capacity reduction trials given the last capacity reduction trial is effective, and accelerate the search for the desired link capacity reduction. Otherwise, the capacity reduction will be reverted and the new capacity reduction to be considered is $0.382 * this\_CapReduc$, and the UE problem will be solved to check whether more efficient UE flow is obtained. This process continues until either r_i^{max} is reached or the total travel time at UE of two consecutive capacity reduction attempts are very close.

If the effective link capacity reduction that results in UE with less total travel time is found, the SO problem will be solved based on the accepted link capacity reductions, and links are sorted again in descending order based on the value of $x_i^{UE} - x_i^{SO}$ to start a new round of link capacity reduction tests. If no effective capacity reduction is found for the current link, the capacity reduction test will continue to the next link with the largest value of $x_i^{UE} - x_i^{SO}$. The heuristic stops when the total travel time at UE is very close to the total travel time at SO, or no capacity reduction on any of the links with $x_i^{UE} - x_i^{SO} > 0$ can improve the UE flow efficiency. Let $BetterUE = 1$ indicate more efficient UE obtained through capacity reductions after the most recent link sort. The heuristic is shown on the next page, where $prev\_UETtime$ is the best total travel time obtained before the most recent solution to the SO problem, $last\_UETtime$ is the total travel time from the most recent UE, and $SOTtime$ is the total travel time from the most recent SO solution.

Inside the heuristic, the UE problem is solved by the traffic assignment with paired alternative segments (TAPAS) algorithm developed by Bar-Gera (2010). To solve the SO problem, a Frank-Wolfe type algorithm was developed which is discussed in detail in the next subsection. The heuristic for link-based OCREC is summarized on the next page and is illustrated by a flow-chart on the page after next:

Step 1: Solve the UE and SO problems with no link capacity reductions, let $best\_UETtime$, $prev\_UETtime$ and $last\_UETtime$ equal to $UETtime$.

Step 2: Calculate $x_i^{UE} - x_i^{SO}$.

Step 3: Sort the links in descending order with respect to $x_i^{UE} - x_i^{SO}$. Let $BetterUE = 0$

Step 4: Find the link with the largest positive $x_i^{UE} - x_i^{SO}$ that has not tried capacity reduction after the sort, let $this\_CapReduc = 0.618(x_i^{UE} - x_i^{SO})$. If there is no more links with positive $x_i^{UE} - x_i^{SO}$, go to Step 7.

Step 5: If $r_i < r_i^{max}$ and $r_i + this\_CapReduc < r_i^{max}$, $r_i = r_i + this\_CapReduc$;

If $r_i < r_i^{max}$ and $r_i + this\_CapReduc \geq r_i^{max}$, let $last\_CapReduc = r_i^{max} - r_i$, $r_i = r_i^{max}$, $this\_CapReduc = 0$;

If $r_i = r_i^{max}$, go back to Step 4.

Step 6: Let $last\_UETtime = UETtime$, resolve the UE problem based on the new capacity reductions.

If $UETtime$ is very close to $SOTtime$, let $best\_UETtime = UETtime$, exit the solving procedure. local optimality is obtained, exit the solving procedure;

If $best\_UETtime < UETtime$, and $UETtime$ is very close to $best\_UETtime$ or $last\_UETtime$, revert the capacity reduction. If $BetterUE = 0$, continue to the next link with the largest $x_i^{UE} - x_i^{SO}$ and go back to Step 5. If $BetterUE = 1$, solve SO based on accepted capacity reductions and go to Step 2;

If $best\_UETtime < UETtime$ and $r_i = r_i^{max}$, revert the capacity reduction, let $this\_CapReduc = 0.382 * last\_CapReduc$ and $last\_CapReduc = 0$, and go back to Step 5;

If $best_UETtime < UETtime$and $r_i < r_i^{max}$, revert the capacity reduction, let $this\_CapReduc = 0.382 * this\_CapReduc$, and go back to Step 5;

If $best\_UETtime > UETtime$ and $r_i = r_i^{max}$, let $BetterUE = 1$, accept the capacity reduction, let $best\_UETtime = UETtime$, resolve SO based on the accepted link capacity reductions and go back to Step 2;

If $best\_UETtime > UETtime$ and $r_i < r_i^{max}$, let $BetterUE = 1$, accept the capacity reduction, let $this\_CapReduc = \frac{1}{0.618} * this\_CapReduc$, let $best\_UETtime = UETtime$, and go back to Step 5;

Step 7: Compare $best\_UETtime$ and $prev\_UETtime$,

If $best\_UETtime < prev\_UETtime$, let $prev\_UETtime = best\_UETtime$, solve SO and go back to Step 2;

If $best\_UETtime = prev\_UETtime$, local optimality is obtained, exit the solving procedure.

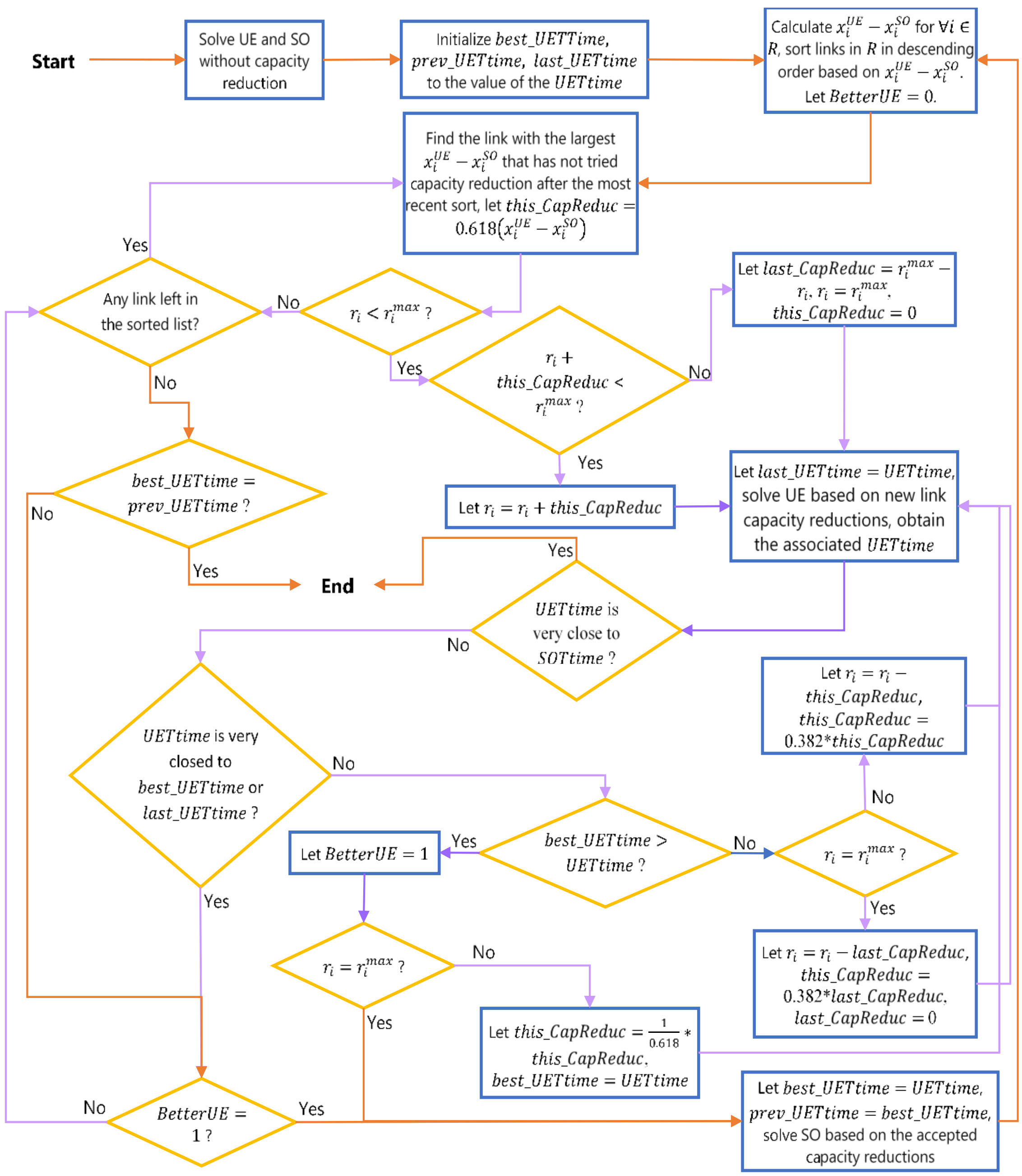


Figure 4.4.1-i: Flow Chart of Link-based OCREC

4.4.2 The Frank-Wolfe (FW) Algorithm for the System Optimum Problem.

The system optimum (SO) problem is a type of the nonlinear multi-commodity flows problem. Research on the solution methods for this problem have mostly focused on how to deal with integer flows, how to avoid oscillation during the solution procedure, and how to solve this problem as a general nonlinear programming problem. For a brief review of the literature on this topic, please refer to section 2.1.4 in Chapter 2. To our knowledge, very little research has been done on the solution method for the SO problem that takes advantage of its network flows features to improve the efficiency of the solution procedure. This subsection introduces the FW algorithm for the SO problem, which resembles the FW algorithm approach to the UE problem.

The Frank-Wolfe (FW) algorithm is a well-known exact approach for nonlinear convex problems. Based on an initial feasible solution, it calculates the gradient vector and minimizes the cross product of the gradient vector and the variable vector within the feasible region of the original problem. This minimization problem is called the direction-finding problem and formulated as a linear optimization program. Its solution produces the descending direction at the initial solution to the original problem. After the descending direction is found, a line search problem is solved to obtain the optimal step size to proceed from the current feasible solution. After a new feasible solution is obtained, another iteration of FW starts. The objective value evaluated at each feasible solution is the upper bound of the original problem. A lower bound is computed as the upper bound subtracting the cross product of the new feasible solution and the gradient at the current feasible solution. The FW algorithm iterates until the lower bound and upper bound are within a predefined tolerance range (e.g., 10^{-6}).

Because of its straightforward structure, the FW has often been applied to solve the UE problem. With an initial set of feasible flows, the basic procedure of the Frank-Wolfe algorithm for the UE problem (Sheffi, 1984) is first to solve the direction-finding problem by performing all-or-nothing assignment based on the link travel times evaluated at the initial feasible flows, with are also called primary flows. All-or-nothing assignment is to send all the travel demand of an OD pair along its shortest path. During this assignment, the link travel times are considered fixed regardless of the flows on the link. The solution obtained in the direction-finding problem is the descending direction for the primary flows. Because the objective function of the UE problem is the Beckmann's function $minimiz\ \sum_{i\in E}\int^{x_i} t_i(\omega)d\omega$, and the gradient evaluated at the feasible link flows $\overline{\boldsymbol{x}}$ is

$$\frac{d}{d\omega}\int^{\overline{x_i}} t_i(\omega)d\omega = t_i(\overline{x_i}), \forall i \in E$$

which coincidentally is the link travel time evaluated at current solution. The direction-finding problem has the objective to $\sum_{i\in E} t_i(\overline{x_i}) * x_i$, and it is essentially a series of min-cost flow problems with fixed link cost (a.k.a. link travel time) for each OD pair and without hard link capacity constraints. Hence the direction-finding problem can be solved by finding the shortest path for the OD pair and assign all the OD flow on the path found, which is the all-or-nothing assignment. Corresponding to the primary flows, the descending directions obtained in the direction-finding problem are called auxiliary flows.

With the primary and auxiliary link flows, a line search is performed to find the optimal convex combination of these two flows for the Beckmann's objective function. Once the optimal convex combination of the primary and auxiliary links flows is found, it will be the primary link flows for the next iteration. This procedure repeats until the link flows converge.

Inspired by the FW algorithm applied to the UE problem, we develop the FW algorithm tailored for the SO problem. Adopting the notations from the link-based OCREC, the SO problem can be formulated as follows:

$$minimize\ z_{SO}(\boldsymbol{x}) = \sum_{i\in E} t_i(x_i)x_i \quad (1)$$

$s.t.$:

$$x_i = \sum_{k\in OD} x_i^k \qquad \forall i \in E \quad (2)$$

$$D_k = \sum_{\{i:E_i^- = OD_k^-, i\in E\}} x_i^k - \sum_{\{j:E_j^+ = OD_k^-, j\in E\}} x_j^k \qquad \forall k \in OD \quad (3)$$

$$D_k = \sum_{\{i:E_i^+ = OD_k^+, i\in E\}} x_i^k - \sum_{\{j:E_j^- = OD_k^+, j\in E\}} x_j^k \qquad \forall k \in OD \quad (4)$$

$$\sum_{\{i:E_i^- = l, i\in E\}} x_i^k = \sum_{\{j:E_j^+ = l, j\in E\}} x_j^k \qquad \forall l \in N,\ \forall k \in \{k: OD_k^- \neq l\} \cap \{k: OD_k^+ \neq l\} \quad (5)$$

The objective is to minimize the total travel time of all OD flow. Constraint (2) ensures the link flow consistency. Constraint (3) to (5) are flow conservation constraints.

Suppose at n^{th} iteration, feasible flows $\bar{x}_i^k(n)\ \forall i \in E, \forall k \in OD$ are obtained. The gradient evaluated at $\bar{\boldsymbol{x}}_n$ is: $\nabla z(\bar{\boldsymbol{x}}_{\boldsymbol{n}}) = \boldsymbol{t}(\bar{\boldsymbol{x}}_n) + \frac{d\boldsymbol{t}(\bar{\boldsymbol{x}}_n)}{d\boldsymbol{x}}\bar{\boldsymbol{x}}_n$, where $\bar{\boldsymbol{x}}_n$ is the vector of link flows at n^{th} iteration and $\boldsymbol{t}(\bar{\boldsymbol{x}}_n)$ is the corresponding link travel time vector. If the BPR link travel time function is used, $z_{SO}(\boldsymbol{x}) = \sum_{i\in E} t_i^0 \left(1 + \alpha \left(\frac{x_i}{C_i}\right)^{\beta}\right) * x_i$, and $\nabla z(\bar{x}_i(n)) = t_i^0 \left(1 + \alpha(\beta+1)\left(\frac{\bar{x}_i(n)}{C_i}\right)^{\beta}\right)$ for $\forall i \in E$. To demonstrate the FW algorithm for the SO problem, the discussion forward adopts BPR function as the link travel time function. Let $\boldsymbol{y}_n$ denote the descending direction (a.k.a. auxiliary flows) for the feasible solution $\boldsymbol{x}_n$, then the <u>direction-finding problem</u> is:

$$minimize\ \boldsymbol{y}_n^T \nabla z(\bar{\boldsymbol{x}}_{\boldsymbol{n}}) = \sum_{i\in E} t_i^0 \left(1 + \alpha(\beta+1)\left(\frac{\bar{x}_i(n)}{C_i}\right)^{\beta}\right) y_i(n) \quad (1')$$

$s.t.$:

$$y_i = \sum_{k\in OD} y_i^k \qquad \forall i \in E \quad (2')$$

$$D_k = \sum_{\{i:E_i^- = OD_k^-, i\in E\}} y_i^k - \sum_{\{j:E_j^+ = OD_k^-, j\in E\}} y_j^k \qquad \forall k \in OD \quad (3')$$

$$D_k = \sum_{\{i:E_i^+ = OD_k^+, i\in E\}} y_i^k - \sum_{\{j:E_j^- = OD_k^+, j\in E\}} y_j^k \qquad \forall k \in OD \quad (4')$$

$$\sum_{\{i:E_i^-=l,i\in E\}} y_i^k = \sum_{\{j:E_j^+=l,j\in E\}} y_j^k, \forall l \in N, \forall k \in \{k: OD_k^- \neq l\} \cap \{k: OD_k^+ \neq l\} \qquad (5')$$

Since all parts other than $\boldsymbol{y}_n$ in the objective function (1') are fixed, this direction-finding problem can be perceived as a series of min-cost flow problems for the OD pairs with fixed link travel time $t_i^0\left(1+\alpha(\beta+1)\left(\frac{\bar{x}_\iota(n)}{C_i}\right)^\beta\right)$ for $\forall i \in E$. And this means $\boldsymbol{y}_n$ can be obtained by all-or-nothing assignment based on skewed link travel time $t_i^0\left(1+\alpha(\beta+1)\left(\frac{\bar{x}_\iota(n)}{C_i}\right)^\beta\right)$ for $\forall i \in E$. As a comparison, the true link travel time is $t_i^0\left(1+\alpha\left(\frac{\bar{x}_\iota(n)}{C_i}\right)^\beta\right)$ for $\forall i \in E$. The shortest paths of each OD pair for the all-or-nothing assignment can be found by a label correcting algorithm such as the Dijkstra algorithm. Let $\overline{\boldsymbol{y}}_n$ be the descending direction obtained from the direction-finding subproblem, then the <u>step-size problem</u> can be formulated as:

$$minimize\ z_{SO}(\lambda) = \sum_{i\in E} t_i^0\left[1+\alpha\left(\frac{\bar{x}_\iota(n)+\lambda[\bar{y}_\iota(n)-\bar{x}_\iota(n)]}{C_i}\right)^\beta\right]\{\bar{x}_\iota(n)+\lambda[\bar{y}_\iota(n)-\bar{x}_\iota(n)]\} \quad (6')$$

$$s.t.: \qquad \lambda \in (0,1)$$

The flow consistency and flow conservation constraints are not needed since both $\overline{\boldsymbol{x}}_{\boldsymbol{n}}$ and $\overline{\boldsymbol{y}}_{\boldsymbol{n}}$ satisfy these constraints and $\overline{\boldsymbol{x}}_n + \lambda[\overline{\boldsymbol{y}}_n - \overline{\boldsymbol{x}}_n]$ is a convex combination of these two sets of flows. The step size problem is solved by a quadratic approximation method, which is a line search method that approximates the objective function using a quadratic function based on the value of λ, the corresponding objective value and the first derivatives of the objective function evaluated at the value of λ. New λ value is obtained by optimizing the quadratic function within the range defined by the previous λ values. As more iterations being computed, the range defined by previous λ values keeps contracting until the lower bound and upper bound of the range nearly coincide. The optimal λ for the step-size problem is the lower bound and/or the upper bound.

As to the convergence of FW developed for the SO problem, since the objective function of the SO problem is convex (its Hessian matrix is positive definite) and so is the feasible region (intersection of hyperplanes), the SO problem is convex. Hence for any two feasible flows $\boldsymbol{x}$ and $\boldsymbol{y}$, we have: $z_{SO}(\boldsymbol{y}) \geq z_{SO}(\boldsymbol{x}) + (\boldsymbol{y}-\boldsymbol{x})^T \nabla z_{SO}(\boldsymbol{x})$. Suppose the optimal flow for the SO problem is $\boldsymbol{x}^*$, we have: $z_{SO}(\boldsymbol{x}^*) \geq z_{SO}(\overline{\boldsymbol{x}}) + (\boldsymbol{x}^* - \overline{\boldsymbol{x}})^T \nabla z_{SO}(\overline{\boldsymbol{x}})$ where $\overline{\boldsymbol{x}}$ is a set of feasible link flows. Let F be the feasible region defined by constraints from (2) to (5), since

$$z_{SO}(\overline{\boldsymbol{x}}) + (\boldsymbol{x}^* - \overline{\boldsymbol{x}})^T \nabla z_{SO}(\overline{\boldsymbol{x}}) \geq \min_{\{\boldsymbol{y}\in F\}} \{z_{SO}(\overline{\boldsymbol{x}}) + (\boldsymbol{y} - \overline{\boldsymbol{x}})^T \nabla z_{SO}(\overline{\boldsymbol{x}})\}$$

$$= z_{SO}(\overline{\boldsymbol{x}}) - \overline{\boldsymbol{x}}^{\mathrm{T}} \nabla z_{SO}(\overline{\boldsymbol{x}}) + \min_{\{\boldsymbol{y}\in F\}} \boldsymbol{y}^T \nabla z_{SO}(\overline{\boldsymbol{x}})$$

we have:

$$z_{SO}(\boldsymbol{x}^*) \geq z_{SO}(\overline{\boldsymbol{x}}) - \overline{\boldsymbol{x}}^{\mathrm{T}} \nabla z_{SO}(\overline{\boldsymbol{x}}) + \min_{\{\boldsymbol{y}\in F\}} \boldsymbol{y}^T \nabla z_{SO}(\overline{\boldsymbol{x}})$$

where $\min_{\{\boldsymbol{y}\in F\}} \boldsymbol{y}^T \nabla z_{SO}(\overline{\boldsymbol{x}})$ is the objective of the direction-finding subproblem. Therefore, $z_{SO}(\overline{\boldsymbol{x}}) - \overline{\boldsymbol{x}}^{\mathrm{T}} \nabla z_{SO}(\overline{\boldsymbol{x}}) + \min_{\{\boldsymbol{y}\in F\}} \boldsymbol{y}^T \nabla z_{SO}(\overline{\boldsymbol{x}})$ is the lower bound obtained at $\overline{\boldsymbol{x}}$. Let LB_{n-1} be the lower bound obtained at iteration $n-1$, we have:

$$LB_n = \max\{LB_{n-1},\ z_{SO}(\overline{\boldsymbol{x}}_{\boldsymbol{n}}) - \overline{\boldsymbol{x}}_{\boldsymbol{n}}^{\boldsymbol{T}} \nabla z_{SO}(\overline{\boldsymbol{x}}_{\boldsymbol{n}}) + \overline{\boldsymbol{y}}_{\boldsymbol{n}}^T \nabla z_{SO}(\overline{\boldsymbol{x}}_{\boldsymbol{n}})\}$$

The upper bound for the SO problem at n^{th} iteration is simply the objective value evaluated at $\overline{\boldsymbol{x}}_{\boldsymbol{n}}$, that is $UB_n = z_{SO}(\overline{\boldsymbol{x}}_{\boldsymbol{n}})$.With the continuously improving upper bound and lower bound of SO, the solution procedure will end when the optimality gap is smaller than a predefined value. It has been shown that the FW algorithm converges at the rate

$O\left(\frac{1}{n}\right)$. (Frank and Wolfe, 1956) Therefore, the Frank-Wolfe (FW) algorithm for the system optimum (SO) problem is convergent and can be summarized as:

Step 1: $n = 0$, perform all-or-nothing assignment based on free-flow travel times and obtain $\boldsymbol{x}_0$. $LB_0 = 0$.

Step 2: Calculate $\nabla z(\overline{\boldsymbol{x}}_{\boldsymbol{n}})$ and solve the direction-finding subproblem with a shortest path algorithm. Obtain auxiliary flows $\overline{\boldsymbol{y}}_n$.

Step 3: Update $LB_n = \max\{LB_{n-1},\ \ z(\overline{\boldsymbol{x}}_n) - \overline{\boldsymbol{x}}_n^{\boldsymbol{T}}\nabla z(\overline{\boldsymbol{x}}_n) + \overline{\boldsymbol{y}}_n^T\nabla z(\overline{\boldsymbol{x}}_n)\}$. If $\frac{|z(\overline{x}_n)-LB_n|}{LB_n} \leq \epsilon$, optimal flow has been find, exit the algorithm; otherwise, go to step 4.

Step 4: Solve the step size subproblem, obtain λ and let $\overline{\boldsymbol{x}}_{n+1} = \overline{\boldsymbol{x}}_n + \lambda[\overline{\boldsymbol{y}}_n - \overline{\boldsymbol{x}}_n]$, which is the set of primary flows for the next iteration. Go back to step 2.

Since the search directions of the FW method tend to become orthogonal to the gradient as the solution gets close to the optimum, FW method usually converges slowly when the optimality gap is less than 10^{-4} because of the extremely zigzagging effect. To obtain search directions that are not orthogonal to the gradient, Mitradjieva and Lindberg (2013) developed a conjugate Frank-Wolfe (CFW) algorithm which utilizes the search directions obtained in last two iterations, and bi-conjugate Frank-Wolfe (BFW) algorithm taht incorporates the search directions obtained in last three iterations. The BFW was implemented in this research for faster convergence in the solving the SO problem.

4.5 Computational Experiments

The entire heuristic for link-based OCREC was implemented in C++. The C++ implementation of TAPAS algorithm for the UE problem is adopted from the UE algorithm package developed by Perederieieva et al. (2015). To get a baseline for the heuristic for comparison purposes, the heuristic was also implemented in AMPL® and uses the nonlinear solver MINOS® to solve the UE problems and SO problems. The optimality tolerance for the UE problem was set to 10^{-8} for both TAPAS algorithm and MINOS, and the optimality tolerance for the SO problem was set to 10^{-6} for both BFW algorithm and

MINOS. The reason for this configuration is because smaller optimality gap is required for more accurate UE flows and it can be obtained by the TAPAS algorithm within a short amount of time. And BFW struggles to reduce the optimality gap when it is less than 10^{-6} for the problems instances tested. All the computational experiments were conducted on a personal computer with a 3.7 GHz quad-core CUP and 24.0 GB memory.

First, both the C++ and AMPL implementation are tested on the original problem instance of the simple four-node network shown in Figure 5.5-i, which is solved by Knitro successfully for the path-based OCREC. The solution given by Knitro is to reduce the capacity of link 5 by 59.999 units, and he total travel time at UE after the capacity reduction is 3042.555. As it is shown in table 4.5-i, both the C++ implementation and the AMPL implementation give the same solution as the solution given by Knitro, but the AMPL implementation solves this problem instance a little faster.

Table 4.5-i: C++-TAPAS-BFW vs. AMPL-MINOS on Four-Node Network

Implementation Methods		C++-TAPAS-BFW	AMPL-Minos
Total Travel Time at UE	Before Capacity Reduction	3066.637	3066.637
	After Capacity Reduction	3042.555	3042.555
	% Reduced	0.79%	0.79%
Total Travel Time at SO	Before Capacity Reduction	2901.54	2901.54
	After Capacity Reduction	3042.552	3042.552
Computation Time		0.343 sec	0.2 sec
Link Capacity Reduction		CapRed[5]=59.999	CapRed[5]=59.999

Then, the square network problem instance, which Knitro fails to solve for the path-based model as discussed in Section 4.3.2, is solved by the C++ implementation and the AMPL implementation for the link-based model. The solutions obtained by the two implementations are given in Table 4.5-ii.

Table 4.5-ii: C++-TAPAS-BFW vs. AMPL-MINOS on Square Network

Implementation Methods		C++-TAPAS-BFW	AMPL-Minos
Total Travel Time at UE	**Before Capacity Reduction**	5137807.64	5137807.866
	After Capacity Reduction	5008575.52	5008577.066
	% Reduced	2.52%	2.52%
Total Travel Time at SO	**Before Capacity Reduction**	4729754.84	4729753.387
	After Capacity Reduction	4936195.04	4938203.792
Computation Time		4.32 min	0.35 min
Link Capacity Reduction		CapRed[2]=499.999	CapRed[2]=499.999
		CapRed[13]=599.999	CapRed[13]=599.999
		CapRed[15]=213.755	CapRed[15]=225.237
		CapRed[16]=599.999	CapRed[16]=599.999
		CapRed[29]=599.999	CapRed[25]=0.103
		CapRed[32]=99.349	CapRed[26]=29.881
		CapRed[42]=599.999	CapRed[29]=599.999
		CapRed[45]=599.999	CapRed[32]=117.858
		CapRed[53]=599.999	CapRed[42]=599.999
		CapRed[55]=599.999	CapRed[45]=599.999
			CapRed[53]=599.999
			CapRed[55]=599.999

It can be observed in Table 5.5-ii that the solutions given by these two implementations are very similar but the C++ implementation took much longer to solve the problem instance. Comparing the performance of the heuristic for the link-based model to the approach that uses Knitro to solve the path-based model, for the square network with 12 paths found for each OD pair, it took Knitro hours to obtain a capacity reduction scheme that decreased the total UE travel time to 5135620. Although this total UE travel time is less than the total travel time of 5137808 for the original network, meaning Knitro does find an effective capacity reduction scheme that improves the efficiency of UE flows, it is much higher than the UE travel time from the solutions obtained by the heuristic developed for the link-based model. Hence, it is concluded that the heuristic for the link-based model is much more efficient and effective than using Knitro to solve the path-based model.

Another problem instance is created for the same square network by doubling the original travel demand for each OD. The solution from the two implementations are summarized in Table 4.5-iii.

Table 4.5-iii: C++-TAPAS-BFW vs. AMPL-MINOS on Square Network with Demand Doubled

Implementation Methods		C++-TAPAS-BFW	AMPL-Minos
Total Travel Time at UE	**Before Capacity Reduction**	10990702.5	10990705.51
	After Capacity Reduction	10473795.9	10473802.92
	% Reduced	4.70%	4.70%
Total Travel Time at SO	**Before Capacity Reduction**	10333230.5	1033223.488
	After Capacity Reduction	10473804.7	2.03E+24
Computation Time		5.38 min	0.08 min
Link Capacity Reduction		CapRed[10]=599.999	CapRed[10]=599.999
		CapRed[13]=599.999	CapRed[13]=599.999
		CapRed[16]=599.999	CapRed[16]=599.999
		CapRed[19]=599.999	CapRed[19]=599.999
		CapRed[23]=599.999	CapRed[23]=599.999
		CapRed[26]=599.999	CapRed[26]=599.999
		CapRed[29]=599.999	CapRed[29]=599.999
		CapRed[32]=599.999	CapRed[32]=599.999
		CapRed[36]=599.999	CapRed[36]=599.999
		CapRed[39]=599.999	CapRed[39]=599.999
		CapRed[42]=599.999	CapRed[42]=599.999
		CapRed[45]=599.999	CapRed[45]=599.999
		CapRed[49]=599.999	CapRed[49]=599.999
		CapRed[51]=599.999	CapRed[51]=599.999
		CapRed[53]=599.999	CapRed[53]=599.999
		CapRed[55]=599.999	CapRed[55]=599.999

Both implementation give the same link capacity reductions and AMPL implementation is much faster than the C++ implementation. However, the BFW algorithm outperforms MINOS in solving the SO problem. The MINOS effectively gives up and returns an SO solution with total travel time of 2.03×10^{24}, which is nowhere close to the total travel time at SO before the capacity reduction, whereas BFW gives the SO that is very close to the UE after capacity reduction. Although the total travel time at SO after the capacity reduction is still larger than the total travel time at UE for the C++ implementation, which should not happen since SO is supposed to minimize total travel time, the difference on total travel times of these two flow patterns is less than 10^{-6} and can be considered to be practically same. Another the reason that the total travel time at SO is slightly larger than that at UE is because the optimality tolerance for the SO problem is 10^{-6}; if the optimality tolerance is set much smaller (i.e., 10^{-8}), the BFW could obtain an SO with total travel time less than that at UE, but would require significantly longer computation time.

Besides the square network, both implementations were also tested on the Sioux Falls network shown in Figure 3.4.3-iii in Chapter 3, which is a real-world network with 76 links and 528 OD pairs. The detailed information of Sioux network test case is not attached in the appendix since it is a widely used test case in research and can be found online.

Table 4.5-iv compares the solution given by the C++ implementation and the AMPL implementation. Both implementations conclude no link capacity reduction can improve the efficiency of the UE flows. However, the total travel times at UE given by these two implementations differ significantly, and so do the total travel times at SO. To verify which of the two implementations gave the correction solution, the same Sioux Falls test case is solved with the bi-conjugate Frank-Wolfe (BFW) algorithm for the UE problem implemented in Julia, which is developed by Kwon (2017). With optimality tolerance set

to 10^{-6}, the total travel time at UE obtained by the Julia implementation of BFW is 7480217.15, matching the result from our C++ implementation. And thus, it may be concluded that MINOS gave the wrong solution. Also, it can be seen from Table 4.5-iv that the time consumptions of these two implementations are getting closer, indicating the efficiency of MINOS is affected by the scale of the problem instance and deteriorates quickly as the dimension of the problem instance increases.

Table 4.5-iv: C++-TAPAS-BFW vs. AMPL-MINOS on Sioux Falls Network

Implementation Methods		C++-TAPAS-BFW	AMPL-Minos
Total Travel Time at UE	**Before Capacity Reduction**	7480224.53 *	5137687.454
	After Capacity Reduction	No effective link capacity reduction	No effective link capacity reduction
	% Reduced	NA	NA
Total Travel Time at SO	**Before Capacity Reduction**	7194258.56	4902374.426
	After Capacity Reduction	No effective link capacity reduction	No effective link capacity reduction
Computation Time		3.82 min	1.63 min
Link Capacity Reduction		NA	NA

*The total travel time at UE before capacity reduction matches results from the Julia implementation of BFW, which is 7480217.15

The C++ implementation was also experimented on the Sioux Falls network after the OD demands were reduced by half. The solution from the C++ implementation shows that the efficiency of UE flows can be improved by 0.1655% with capacity reductions on link 22 and 47. For the same problem instance, the AMPL implementation gives a totally different solution that concludes no capacity reduction can improve the efficiency of the UE flows. What's more, the total travel time at UE when there is no capacity reduction given by the AMPL implementation deviates far from that given by the C++ implementation. Based on the validation just conducted on the Sioux Falls with the original OD demand, it is concluded that AMPL implementation gave the wrong solution again. Solutions for this test case from the two implementations are summarized in Table 4.5-v.

Table 4.5-v: C++-TAPAS-BFW vs. AMPL-MINOS on Sioux Falls Network with Demand Reduced by Half

<table>
<tr><th colspan="2">Implementation Methods</th><th>C++-TAPAS-BFW</th><th>AMPL-Minos</th></tr>
<tr><td rowspan="3">Total Travel Time at UE</td><td>Before Capacity Reduction</td><td>1870591.65</td><td>1637771.024</td></tr>
<tr><td>After Capacity Reduction</td><td>1867495.14</td><td>No effective link capacity reduction</td></tr>
<tr><td>% Reduced</td><td>0.1655%</td><td>NA</td></tr>
<tr><td rowspan="2">Total Travel Time at SO</td><td>Before Capacity Reduction</td><td>1815464.81</td><td>1585060.691</td></tr>
<tr><td>After Capacity Reduction</td><td>1823643.35</td><td>No effective link capacity reduction</td></tr>
<tr><td colspan="2">Computation Time</td><td>3.85 min</td><td>1.61 min</td></tr>
<tr><td colspan="2" rowspan="2">Link Capacity Reduction</td><td>CapRed[22]=854.948</td><td rowspan="2">No effective link capacity reduction</td></tr>
<tr><td>CapRed[47]=2892.079</td></tr>
</table>

Besides the Sioux Falls network, the C++ and AMPL implementations are also tested on the Anaheim network shown in Figure 4.5-vi on the next page, which is a much larger real-world network test case with 914 links and 1406 OD pairs. The detailed information of the Anaheim network and the OD demand can also be found online and thus is not attached in the appendix.

The C++ implementation solved this problem instance in 8 minutes and obtained a capacity reduction scheme that improved the UE flow efficiency by 0.24%. As a comparison, the AMPL implementation spends 58.3 hours on solving this problem instance and returns a solution that is obviously wrong: the total travel time at SO after capacity reduction is less than that before the capacity reduction. This cannot happen for the SO problem which is a convex minimization problem. Since the objective function of SO is to minimize the total travel time, the total travel time will not decrease after the capacity reductions because these capacity reductions will increase the link travel times.

If no OD flow is traveling on the link that has capacity reduction, the total travel time will stay the same; otherwise it will increase because of the steeper curve of the link travel time to the link flow. Also, the total travel times at the UE from the AMPL implementation and the UE from the C++ implementation differ considerably. The solutions from both implementations are summarized in Table 4.5-vi on the next page.

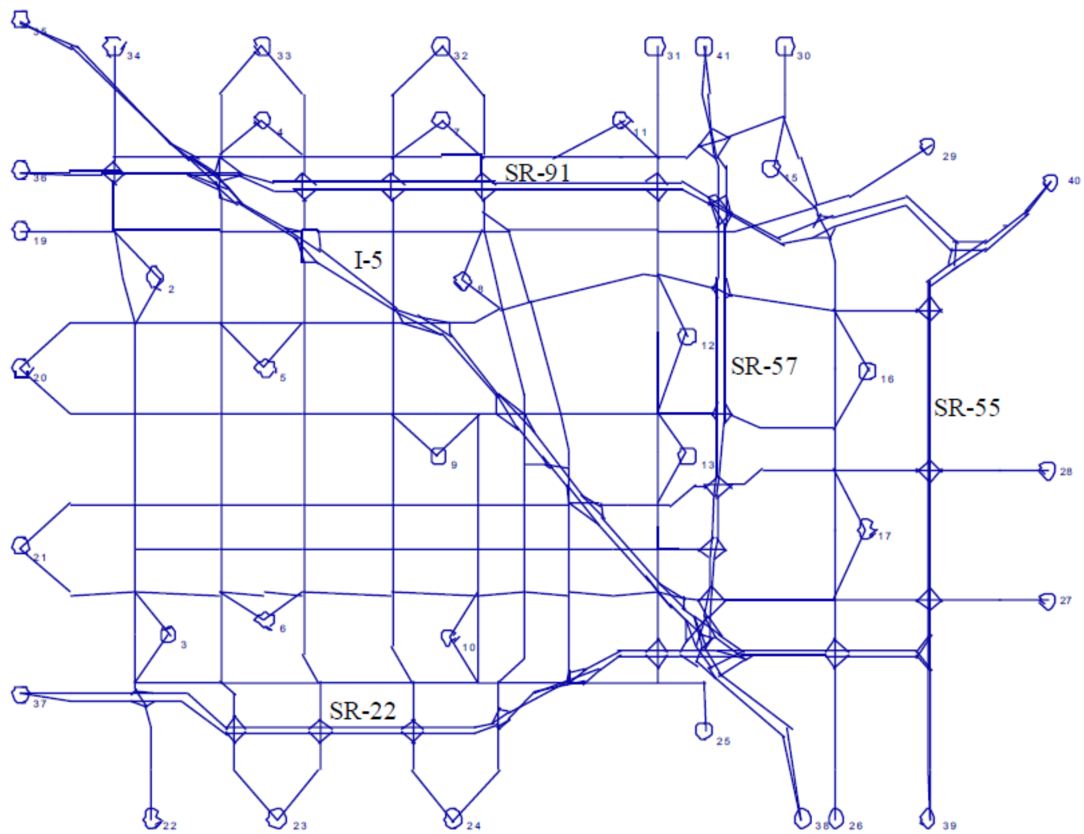


Figure 4.5-i: Anaheim Network

The comparison between these two implementations demonstrates how sensitive a nonlinear commercial solver is to the scale of the problem it tries to solve. In our case, nonlinear solvers like MINOS and Knitro are not able to solve UE problems and SO problems correctly for networks of moderate size (i.e., the Sioux Falls network). The efficiency of the C++ implementation is resistant to the dimension escalation of the problem instances. And this is because it adopts solution methods that exploit the structures of the specific nonlinear optimization problem, and that are particularly developed based on the features of the problem (i.e., TAPAS for the UE problem and BFW for the SO problem).

Table 4.5-vi: C++-TAPAS-BFW vs. AMPL-MINOS on Anaheim Network

Implementation Methods		**C++-TAPAS-BFW**	**AMPL-Minos**
Total Travel Time at UE	**Before Capacity Reduction**	1419914.03	1337004
	After Capacity Reduction	1416527.17	1098210
	% Reduced	0.24%	17.86%
Total Travel Time at SO	**Before Capacity Reduction**	1398386.57	1173096
	After Capacity Reduction	1399015.40	1018116
Computation Time		7.98 min	58.3 hr
Link Capacity Reduction		CapRed[115]=1799.999	CapRed[124]=3156.8
		CapRed[187]=44.318	CapRed[130]=244.815
		CapRed[207]=365.729	CapRed[132]=981.843
		CapRed[218]=3085.825	CapRed[133]=2814.85
		CapRed[230]=2449.265	CapRed[134]=500.832
		CapRed[255]=1799.999	CapRed[162]=7858.65
		CapRed[291]=996.883	CapRed[208]=1786.13
		CapRed[297]=68.6416	CapRed[210]=702.264
		CapRed[304]=33.434	CapRed[250]=279.913
		CapRed[307]=35.541	CapRed[283]=1799.9
		CapRed[479]=1491.557	...

4.6 Conclusion

This chapter demonstrates the existence of Braess paradox when links are not entirely removed from the network but the capacity is reduced by a certain amount, which can be considered as the generalized version of the Braess paradox. With user equilibrium (UE) flows, the generalized Braess paradox can be applied to network-level traffic management through selective link capacity reductions. OCREC aims at identifying the links whose capacity reduction can reduce the total travel time at UE, and finding the optimal amount of capacity to decrease for the links identified.

Both path-based model and link-based model are developed for OCREC. The path-based OCREC model has the objective of minimizing the total flow cost at UE after link capacity reductions, and has complementarity constraints to ensure the UE flow condition. The nonlinear solver Knitro is used to solve the path-based OCREC but can only handle very small problem instances. The link-based OCREC is a bi-level model where the upper level determines the link capacity reductions that minimizes the total travel time at UE, and the lower level finds the UE flow under a certain capacity reduction scenario. A heuristic is designed for the link-based OCREC to find a good combination of links and the desired capacity reduction amount on the links found. The heuristic compares the link flows at UE and SO, identifies the links that are overused by the UE flows, and attempts capacity reductions on these links. If a more efficient UE is obtained, the capacity reduction will be accepted and the UE and SO will be recalculated based on the accepted link capacity reductions. The heuristic stops when no further capacity reduction on any of the links can improve the efficiency at UE. Inside the heuristic, the TAPAS algorithm is adopted to solve the UE problem and a bi-conjugate Frank-Wolfe (BFW) algorithm is developed to solve the SO problem. The heuristic is implemented in C++ and compared with the implementation in AMPL with MINOS to solve UE and SO. Computational results on various test cases show that the C++ implementation with TAPAS and BFW is much more resistant to the escalation of problem size, and can give correct solutions to large problem instances where the AMPL implementation fails.

Since OCREC is a static model, its goal is to influence commuters' routing habits by reducing the road capacities, so as to improve the overall efficiency of the traffic flows. The link capacity reduction can be achieved through traffic control methods like traffic light phase adjustment, ramp meter phase adjustment, speed limit change, etc. A direction for possible future research is to investigate how these link capacity reductions will affect the dynamic evolution of traffic flows during the day. And design a system to dynamically identify links and the amount of capacity reductions to alleviate traffic congestion in real-time based on a network-wide perspective.

Chapter 5

MAINTENANCE SCHEDULING IN MULTI-MODAL NETWORKS (MS-MMN)

5.1 Introduction

In large cities, people often have the options of traveling to their destinations through different transportation modes, such as private cars, buses, light-rails, ride-sharing cars/vans, autonomous vehicles (in the near future), etc. Different travel modes serve portions of the origin-destination (OD) demands and/or compete for the same transportation infrastructure (i.e., road network). For the multi-modal traffic that competes for the road capacity, numerous studies have investigated the mixed flows of cars and trucks (e.g., Bliemer, 2000; Chanut and Buisson, 2003; Ferrari, 2009; Ferrari, 2011; Mesa-Arango and Ukkusuri, 2014; Wu et al., 2006; Zhang et al., 2002; etc.). As greater traffic of electric vehicles and self-driving cars being predicted, more research attention has been drawn to the multi-modal traffic consisting gasoline vehicles and electric vehicles (e.g., Agrawal et al., 2016; Jiang and Xie, 2014; Xu et al., 2017), and the mixed flows of human-driving vehicles and autonomous vehicles (e.g., Davis, 2007; Mahmassani, 2016). These studies, albeit innovative, are limited to the assumption that all traffic flows of different travel modes are user equilibrium (UE) flows as described in Wardrop's First Principle (Wardrop, 1952), where every traveler routes through the network to minimize his/her own travel time.

This chapter studies the mixed flow of two travel modes where the travelers of each mode have distinct routing objectives. Travelers of the first travel mode (i.e., private cars) choose the routes that minimize individual travel times and reach user equilibrium. And the travelers of the second travel mode choose the routes that minimize the overall travel time of all travelers and achieve system optimum (SO). One example of such travel mode is autonomous vehicles mode where the route to take passengers may be decided centrally.

As discussed in Section 2.3, literature reviewed on maintenance scheduling in transportation networks only considered single mode traffic flows -- either pure UE flows or pure SO flows. This chapter makes the first attempt to investigate the maintenance scheduling problem with the consideration of multi-modal traffic flows that consist of both UE flows and SO flows. To approach this problem, a bi-level optimization model is developed in the next section, where the upper level is a scheduling problem and the lower level are a series of UE flow and SO flow assignment problems for each day in the planning horizon based on a feasible schedule. An iterative UE-SO assignment algorithm is developed for the lower level problem in Section 5.3. Section 5.4 applies the genetic algorithm to solve the problem of maintenance scheduling in multi-modal networks (MS-MMN). The computational experiments conducted on various test cases are summarized in Section 5.5. The research findings presented in this chapter are summarized in Section 5.6.

5.2 MS-MMN Model

In the problem of maintenance scheduling in multi-modal networks (MS-MMN), a set of links need to be repaired before a common due date and each lane of these links can constitute an independent work zone to be scheduled. Once a lane is closed for repair, it cannot open to serve flows until it is repaired. Upon maintenance completion, lanes will have a small capacity increase since it is commonsense that the road condition should be improved and the capacity should increase after maintenance. The available capacity of the links may change from day to day due to closing lanes for repair and reopening lanes that are repaired. On each day in the planning horizon, there are some OD flows which are UE flows and other OD flows that are SO flows. They route through the network based on the available link capacities on each day. The objective of the MS-MMN problem is to schedule lane closures so that all maintenance work can be completed before the common due date, and the total travel time of all OD flows are minimized in the planning time horizon.

The MS-MMN is formulated as a bi-level mixed integer nonlinear program. The upper level is the scheduling problem that obtains lane closure schedules. Denote y_{it} as the total flow from all OD demand on link i on day t, and $c_i(y_{it})$ as the travel time function of link i evaluated at y_{it}, the objective of the upper level problem, which is also the objective of the MS-MMN, is $minimize \sum_{t\in[1,T]} \sum_{i\in E} c_i(y_{it}) * y_{it}$, where T is the maintenance completion date and E is the link set in the network.

Binary variables s_{imt} are introduced to indicate whether the repair of the m^{th} lane of link i starts on day t, and $s_{imt} = 1$ if it is. Hence, we have the constraints $\sum_{t=1}^{t=T} s_{imt} = 1$ for $\forall i \in R$ and $\forall m \in [1, n_i]$, where R is the set of links that need repair and n_i is the number of lanes in link i. This set of constraints force every lane of all the links that need repair to have one and only one repair start date.

To indicate whether m^{th} lane of link i is closed for maintenance on day t, binary variables x_{imt} are added to the model and it equals to 1 if the lane is closed. Let p_i be the number of days needed to repair a lane of link i, constraints $\sum_{t=1}^{t=T} x_{imt} = p_i$ for $\forall i \in R$ and $\forall m \in [1, n_i]$ are formulated to ensure the repair on all the links be completed by the common due date T. Since each lane of the links that need maintenance have one and only one repair start date and the number of days needed to repair a lane is given, whether a lane is closed or not on a day is determined once the repair start date of that lane is determined. And thus, we develop the set of constraints $x_{imt} = \sum_{a=max(t-p_i+1,1)}^{a=t} s_{ima}$ for $\forall i \in R, \forall t \in T$ and $\forall m \in [1, n_i]$ to make sure that once a lane is closed for repair, it will not open to serve traffic flows until the repair work on this lane is finished and that it will be open on other dates. Constraints $\sum_{t=1}^{t=T} s_{imt} = 0$ for $\forall i \notin R, \forall m \in [1, n_i]$ and $\sum_{t=1}^{t=T} x_{imt} = 0$ for $\forall i \notin R$ and $\forall m \in [1, n_i]$ are added to the model so that all the lanes of links that do not need repair will not have maintenance start date and will be open to serve the flows throughout the project period.

In addition, binary variables v_{imt} are introduced to calculate the increased lane capacities and v_{imt} equals to 1 if lane m of link i is repaired before day t. Constraints $v_{imt} = \sum_{a=1}^{a=t-p_i} s_{ima}$, for $\forall i \in R, \forall m \in [1, n_i]$ and $\forall t \in [p_i + 1, T]$ determine the values of v_{imt} given the values of s_{imt}. In the constraints, the date ranges from $p_i + 1$ to T since $p_i + 1$ is the earliest day that the lane can open and serve traffic flows, because even if the maintenance starts on day 1, it would take p_i days to complete the repair work for this lane. Constraints $v_{imt} = 0$, for $\forall i \in R, \forall m \in [1, n_i]$ and $\forall t \in [1, p_i]$ make sure each lane of the links that need maintenance stay in the status of not repaired in the first p_i days. And constraints $v_{imt} = 0$, for $\forall i \notin R, \forall m \in [1, n_i]$ and $\forall t \in [1, T]$ force lanes of links that do not need repair stay in the status quo throughout the project period.

Let θ be the percentage of capacity increase after a lane is repaired, and let u_i be the lane capacity of link i, then the available capacity of link i on day t is $\left(n_i - \sum_{m=1}^{n_i} x_{imt} + \sum_{m=1}^{n_i} \theta v_{imt}\right) u_i$. Although there is no constraint based on link capacity being explicitly formulated in MS-MMN, link overflow is contained by adopting link travel time functions that increase exponentially once the link flow exceeds the link available capacity. One example of this type of link travel time function is the function developed by Bureau of Public Roads (BPR), which is:

$$c_i(y_{it}) = c_i^0\left[1+\alpha\left(\frac{y_{it}}{u_i\left(n_i - \sum_{m=1}^{n_i} x_{imt} \; + \theta \sum_{m=1}^{n_i} v_{imt}\right)}\right)^{\beta}\right]$$

where c_i^0 is the free-flow travel time on link i and α and β are parameters. This BPR function is adopted as the link travel time function for in this chapter.

Denote y_{it}^{UE} as the total flow from all OD pairs that generate UE flows, and denote y_{it}^{SO} as the total flow from all OD pairs that generate SO flows, the flow consistency constraints $y_{it} = y_{it}^{UE} + y_{it}^{SO}$ are formulated for $\forall i \in E, \forall t \in [1, T]$ with the presumption that each unit of UE flow has the same effect on the link travel time as each unit of SO flow does. Denote D_k^{UE} as the UE flow and D_k^{SO} as the SO flow generated by OD pair k respectively, constraint $y_{it} \leq \left(\sum_{k \in OD^{UE}} D_k^{UE} + \sum_{k \in OD^{SO}} D_k^{SO}\right)\left(n_i - \sum_{m=1}^{n_i} x_{imt}\right)$ is added for $\forall i \in R, \forall t \in [1,\ T]$ to ensure entirely closed links not to serve any flows.

The lower-level UE flow assignment problem and SO flow assignment problem are formulated for each day in the planning horizon. For a specific day, the objective of the UE assignment problem is the Beckmann's function $minimize\ \sum_{i \in E} \int^{y_{it}^{UE}} c_i(\omega) d\omega$ that ensures the UE flow condition. The flow consistency constraint $y_{it}^{UE} = \sum_{k \in OD^{UE}} y_{ikt}^{UE}$ is added for $\forall i \in E$ so that the UE flows from all OD pairs are accounted for the total UE flow on link i.

For each OD pair that generates UE flows on each day, flow conservation constraints, consisting of three groups, are needed. The first group of constraints makes sure the total incoming UE flow units minus the total outgoing UE flow units equal to the OD demand for the <u>origin node</u> of the OD pair. The first part is formulated as $D_k^{UE} = \sum_{\{i: E_i^- = OD_k^{UE-}, i \in E\}} y_{ikt}^{UE} - \sum_{\{j: E_j^+ = OD_k^{UE-}, j \in E\}} y_{jkt}^{UE}$ for $\forall k \in OD^{UE}$, where y_{jkt}^{UE} is the UE flow of OD pair k on link j on day t, OD_K^{UE-} is the origin node of OD pair k that generates the UE flow, E_i^- is the head node of link i and E_j^+ is the tail node of link j. The second group ensures the total outgoing UE flow units minus the total incoming UE flow units equal to the demand of OD pair k for its destination node, and is formulated as $D_k^{UE} = \sum_{\{i:E_i^+=OD_k^{UE+},i\in E\}} y_{ikt}^{UE} - \sum_{\{j:E_j^-=OD_k^{UE+},j\in E\}} y_{jkt}^{UE}$ for $\forall k \in OD$, where OD_K^{UE+} is the destination node of OD pair k that generates the UE flow, E_i^+ is the tail node of link i and E_j^- is the head node of link j. For the rest of the nodes, other than origin and destination nodes of OD pair k, the total incoming UE flows on the node from the origin of OD pair k should equal to the total outgoing UE flows from the node to the destination of the OD pair k. This is the third group of the flow conservation constraints and it is formulated as $\sum_{\{i:E_i^-=l,i\in E\}} y_{ikt}^{UE} = \sum_{\{j:E_j^+=l,j\in E\}} y_{jkt}^{UE}$ for $\forall l \in N, \forall k \in \{k: OD_k^{UE-} \neq l\} \cap \{k: OD_k^{UE+} \neq l\}$, where N is the set of nodes in the network.

As to the SO assignment problem on each day, the objective function is $minimize \sum_{i\in E} c_i(y_{it}) * y_{it}$, which is to have the SO flows to choose the routes that will minimize the total travel time of all the OD flows. It has flow consistency constraints and flow conservation constraints that are similar to those of the UE assignment problem, but are formulated with respect to the SO flows and OD pairs that generate SO flows.

The aforementioned sets, parameters, variables and functions are listed in Table 5.2 – i:

Table 5.2-i: Notations for MS-MMN

Term	Definition
Sets	
$\boldsymbol{N}$	Node set of the network
$\boldsymbol{E}$	The set of existing links in the network
$\boldsymbol{R}$	The set of existing links that need to be repaired in the network, $R \subseteq E$
$\boldsymbol{OD^{UE}}$	The set of Origin-Destination pairs of UE flows

Term	Definition
Sets	
$\boldsymbol{OD^{SO}}$	The set of Origin-Destination pairs of SO flows
Parameters	
$\boldsymbol{T}$	Completion date for all the maintenance work (the earliest start date of a work zone is Day 1)
$\boldsymbol{n_i}$	Number of lanes of link i, $i \in E$
$\boldsymbol{u_i}$	Capacity of a lane of link i, $i \in E$
$\boldsymbol{\theta}$	The percentage of capacity increase after a lane is repaired
$\boldsymbol{c_i^0}$	The free-flow travel time on link i, $i \in E$
$\boldsymbol{p_i}$	The number of days needed to repair a lane of link i, $i \in R$
$\boldsymbol{E_i^-}$	Start node of link i, $i \in E$
$\boldsymbol{E_i^+}$	End node of link i, $i \in E$
$\boldsymbol{OD_k^{UE-}}$	Origin node of OD pair k, $k \in OD^{UE}$
$\boldsymbol{OD_k^{UE+}}$	Destination node of OD pair k, $k \in OD^{UE}$
$\boldsymbol{OD_k^{SO-}}$	Origin node of OD pair k, $k \in OD^{SO}$
$\boldsymbol{OD_k^{SO+}}$	Destination node of OD pair k, $k \in OD^{SO}$
$\boldsymbol{D_k^{UE}}$	Flow demand of OD pair k, $k \in OD^{UE}$
$\boldsymbol{D_k^{SO}}$	Flow demand of OD pair k, $k \in OD^{SO}$
Variables	
$\boldsymbol{s_{imt}}$	Binary variable indicating whether to repair on the m^{th} lane of link i starts on day t. If repair work starts on day t, $s_{imt} = 1$; otherwise, $s_{imt} = 0$
$\boldsymbol{x_{imt}}$	Binary variable indicating whether the m^{th} lane of link i is closed for maintenance on day t, if it is closed, $x_{imt} = 1$; otherwise $x_{imt} = 0$
$\boldsymbol{y_{ikt}^{UE}}$	The flow units incurred by the UE flow of OD pair k on link i on day t
$\boldsymbol{y_{it}^{UE}}$	The flow units from all UE flows on link i on day t
$\boldsymbol{y_{ikt}^{SO}}$	The flow units incurred by the SO flow of OD pair k on link i on day t
$\boldsymbol{y_{it}^{SO}}$	The flow units from all SO flows on link i on day t
$\boldsymbol{y_{it}}$	The total amount of flows on link i on day t from all UE and SO OD pairs
$\boldsymbol{v_{imt}}$	Binary variable indicating whether the m^{th} lane of link i is repaired before day t, if it is, $v_{imt} = 1$, otherwise 0; for all the links that don't need maintenance, $v_{imt} = 0$ all the time

Term	Definition
Functions	
$\boldsymbol{c_i(y_{it})}$	Travel time on link i when the flow on the link is y_{it}. BPR function is used, for $\forall i \in E\backslash R$, $c_i(y_{it}) = c_i^0\left[1+\alpha\left(\frac{y_{it}}{u_i n_i}\right)^\beta\right]$; for $\forall i \in R$, $c_i(y_{it}) = c_i^0\left[1+\alpha\left(\frac{y_{it}}{u_i\left(n_i-\sum_{m=1}^{n_i} x_{imt}+\theta\sum_{m=1}^{n_i} v_{imt}\right)}\right)^\beta\right]$. $\alpha > 0, \beta > 0$.

With the notations above, the complete MS-MMN model is presented below:

MS-MMN:

Upper Level:

$$minimize\ z(\boldsymbol{s}) = \sum_{i\in E}\sum_{t=1}^{T} c_i(y_{it}) * y_{it} \tag{1}$$

$s.t.$

$$\sum_{t=1}^{t=T} s_{imt} = 1, \qquad \forall i \in R, \forall m \in [1, n_i] \tag{2}$$

$$\sum_{t=1}^{t=T} s_{imt} = 0, \qquad \forall i \notin R, \forall m \in [1, n_i] \tag{3}$$

$$x_{imt} = \sum_{a=max(t-p_i+1,1)}^{a=t} s_{ima}, \qquad \forall i \in R, \forall t \in T, \forall m \in [1, n_i] \tag{4}$$

$$\sum_{t=1}^{t=T} x_{imt} = p_i, \qquad \forall i \in R, \forall m \in [1, n_i] \tag{5}$$

$$\sum_{t=1}^{t=T} x_{imt} = 0, \qquad \forall i \notin R, \forall m \in [1, n_i] \tag{6}$$

$$v_{imt} = \sum_{a=1}^{a=t-p_i} s_{ima}, \qquad \forall i \in R, \forall m \in [1, n_i], \forall t \in [p_i+1, T] \tag{7}$$

$$v_{imt} = 0, \qquad \forall i \in R, \forall m \in [1, n_i], \forall t \in [1, p_i] \tag{8}$$

$$v_{imt} = 0, \qquad \forall i \notin R, \forall m \in [1, n_i], \forall t \in [1, T] \tag{9}$$

$$s_{imt}, x_{imt}, v_{imt} \in \{0, 1\}, \qquad \forall i \in E, \forall m \in [1, n_i], \forall t \in [1,\ T] \tag{10}$$

$$y_{it} = y_{it}^{UE} + y_{it}^{SO}, \qquad \forall i \in E, \forall t \in [1, T] \tag{12}$$

$$y_{it} \le \left(\sum_{k\in OD^{UE}} D_k^{UE} + \sum_{k\in OD^{SO}} D_k^{SO}\right)\left(n_i - \sum_{m=1}^{n_i} x_{imt}\right), \forall i \in R, \forall t \in [1,\ T] \tag{13}$$

Lower Level – UE Flow Assignment:

For $\forall t \in [1, T]$:

$$minimize\ \sum_{i \in E} \int^{y_{it}^{UE}} c_i\left(\omega, y_{it}^{SO}\right) d\omega \tag{14}$$
$s.t.$

$$y_{it}^{UE} = \sum_{k \in OD^{UE}} y_{ikt}^{UE}, \qquad \forall i \in E \tag{15}$$

$$D_k^{UE} = \sum_{\{i:E_i^- = OD_k^{UE-}, i \in E\}} y_{ikt}^{UE} - \sum_{\{j:E_j^+ = OD_k^{UE-}, j \in E\}} y_{jkt}^{UE}, \ \forall k \in OD^{UE} \tag{16}$$

$$D_k^{UE} = \sum_{\{i:E_i^+ = OD_k^{UE+}, i \in E\}} y_{ikt}^{UE} - \sum_{\{j:E_j^- = OD_k^{UE+}, j \in E\}} y_{jkt}^{UE}, \ \forall k \in OD^{UE} \tag{17}$$

$$\sum_{\{i:E_i^- = l, i \in E\}} y_{ikt}^{UE} = \sum_{\{j:E_j^+ = l, j \in E\}} y_{jkt}^{UE}, \quad \forall l \in N, \forall k \in \left\{k: OD_k^{UE-} \neq l\right\} \cap \left\{k: OD_k^{UE+} \neq l\right\} \tag{18}$$

$$y_{ikt}^{UE} \geq 0, \qquad \forall i \in E, \forall k \in OD^{UE} \tag{19}$$

Lower Level – SO Flow Assignment:

For $\forall t \in [1, T]$:

$$minimize\ \sum_{i \in E} c_i\left(y_{it}^{SO}, y_{it}^{UE}\right) * \left(y_{it}^{SO} + y_{it}^{UE}\right) \tag{20}$$
$s.t.$

$$y_{it}^{SO} = \sum_{k \in OD^{SO}} y_{ikt}^{SO}, \qquad \forall i \in E \tag{21}$$

$$D_k^{SO} = \sum_{\{i:E_i^- = OD_k^{SO-}, i \in E\}} y_{ikt}^{SO} - \sum_{\{j:E_j^+ = OD_k^{SO-}, j \in E\}} y_{jkt}^{SO}, \ \forall k \in OD^{SO} \tag{22}$$

$$D_k^{SO} = \sum_{\{i:E_i^+ = OD_k^{SO+}, i \in E\}} y_{ikt}^{SO} - \sum_{\{j:E_j^- = OD_k^{SO+}, j \in E\}} y_{jkt}^{SO}, \ \forall k \in OD^{SO} \tag{23}$$

$$\sum_{\{i:E_i^- = l, i \in E\}} y_{ikt}^{SO} = \sum_{\{j:E_j^+ = l, j \in E\}} y_{jkt}^{SO}, \forall l \in N, \forall k \in \left\{k: OD_k^{SO-} \neq l\right\} \cap \left\{k: OD_k^{SO+} \neq l\right\} \tag{24}$$

$$y_{ikt}^{SO} \geq 0, \qquad \forall i \in E, \forall k \in OD^{SO} \tag{25}$$

The MS-MMN model formulated is a challenging bi-level mixed-integer nonlinear program that has two parallel subproblems in the lower level. Currently there is no commercial solver available to handle this type of problem. Based on the bi-level structure of MS-MMN, the solution methods developed in the following two sections address the upper level scheduling problem and the lower level UE and SO assignment problems separately.

5.3 Solution Approach for the Lower Level Problem

Although the UE flow assignment problem and the SO flow assignment problem are two separate problems in the lower level of MS-MMN, they are connected by the link travel times. Given the schedule of lane closures on a certain day, the UE assignment will change if the SO assignment changes because link travel times have changed, and vice versa. Hence, one intuitive solution to the lower level of MS-MMN is the iterative UE-SO assignment algorithm developed in this section, which repetitively fixes the SO flows and solves the UE assignment problem, then fixes the UE flows obtained and solves the SO flow assignment, until the UE flows meet the UE condition and at the same time the SO flows minimizes the total travel time of all the flows. This section first proves the existence of the converged UE-SO flows, and then presents the iterative UE-SO assignment algorithm.

The converged UE and SO flow is the stationary status that both the UE flows and the SO flows are at their optimality for the UE assignment problem and the SO assignment problem respectively. That means the combined UE and SO flows result in the link travel times that satisfy both the UE condition for the UE flows and the SO condition for the SO flows. The existence of this stationary status is stated in the following lemma.

Lemma 5.3-1:

Given link available capacities and the origin-destination (OD) demand for user equilibrium (UE) flows and system optimum (SO) flows, there exists a routing pattern for all the OD demand that both UE flows and SO flows are at their optimality.

Proof of Lemma 5.3-1:

Besides the link-based formulation for the UE assignment problem shown in the previous section, there is an equivalent path-based formulation:

Lower Level – UE Flow Assignment (Path-based Formulation):

$$minimize\ \sum_{i\in E} \int^{y_{it}^{UE}} c_i(y_{it}^{SO}, \omega)d\omega \qquad (14)$$

$s.t.$

$$\sum_{p\in P_k} f_p^{k,t} = D_k^{UE} \qquad \forall k \in OD^{UE} \qquad (26)$$

$$y_{it}^{UE} = \sum_{k\in OD^{UE}} \sum_{p\in P_k} f_p^{k,t} \delta_{i,p}^k \qquad \forall i \in E \qquad (27)$$

$$f_p^{k,t} \geq 0 \qquad \forall k \in OD^{UE}, \forall p \in P_k \qquad (28)$$

$$y_{it}^{UE} \geq 0 \qquad \forall i \in E \qquad (19)$$

On any specific day t, variable $f_p^{k,t}$ is the amount of flows of OD pair k that travel on path p. $\delta_{i,p}^k$ is the parameter indicating whether link i is along path p for OD pair k. $\delta_{i,p}^k = 1$ if it is and $\delta_{i,p}^k = 0$ otherwise. P_k is the path set of the OD pair k. Constraint (26) makes sure all OD demands are satisfied. Constraint (27) ensures the flows from all OD pairs that generate UE flows are accounted for the total UE flow on the link.

Since another way to interpret the UE principle is that paths being used by flows have the same path travel time, and it equals to the minimum travel time between the OD pair, the UE condition can be ensured by a set of linear constraints with the introduction of binary variables instead of using Beckmann's function as the objective. For day t in the planning horizon, denote $c_p^{k,t}$ as the travel time on path p of OD pair k, and $c_{\min}^{k,t}$ as the minimum travel time of all the paths of OD pair k. Introduce binary variable $w_p^{k,t}$ for $\forall k \in OD, \forall p \in P_k$, which equals 1 if path p has longer travel time than the minimum travel time between OD pair k and 0 otherwise, the UE condition can be ensured by the following constraints:

$$c_p^{k,t} = \sum_{i \in E} \delta_{i,p}^k c_i(y_{it}^{SO}, y_{it}^{UE}) \qquad \forall k \in OD^{UE}, \forall p \in P_k \tag{29}$$

$$c_{min}^{k,t} \le c_p^{k,t} \qquad \forall k \in OD^{UE}, \forall p \in P_k \tag{30}$$

$$c_p^{k,t} - c_{min}^{k,t} \le M w_p^{k,t} \qquad \forall k \in OD^{UE}, \forall p \in P_k \tag{31}$$

$$f_p^{k,t} \le M\left(1 - w_p^{k,t}\right) \qquad \forall k \in OD^{UE}, \forall p \in P_k \tag{32}$$

$$c_p^{k,t} \ge 0 \qquad \forall k \in OD^{UE}, \forall p \in P_k \tag{33}$$

$$c_{min}^{k,t} \ge 0 \qquad \forall k \in OD^{UE}, \forall p \in P_k \tag{34}$$

$$w_p^{k,t} \in \{0, 1\} \qquad \forall k \in OD^{UE}, \forall p \in P_k \tag{35}$$

Constraint (29) calculates the path travel time and constraint (30) ensures $c_{\min}^{k,t}$ is the minimum travel time between OD pair k. Constraint (31) and (32) make sure paths will not be used by flows of OD pair k if its travel time is longer than the minimum travel time between the OD pair, and only paths with travel time equal to the minimum travel time can have flows on them. Hence the UE assignment problem is equivalent to finding a feasible solution to the set of constraints from (26) to (35). Therefore, the UE flow assignment problem and the SO flow assignment problem in the lower level can be combined as one optimization problem:

Lower Level: UE-SO Flow Assignment

For $\forall t \in [1, T]$:

$$minimize \ \sum_{i \in E} c_i\left(y_{it}^{SO}, y_{it}^{UE}\right) * \left(y_{it}^{SO} + y_{it}^{UE}\right) \tag{20}$$

$s.t.$

$$y_{it}^{SO} = \sum_{k \in OD^{SO}} y_{ikt}^{SO} \qquad \forall i \in E \tag{21}$$

$$D_k^{SO} = \sum_{\{i: E_i^- = OD_k^{SO-}, i \in E\}} y_{ikt}^{SO} - \sum_{\{j: E_j^+ = OD_k^{SO-}, j \in E\}} y_{jkt}^{SO} \ \forall k \in OD^{SO} \tag{22}$$

$$D_k^{SO} = \sum_{\{i:E_i^+=OD_k^{SO+},i\in E\}} y_{ikt}^{SO} - \sum_{\{j:E_j^-=OD_k^{SO+},j\in E\}} y_{jkt}^{SO}\ \forall k \in OD^{SO} \tag{23}$$

$$\sum_{\{i:E_i^-=l,i\in E\}} y_{ikt}^{SO} = \sum_{\{j:E_j^+=l,j\in E\}} y_{jkt}^{SO},\ \forall l \in N, \forall k \in \{k: OD_k^{SO-} \neq l\} \cap \{k: OD_k^{SO+} \neq l\} \tag{24}$$

$$\sum_{p\in P_k} f_p^{k,t} = D_k^{UE} \qquad \forall k \in OD^{UE} \tag{26}$$

$$y_{it}^{UE} = \sum_{k\in OD^{UE}} \sum_{p\in P_k} f_p^{k,t} \delta_{i,p}^k \qquad \forall i \in E \tag{27}$$

$$c_p^{k,t} = \sum_{\mathrm{i}\in E} \delta_{i,p}^k c_i(y_{it}^{SO}, y_{it}^{UE}) \qquad \forall k \in OD^{UE}, \forall p \in P_k \tag{29}$$

$$c_{min}^{k,t} \le c_p^{k,t} \qquad \forall k \in OD^{UE}, \forall p \in P_k \tag{30}$$

$$c_p^{k,t} - c_{min}^{k,t} \le M w_p^{k,t} \qquad \forall k \in OD^{UE}, \forall p \in P_k \tag{31}$$

$$f_p^{k,t} \le M\left(1 - w_p^{k,t}\right) \qquad \forall k \in OD^{UE}, \forall p \in P_k \tag{32}$$

$$f_p^{k,t} \ge 0 \qquad \forall k \in OD^{UE}, \forall p \in P_k \tag{28}$$

$$y_{ikt}^{SO} \ge 0 \qquad \forall i \in E, \forall k \in OD^{SO} \tag{25}$$

$$y_{it}^{UE} \ge 0 \qquad \forall i \in E \tag{19}$$

$$c_p^{k,t} \ge 0 \qquad \forall k \in OD^{UE}, \forall p \in P_k \tag{33}$$

$$c_{min}^{k,t} \ge 0 \qquad \forall k \in OD^{UE}, \forall p \in P_k \tag{34}$$

$$w_p^{k,t} \in \{0, 1\} \qquad \forall k \in OD^{UE}, \forall p \in P_k \tag{35}$$

The UE-SO flow assignment problem is feasible since it does not have contradicting constraints. Also, the feasible region is bounded and closed, because all variables are bounded by the OD demand either directly or indirectly, and the feasible space defined by each constraint contains its boundary. Hence, there exist optimal solutions to the UE-SO flow assignment problem. Because at the optimality the SO flows ($\boldsymbol{y^{SO}}$) minimize the total travel time of all flows and the UE flows ($\boldsymbol{y^{UE}}$) must satisfy the UE condition ensured by constraint (31) and (32), there exists a routing pattern for all the OD demand that UE flows satisfy the UE condition and SO flows are at their optimality. ☐

The iterative UE-SO assignment solves the UE assignment and the SO assignment alternately. The algorithm adopted for the UE assignment problem is the Traffic Assignment with Paired Alternative Segments (TAPAS) algorithm developed by Bar-Gera (2010). When the UE assignment is being solved, the SO flows are considered fixed. And thus, the link travel time function in the objective of the UE assignment becomes:

$$c_i(y_{it}^{UE}) = c_i^0\left[1 + \alpha\left(\frac{y_{it}^{UE} + \overline{y_{it}^{SO}}}{u_i\left(n_i - \sum_{m=1}^{n_i}\overline{x_{imt}} + \theta\sum_{m=1}^{n_i}\overline{v_{imt}}\right)}\right)^{\beta}\right]$$

where $\overline{y_{it}^{SO}}$ $(\forall i \in E)$ are the fixed SO flows and $\overline{x_{imt}}$ and $\overline{v_{imt}}$ have known values derived from a given lane closure schedule. Since the convergence of TAPAS algorithm is proved in Bar-Gera (2010), the UE flows will converge given fixed $\boldsymbol{y_t^{SO}}$.

The SO assignment problem with fixed $\boldsymbol{y_t^{UE}}$ is a convex optimization problem because its objective function is convex since its Hessian is a positive definite diagonal matrix, and the feasible region is a convex set since it is defined by linear constraints. Hence, the SO assignment given fixed $\boldsymbol{y_t^{UE}}$ can be solved by the Bi-conjugate Frank-Wolfe (BFW) algorithm adopted in Chapter 4 with a minor adjustment in the direction-finding subproblem and step size subproblem. The objective function for the SO assignment with fixed $\boldsymbol{y_t^{UE}}$ is:

$$minimize\ z(\boldsymbol{y^{SO}}) = \sum_{i\in E} c_i^0\left[1 + \alpha\left(\frac{y_{it}^{SO} + \overline{y_{it}^{UE}}}{u_i\left(n_i - \sum_{m=1}^{n_i}\overline{x_{imt}} + \theta\sum_{m=1}^{n_i}\overline{v_{imt}}\right)}\right)^{\beta}\right]\left(y_{it}^{SO} + \overline{y_{it}^{UE}}\right)$$

where $\overline{y_{it}^{UE}}$, $\overline{x_{imt}}$ and $\overline{v_{imt}}$ are all treated as parameters.

Suppose at n^{th} iteration, feasible flows $\overline{y_{it}^{SO}(n)}$ for $\forall i \in E$ are obtained, the gradient of the objective function is:

$$\nabla z_{it}\left(\overline{\boldsymbol{y}_n^{SO}}\right) = c_i^0 * \left(1 + \alpha(\beta+1)\left(\frac{\overline{y_{it}^{SO}(n)} + \overline{y_{it}^{UE}}}{u_i\left(n_i - \sum_{m=1}^{n_i} x_{imt} + \theta \sum_{m=1}^{n_i} v_{imt}\right)}\right)^{\beta}\right), \forall i \in E$$

where $\boldsymbol{y}_n^{SO}$ is the vector of $y_{it}^{SO}(n)\ \forall i \in E$. Let $\boldsymbol{w}_n$ denote the descending direction for the feasible solution $\overline{\boldsymbol{y}_n^{SO}}$, the **direction-finding subproblem** is:

$$minimize\ \boldsymbol{w}_n^T \nabla \boldsymbol{z_t}\left(\overline{\boldsymbol{y}_n^{SO}}\right) = \sum_{i\in E} c_i^0 * \left(1 + \alpha(\beta+1)\left(\frac{\overline{y_{it}^{SO}(n)} + \overline{y_{it}^{UE}}}{u_i\left(n_i - \sum_{m=1}^{n_i} x_{imt} + \theta \sum_{m=1}^{n_i} v_{imt}\right)}\right)^{\beta}\right) w_{it}(n) \quad (20')$$

$s.t.$:

$$w_{it} = \sum_{k\in OD^{SO}} w_{ikt} \qquad \forall i \in E \qquad (21')$$

$$D_k^{SO} = \sum_{\{i:E_i^- = OD_k^{SO-}, i\in E\}} w_{ikt} - \sum_{\{j:E_j^+ = OD_k^{SO-}, j\in E\}} w_{ikt} \quad \forall k \in OD^{SO} \qquad (22')$$

$$D_k^{SO} = \sum_{\{i:E_i^+ = OD_k^{SO+}, i\in E\}} w_{ikt} - \sum_{\{j:E_j^- = OD_k^{SO+}, j\in E\}} w_{ikt} \quad \forall k \in OD^{SO} \qquad (23')$$

$$\sum_{\{i:E_i^- = l, i\in E\}} w_{ikt} = \sum_{\{j:E_j^+ = l, j\in E\}} w_{ikt}, \forall l \in N, \forall k \in \left\{k: OD_k^{SO-} \neq l\right\} \cap \left\{k: OD_k^{SO+} \neq l\right\} \qquad (24')$$

$$w_{ikt} \geq 0 \qquad \forall i \in E, \forall k \in OD^{SO} \qquad (25')$$

This direction-finding subproblem can be perceived as a series of min-cost flow problems for the OD pairs with fixed unit flow cost $c_i^0 * \left(1 + \alpha(\beta+1)\left(\frac{\overline{y_{it}^{SO}(n)} + \overline{y_{it}^{UE}}}{u_i\left(n_i - \sum_{m=1}^{n_i} x_{imt} + \theta \sum_{m=1}^{n_i} v_{imt}\right)}\right)^{\beta}\right)\ \forall i \in E$. Since there is no hard link capacity constraint, $\boldsymbol{y}_n$ can be obtained by all-or-nothing assignment based on the "skewed" link cost $c_i^0 * \left(1 + \alpha(\beta+1)\left(\frac{\overline{y_{it}^{SO}(n)} + \overline{y_{it}^{UE}}}{u_i\left(n_i - \sum_{m=1}^{n_i} x_{imt} + \theta \sum_{m=1}^{n_i} v_{imt}\right)}\right)^{\beta}\right)$, which finds the shortest path for each OD pair and then send all the flows of the OD pair along that path. As a comparison, the true link travel time is $c_i^0 * \left(1 + \alpha\left(\frac{\overline{y_{it}^{SO}(n)} + \overline{y_{it}^{UE}}}{u_i\left(n_i - \sum_{m=1}^{n_i} x_{imt} + \theta \sum_{m=1}^{n_i} v_{imt}\right)}\right)^{\beta}\right)$,

Let $\overline{\boldsymbol{w}}_n$ be the descending direction obtained from the direction-finding subproblem, the **step size subproblem** is:

$$minimize\ z_t(\lambda)$$

$$= \sum_{i \in E} c_i^0 * \left(1 + \alpha \left(\frac{\left(\overline{y_{it}^{SO}(n)} + \overline{y_{it}^{UE}}\right) + \lambda\left(\overline{w_{it}(n)} - \overline{y_{it}^{SO}(n)}\right)}{u_i\left(n_i - \sum_{m=1}^{n_i} x_{imt}\ + \theta \sum_{m=1}^{n_i} v_{imt}\right)} \right)^{\beta} \right)$$

$$* \left[\left(\overline{y_{it}^{SO}(n)} + \overline{y_{it}^{UE}}\right) + \lambda\left(\overline{w_{it}(n)} - \overline{y_{it}^{SO}(n)}\right)\right],$$

$$s.t.: \lambda \in (0,\ 1)$$

which is,

$$minimize\ z_t(\lambda)$$

$$= \sum_{i \in E} c_i^0 * \left(1 + \alpha \left(\frac{\left(\overline{y_{it}^{SO}(n)} + \lambda\left(\overline{w_{it}(n)} - \overline{y_{it}^{SO}(n)}\right)\right) + \overline{y_{it}^{UE}}}{u_i\left(n_i - \sum_{m=1}^{n_i} x_{imt}\ + \theta \sum_{m=1}^{n_i} v_{imt}\right)} \right)^{\beta} \right)$$

$$* \left[\left(\overline{y_{it}^{SO}(n)} + \lambda\left(\overline{w_{it}(n)} - \overline{y_{it}^{SO}(n)}\right)\right) + \overline{y_{it}^{UE}}\right]$$

$$s.t.: \lambda \in (0,\ 1)$$

We do not need the flow feasibility constraints since both $\overline{\boldsymbol{y}_t^{SO}(n)}$ and $\overline{\boldsymbol{w}_t(n)}$ satisfy the flow feasibility constraints and $\overline{\boldsymbol{y}_t^{SO}(n)} + \lambda\left[\overline{\boldsymbol{w_t}(n)} - \overline{\boldsymbol{y}_t^{SO}(n)}\right]$ is their convex combination. The quadratic approximation algorithm is applied to solve the step size subproblem. For detailed execution procedure of the Frank-Wolfe algorithm and the discussion on its convergence, please refer to Section 4.4.2 in Chapter 4. Since it has been proven that the FW will converge (Frank and Wolfe, 1956), the SO flow assignment will converge with $\boldsymbol{y_t^{UE}}$ fixed.

Suppose a feasible solution is obtained for the UE-SO assignment problem on day t and it is $\overline{y_{it}^{SO}}$ and $\overline{y_{it}^{UE}}^{*}$ $(\forall i \in E)$. The star in the superscript means the flows are optimal for the lower level UE assignment problem. This feasible solution satisfies all the constraints in the integrated UE-SO assignment model including the UE condition constraints, but is sub-optimal since the SO flows are not optimized. If the UE flows are fixed at $\overline{y_{it}^{UE}}$ and the SO flows are optimized based on the fixed UE flows, a new solution to the UE-SO assignment problem can be obtained. Suppose it is $\overline{\overline{y_{it}^{SO}}}^{*}$ and $\overline{y_{it}^{UE}}$ $(\forall i \in E)$, where the star in the superscript means the flows are optimal for the lower level SO assignment problem, and the double bars indicate the SO flows are different from the previous $\overline{y_{it}^{SO}}$. However, the combination of $\overline{\overline{y_{it}^{SO}}}^{*}$ and $\overline{y_{it}^{UE}}$ $(\forall i \in E)$ is an infeasible solution to the integrated UE-SO assignment problem since $\overline{y_{it}^{UE}}$ $(\forall i \in E)$ no longer satisfy the UE condition constraints because the link travel times have changed. And thus $\overline{\overline{y_{it}^{SO}}}^{*}$ and $\overline{y_{it}^{UE}}$ $(\forall i \in E)$ is an infeasible solution to the UE-SO assignment problem. Hence, the iterative UE-SO assignment algorithm switches between the solutions obtained from UE assignment that are feasible and sub-optimal to the UE-SO assignment problem, and the solutions obtained from SO assignment which are infeasible, and eventually reaches the flow pattern that is optimal to the integrated UE-SO assignment problem.

Figure 5.3-i on the next page demonstrates the evolution of the mixed flow pattern over the iterative UE-SO flow assignment process. The horizontal axis represents the iterative UE-SO flow assignment iterations, the vertical axis is the total travel time of all flows. The horizontal dashed line is the total travel time associated with the optimal UE-SO flow assignment, where UE flows satisfy the UE conditions and SO flows are optimal at the same time. Compared with the total travel time of the optimal UE-SO flows, initially the total travel time of the mixed flows where UE flows meet the UE conditions but SO flows are sub-optimal is much higher, and the total travel time of the mixed flows where SO flows are optimal but UE flows don't satisfy the UE conditions is much lower. But as the iterative UE-SO assignment proceeds, the total travel time of the mixed flows is getting closer to that of the optimal UE-SO flows and eventually will be the same.

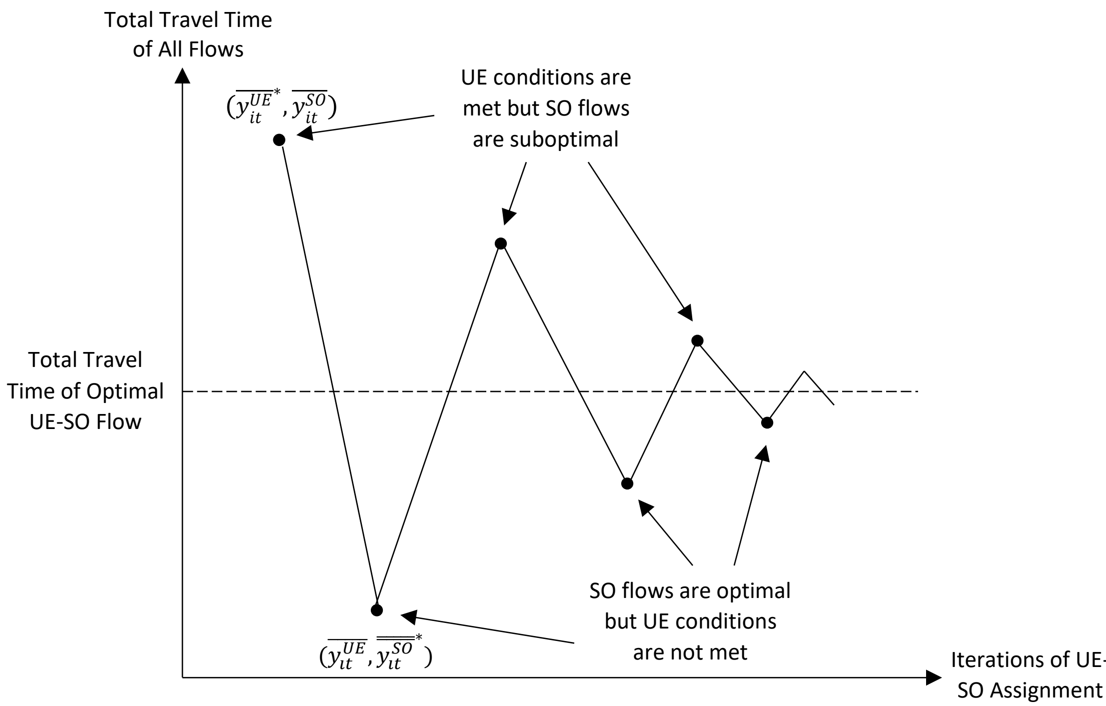


Figure 5.3-i: Total Travel Time Change in the Iterative UE-SO Assignment Process

This iterative UE-SO flow assignment procedure is illustrated in Figure 5.3-i:

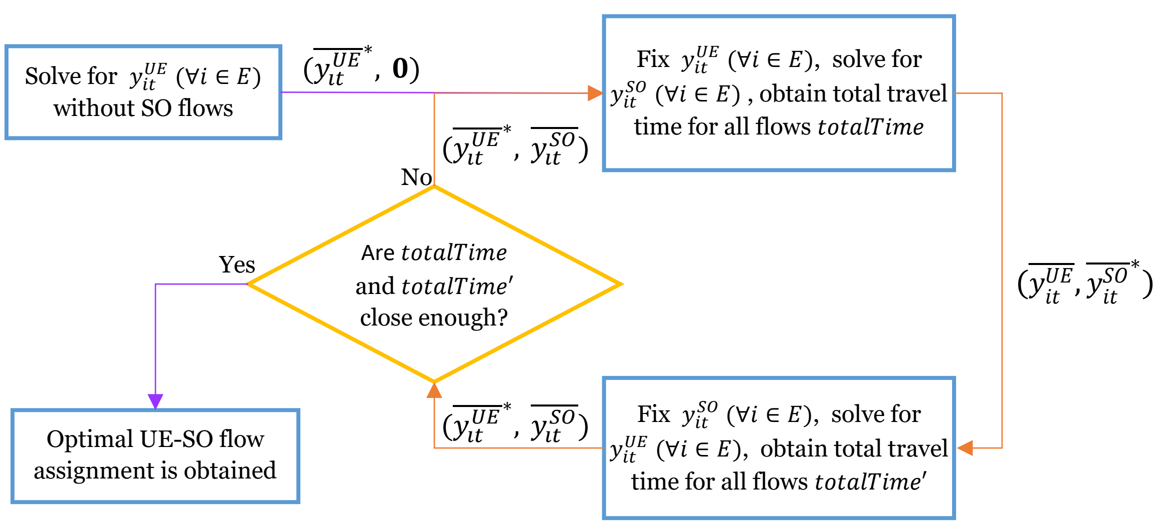


Figure 5.3-ii: Iterative UE-SO Assignment Algorithm

The computation procedure of the iterative UE-SO assignment algorithm is summarized as follows:

Iterative UE-SO Assignment Algorithm

Step 1: Solve the UE assignment problem without the SO flows.

Step 2: Fix the UE flows and solve the SO assignment problem. Record the travel time for all the flows $totalTime$.

Step 3: Fix the SO flows and solve the UE assignment problem. Record the travel time for all the flows $totalTime'$.

Step 4: Check whether $totalTime = totalTime'$. If it is, exit the algorithm; otherwise go back to Step 2.

The iterative UE-SO assignment algorithm, which contains the TAPAS algorithm for the UE assignment and the BFW algorithm for the SO assignment, is programmed in C++ and tested on two networks: the simple four-node network shown in Figure 4.3.1-i in Chapter 4 and the Sioux Falls Network shown in Figure3.4.3-iii in Chapter 3. The total OD demand in each network does not change but a certain percentage of the demand are SO flows and the rest of the demand are UE flows. The test cases are generated by varying the percentage of the demand that are SO flows. For example, if the SO flow percentage is 0%, all the demand are UE flows; and if the SO flow percentage is 100%, all the demand are SO flows.

Table 5.3-i gives the total travel time of converged UE-SO flows associated with different SO flow percentages in the simple four-node network. All five instances are solved within a second. It can be observed that as the SO flow percentage increases, the total travel time decreases. This is expected since the SO flow pattern is more efficient than the UE flow pattern.

Table 5.3-i: Iterative UE-SO Assignment in Four-Node Network

SO Flow Percentage	0	10%	50%	90%	100%
Total Travel Time	3066.63700	3066.63574	2990.34698	2901.53732	2901.53731

The total travel time and computation time for test cases generated based on the Sioux Falls network is summarized in Table 5.3-ii below. Again, it can be observed that the total travel time decreases as the percentage of SO flows increases.

Table 5.3-ii: Iterative UE-SO Assignment in Sioux Falls Network

SO Flow Percentage	0	10%	50%	90%	100%
Total Travel Time	7480226.09	7467535.71	7299283.73	7216487.21	7194258.54
Computation Time	1.671 sec	2.709 sec	63.97 sec	86.215 sec	11.667 sec

To obtain the total travel time resulted from a lane closure schedule, the UE-SO assignment needs to be solved for each day in the planning horizon based on the link available capacities. The travel time of the UE and SO flows on each day then will be summed up over the planning horizon to obtain the total travel time associated with the schedule.

5.4 Solution Approach for the Upper Level Problem

With the iterative UE-SO assignment algorithm to evaluate lane closure schedules in the lower level, this section develops the solution method for the upper level to obtain the schedules. But before that, the convexity of the objective function and the feasible region of MS-MMN is explored. The following lemma shows the convexity of the objective function of MS-MMN.

Lemma 5.4-1:

The objective function of the MS-MMN problem is convex if the link travel time function is the BPR function.

Proof of Lemma 5.4-1:

For a certain day t in the planning horizon, take the first derivative of the objective function with respect to the UE flows and SO flows on link i, we obtain:

$$\nabla c_i\left(y_{it}^{SO}, y_{it}^{UE}\right) * \left(y_{it}^{SO} + y_{it}^{UE}\right)$$

$$= \left[\frac{\partial c_i\left(y_{it}^{SO}, y_{it}^{UE}\right)}{\partial y_{it}^{SO}} * \left(y_{it}^{SO} + y_{it}^{UE}\right) + c_i\left(y_{it}^{SO}, y_{it}^{UE}\right) \quad \frac{\partial c_i\left(y_{it}^{SO}, y_{it}^{UE}\right)}{\partial y_{it}^{UE}} * \left(y_{it}^{SO} + y_{it}^{UE}\right) + c_i\left(y_{it}^{SO}, y_{it}^{UE}\right)\right]$$

Then take the second derivative of the objective function with respect to the UE and SO flows on link i, we have:

$$\mathrm{H}\left[c_i\left(y_{it}^{SO}, y_{it}^{UE}\right) * \left(y_{it}^{SO} + y_{it}^{UE}\right)\right] =$$

$$\begin{bmatrix} \frac{\partial^2 c_i\left(y_{it}^{SO}, y_{it}^{UE}\right)}{\partial \left(y_{it}^{SO}\right)^2} * \left(y_{it}^{SO} + y_{it}^{UE}\right) + 2\frac{\partial c_i\left(y_{it}^{SO}, y_{it}^{UE}\right)}{\partial y_{it}^{SO}} & \frac{\partial^2 c_i\left(y_{it}^{SO}, y_{it}^{UE}\right)}{\partial y_{it}^{UE} \partial y_{it}^{SO}} * \left(y_{it}^{SO} + y_{it}^{UE}\right) + \frac{\partial c_i\left(y_{it}^{SO}, y_{it}^{UE}\right)}{\partial y_{it}^{UE}} + \frac{\partial c_i\left(y_{it}^{SO}, y_{it}^{UE}\right)}{\partial y_{it}^{SO}} \\ \frac{\partial^2 c_i\left(y_{it}^{SO}, y_{it}^{UE}\right)}{\partial y_{it}^{SO} \partial y_{it}^{UE}} * \left(y_{it}^{SO} + y_{it}^{UE}\right) + \frac{\partial c_i\left(y_{it}^{SO}, y_{it}^{UE}\right)}{\partial y_{it}^{SO}} + \frac{\partial c_i\left(y_{it}^{SO}, y_{it}^{UE}\right)}{\partial y_{it}^{UE}} & \frac{\partial^2 c_i\left(y_{it}^{SO}, y_{it}^{UE}\right)}{\partial \left(y_{it}^{UE}\right)^2} * \left(y_{it}^{SO} + y_{it}^{UE}\right) + 2\frac{\partial c_i\left(y_{it}^{SO}, y_{it}^{UE}\right)}{\partial y_{it}^{UE}} \end{bmatrix}$$

Since BPR function is adopted as the link travel time function, the second derivative (i.e, the Hessian matrix) is simplified to:

$$\mathrm{H}[c_i(y_{it}^{SO}, y_{it}^{UE}) * (y_{it}^{SO} + y_{it}^{UE})]$$

$$= \begin{bmatrix} \alpha(\beta+1)\beta\left(\frac{y_{it}^{SO} + y_{it}^{UE}}{u_i\left(n_i - \sum_{m=1}^{n_i} x_{imt} + \theta \sum_{m=1}^{n_i} v_{imt}\right)}\right)^{\beta-1} & \alpha(\beta+1)\beta\left(\frac{y_{it}^{SO} + y_{it}^{UE}}{u_i\left(n_i - \sum_{m=1}^{n_i} x_{imt} + \theta \sum_{m=1}^{n_i} v_{imt}\right)}\right)^{\beta-1} \\ \alpha(\beta+1)\beta\left(\frac{y_{it}^{SO} + y_{it}^{UE}}{u_i\left(n_i - \sum_{m=1}^{n_i} x_{imt} + \theta \sum_{m=1}^{n_i} v_{imt}\right)}\right)^{\beta-1} & \alpha(\beta+1)\beta\left(\frac{y_{it}^{SO} + y_{it}^{UE}}{u_i\left(n_i - \sum_{m=1}^{n_i} x_{imt} + \theta \sum_{m=1}^{n_i} v_{imt}\right)}\right)^{\beta-1} \end{bmatrix}$$

After a few elementary row operations, it becomes:

$$\mathrm{H}\left[c_i\left(y_{it}^{SO}, y_{it}^{UE}\right) * \left(y_{it}^{SO} + y_{it}^{UE}\right)\right] = \begin{bmatrix} \alpha(\beta+1)\beta\left(\frac{y_{it}^{SO} + y_{it}^{UE}}{u_i\left(n_i - \sum_{m=1}^{n_i} x_{imt} + \theta \sum_{m=1}^{n_i} v_{imt}\right)}\right)^{\beta-1} & 0 \\ 0 & 0 \end{bmatrix}$$

Combining the Hessian of all the link flow variables for the objective function, it is concluded that the Hessian matrix of the objective function is positive semidefinite because it is a diagonal matrix with the elements along the diagonal either have positive values or are zeros. Hence, the objective function is convex.

To find out whether the feasible region of MS-MMN is convex or not, the feasible region of the UE-SO assignment problem, which is the lower level of MS-MMN, is investigated first. ☐

Lemma 5.4-2:

The linear relaxation of the UE-SO assignment model has a non-convex feasible region.

Proof of Lemma 5.4-2:

Since only $w_p^{k,t}$ for $\forall k \in OD^{UE}, \forall p \in P_k$ are not continuous variables, these variables are relaxed from being binary to taking values in $[0,1]$. After the relaxation, all constraints in UE-SO assignment model are linear constraints with continuous variables except Constraint (29), which is a nonlinear equality constraint with all feasible points on the surface. Suppose we have two sets of feasible UE-SO flows $\overline{y_{it}^{UE}}$ and $\overline{y_{it}^{SO}}$, and $\overline{\overline{y_{it}^{UE}}}$ and $\overline{\overline{y_{it}^{SO}}}$ for $\forall i \in E$, from Constraint (29) we have:

$$\overline{c_p^{k,t}} = \sum_{i \in E} \delta_{i,p}^k c_i(\overline{y_{it}^{SO}}, \overline{y_{it}^{UE}})$$

and

$$\overline{\overline{c_p^{k,t}}} = \sum_{i \in E} \delta_{i,p}^k c_i(\overline{\overline{y_{it}^{SO}}}, \overline{\overline{y_{it}^{UE}}})$$

Since $c_i(y_{it}^{SO}, y_{it}^{UE})$ is the nonlinear BPR function with $\beta > 1$, it is obvious that for $\lambda \in [0,1]$

$$\lambda\overline{c_p^{k,t}} + (1-\lambda)\overline{\overline{c_p^{k,t}}} \neq \sum_{i \in E} \delta_{i,p}^k c_i(\lambda\overline{y_{it}^{SO}} + (1-\lambda)\overline{\overline{y_{it}^{SO}}}, \lambda\overline{y_{it}^{UE}} + (1-\lambda)\overline{\overline{y_{it}^{UE}}})$$

Therefore, the feasible region defined by Constrain (29) is not convex and thus the feasible region of the linear relaxation of UE-SO assignment model is not convex.

Lemma 5.3-2 below and shows the linear relaxation of the MS-MMS problem has a non-convex feasible region:

Lemma 5.4-3:

The linear relaxation of the MS-MMN model has a non-convex feasible region.

Proof of Lemma 5.4-3:

With the UE-SO assignment model developed in Section 5.2, the MS-MMN model can also be formulated as a single-level optimization problem by duplicating the UE-SO assignment model for each day in the planning horizon and with the addition of the scheduling variables and constraints, because both models have the same objective of minimizing the total travel time of UE flows and SO flows. The single-level MS-MMN model is shown below:

Single-Level MS-MMN

$$minimize\ z(\boldsymbol{s}) = \sum_{i \in E} \sum_{t=1}^{T} c_i(y_{it}) * y_{it} \qquad (1)$$

$$s.t.$$

$$\sum_{t=1}^{t=T} s_{imt} = 1 \qquad \forall i \in R, \forall m \in [1, n_i] \qquad (2)$$

$$\sum_{t=1}^{t=T} s_{imt} = 0 \qquad \forall i \notin R, \forall m \in [1, n_i] \qquad (3)$$

$$x_{imt} = \sum_{a=max(t-p_i+1,1)}^{a=t} s_{ima} \qquad \forall i \in R, \forall m \in [1, n_i], \forall t \in T \qquad (4)$$

$$\sum_{t=1}^{t=T} x_{imt} = p_i \qquad \forall i \in R, \forall m \in [1, n_i] \qquad (5)$$

$$\sum_{t=1}^{t=T} x_{imt} = 0 \qquad \forall i \notin R, \forall m \in [1, n_i] \qquad (6)$$

$$v_{imt} = \sum_{a=1}^{a=t-p_i} s_{ima} \qquad \forall i \in R, \forall m \in [1, n_i], \forall t \in [p_i + 1, T] \qquad (7)$$

$$v_{imt} = 0 \qquad \forall i \in R, \forall m \in [1, n_i], \forall t \in [1, p_i] \qquad (8)$$

$$v_{imt} = 0 \qquad \forall i \notin R, \forall m \in [1, n_i], \forall t \in [1, T] \qquad (9)$$

$$y_{it} = y_{it}^{UE} + y_{it}^{SO}, \qquad \forall i \in E, \forall t \in [1, T] \qquad (12)$$

$$y_{it} \leq \left(\sum_{k \in OD^{UE}} D_k^{UE} + \sum_{k \in OD^{SO}} D_k^{SO}\right)\left(n_i - \sum_{m=1}^{n_i} x_{imt}\right) \forall i \in R, \forall t \in [1,\ T] \qquad (13)$$

$$y_{it}^{SO} = \sum_{k \in OD^{SO}} y_{ikt}^{SO}, \qquad \forall i \in E, \forall t \in [1,\ T] \qquad (21)$$

$$D_k^{SO} = \sum_{\{i: E_i^- = OD_k^{SO-}, i \in E\}} y_{ikt}^{SO} - \sum_{\{j: E_j^+ = OD_k^{SO-}, j \in E\}} y_{jkt}^{SO} \ \forall k \in OD^{SO}, \forall t \in [1,\ T] \qquad (22)$$

$$D_k^{SO} = \sum_{\{i: E_i^+ = OD_k^{SO+}, i \in E\}} y_{ikt}^{SO} - \sum_{\{j: E_j^- = OD_k^{SO+}, j \in E\}} y_{jkt}^{SO} \ \forall k \in OD^{SO}, \forall t \in [1,\ T] \qquad (23)$$

$$\sum_{\{i: E_i^- = l, i \in E\}} y_{ikt}^{SO} = \sum_{\{j: E_j^+ = l, j \in E\}} y_{jkt}^{SO} \qquad \forall l \in N,$$

$$\forall k \in \{k: OD_k^{SO-} \neq l\} \cap \{k: OD_k^{SO+} \neq l\}, \forall t \in [1, T] \qquad (24)$$

$$\sum_{p \in P_k} f_p^{k,t} = D_k^{UE} \qquad \forall k \in OD^{UE}, \forall t \in [1,\ T] \qquad (26)$$

$$y_{it}^{UE} = \sum_{k \in OD^{UE}} \sum_{p \in P_k} f_p^{k,t} \delta_{i,p}^k \qquad \forall i \in E, \forall t \in [1,\ T] \qquad (27)$$

$$c_p^{k,t} = \sum_{\mathrm{i} \in E} \delta_{i,p}^k c_i(y_{it}^{SO}, y_{it}^{UE}) \qquad \forall k \in OD^{UE}, \forall p \in P_k, \forall t \in [1,\ T] \qquad (29)$$

$$c_{min}^{k,t} \leq c_p^{k,t} \qquad \forall k \in OD^{UE}, \forall p \in P_k, \forall t \in [1,\ T] \qquad (30)$$

$$c_p^{k,t} - c_{min}^{k,t} \leq M w_p^{k,t} \qquad \forall k \in OD^{UE}, \forall p \in P_k, \forall t \in [1,\ T] \qquad (31)$$

$$f_p^{k,t} \leq M\left(1 - w_p^{k,t}\right) \qquad \forall k \in OD^{UE}, \forall p \in P_k, \forall t \in [1,\ T] \qquad (32)$$

$$f_p^{k,t} \geq 0 \qquad \forall k \in OD^{UE}, \forall p \in P_k, \forall t \in [1,\ T] \qquad (28)$$

$$y_{ikt}^{SO} \geq 0 \qquad \forall i \in E, \forall k \in OD^{SO}, \forall t \in [1,\ T] \qquad (25)$$

$$y_{it}^{UE} \geq 0 \qquad \forall i \in E, \forall t \in [1,\ T] \qquad (19)$$

$$c_p^{k,t} \geq 0 \qquad \forall k \in OD^{UE}, \forall p \in P_k, \forall t \in [1,\ T] \qquad (33)$$

$$c_{min}^{k,t} \geq 0 \qquad \forall k \in OD^{UE}, \forall p \in P_k, \forall t \in [1,\ T] \qquad (34)$$

$$s_{imt}, x_{imt}, v_{imt} \in \{0, 1\} \qquad \forall i \in E, \forall m \in [1, n_i], \forall t \in [1,\ T] \qquad (10)$$

$$w_p^{k,t} \in \{0, 1\} \qquad \forall k \in OD^{UE}, \forall p \in P_k, \forall t \in [1,\ T] \qquad (35)$$

After relaxing all the binary variables, all the constraints are linear constraints with continuous variables except Constraint (29) which is a nonlinear equality constraint. Follow the same logic in the proof of Lemma 5.4-2, it is concluded that the feasible region of the linear relaxation for the single-level MS-MMN model is non-convex. □

Because of the non-convexity of the linear relaxation for MS-MMN, it is not easy to find the global optimal solution for MS-MMN, nor to prove a solution obtained is global optimal. Hence, the well-established genetic algorithm (GA) is applied to solve the MS-MMN. The genetic algorithm was first introduced by Holland in 1975. It is a metaheuristic that solves complex optimization problems through bio-inspired operators, such as selection, crossover and mutation. Because implementing GA is relatively easy and requires little knowledge about the problem structure, GA has been applied to solve difficult optimization problems in a broad range of disciplines. Since the MS-MMN is a challenging bi-level mixed-integer nonlinear program with its linear relaxation being non-convex, GA is considered a suitable solution method for the MS-MMN. Here are the key components of the GA for MS-MMN:

Decimal Encoding for GA

The genes of a member in a generation are the repair start dates of each lane in the links that need repair, instead of the binary variables $\boldsymbol{s}$ that indicate whether the repair of a lane starts on a certain day. Thus, the GA for MS-MMN has decimal encoding. Given the repair start dates, the values of variables $\boldsymbol{x}$ and $\boldsymbol{v}$ can be determined, and so are the link available capacities on each day in the planning horizon.

Initial Population for GA

The genes of members in the first generation are generated randomly. For each lane, the repair start date is a random number generated between day 1 and its latest possible repair start date. The latest possible repair start date for a lane is the date that if the repair starts on that day, this lane will be repaired on due date T. For an example, if each lane of link i requires $p_i = 5$ days to repair and the maintenance due date for all the maintenance work is $T = 18$, then the latest possible repair start date for all the lanes in link i is day 14 since otherwise the repair will not be completed on time if the it starts on days later than day 14. Hence, the latest repair start date for the lanes in link i is calculated as $T - p_i + 1$. To ensure the population in each generation is large enough have all possible repair start dates of a lane be present in the same generation, the population size (N) is determined as:

$$N = T - \min_{i \in R}\{p_i\} + 1 \qquad (5.4\text{-a})$$

since the lane that requires the least number of days to repair has the most choices of repair start dates.

Selection Rules for GA

The fitness of a member is evaluated based on the total travel time over the planning horizon associated with the member's gene, which essentially is a schedule of lane closures. The less the total travel time is, the fitter the member is. After the computation of the total travel time associated with each member in a generation, these members are ranked in ascending order with respected to their total travel times. Suppose there are N members in a generation, $rank_N$ is the member whose gene results in the largest total travel time in current generation and $rank_1$ is the member whose gene results in the least total travel time. The fitness of a member with the j^{th} rank is calculated as:

$$Fitness_{rank_j} = totalTravelTime_{rank_N} - totalTravelTime_{rank_j} + 1 \qquad (5.4\text{-b})$$

which is one plus the difference between the largest total travel time in current generation and the total travel time of the member with rank j. The reason to add one in the fitness calculation is to ensure the member with the largest total travel time can also be selected for crossover with a positive probability. The probability of the member with rank j being selected for crossover is:

$$selecProb_{rank_j} = \frac{Fitness_{rank_j}}{\sum_{a=1}^{a=N} Fitness_{rank_a}} \qquad (5.4\text{-c})$$

In the computation procedure, a random number r will be generated between (0, 1]. If $r < selecProb_{rank_1}$, then the member with the least total travel time will be selected for crossover; if $\sum_{a=1}^{a=j-1} selecProb_{rank_a} < r \leq \sum_{a=1}^{a=j} selecProb_{rank_a}$ $\forall j \in [1, N]$, then the member with rank j is selected for crossover.

Handling Entire Link Closures and Infeasible Schedules

Since BPR function is used as the link travel time function and it has link available capacity in the denominator, the available capacity cannot be zero. Thus, if a link is entirely closed on a certain day, the available link capacity is set to 10^{-6} instead of 0 and the free-flow travel time of the link is set to 10^{30}, so that all the paths that contain this link have travel times that are much longer than other paths. As a result of this manipulation, no flow will use these paths and effectively this link is entirely closed.

If one or more links are entirely closed on a certain day, it is possible that some OD pairs may not be able to find a path connecting the origin and destination to send the flows, rendering the maintenance schedule infeasible. But in our computational procedure the schedule is still "feasible" since all those entirely closed links still have the available capacity of 10^{-6}. Therefore, the UE-SO assignment problem can still be solved but the total travel time will be drastically larger than those of the feasible schedules. Since the members with less total travel times are fitter and have a better chance of being selected for crossover, the members whose genes result in drastically large total travel times (i.e., infeasible schedules) will be eliminated in the computational procedure.

Crossover in GA

The GA for MS-MMN applies the multi-point crossover scheme, and the number of crossover points $nbCPoints$ is determined as:

$$nbCPoints = \frac{totalWZ}{\max_{\{i \in R\}}\{n_i\}} \qquad (5.4\text{-d})$$

that is, the total number of lanes to repair ($totalWZ$) divided by the largest number of lanes in a link among the links that need repair. Since too few crossover points will limit the flexibility of the crossover operation on finding better combinations of genes, and too many crossover points will result in offspring not very different from the parents and unnecessarily increasing the computations, it is desirable to have more link-level schedule swaps between the two members selected for crossover because UE-SO flows route through the network based on the link travel times. With the number of crossover points determined by 5.4-d, a total of $nbCPoints$ random numbers are generated between $[1, totalWZ]$ to determine the exact loci to start the gene swap for the members selected for crossover. Preliminary experiments indicate that this method can have more link-level schedule swaps on average.

To demonstrate the crossover procedure, suppose in a network the links that need repair have a total of 16 lanes. Among these links, link 5 has 4 lanes which is the most number of lanes. The number of crossover points in this case is $nbCPoints = 16/4 = 4$. Suppose the four crossover points randomly generated between [1, 16] are 2, 6, 9, 13, Figure 5.4-i on the next page illustrates the crossover operation for this case:

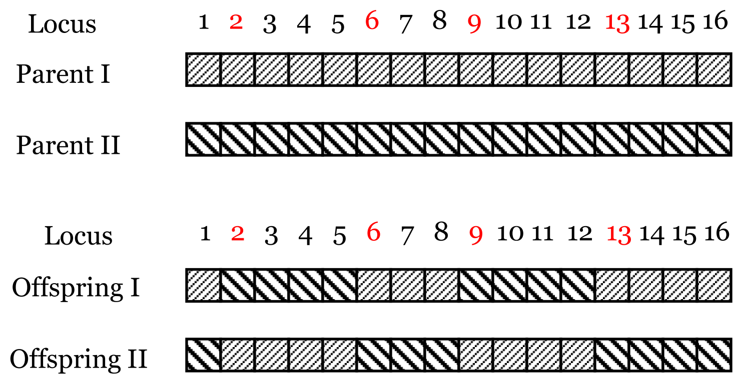


Figure 5.4-i: Four-Point Crossover Example

Mutation in GA

The mutation rate is designed to decrease gradually from a pre-specified upper bound ($MuUB$) towards the lower bound ($MuLB$) from one generation to the next. Suppose the maximum number of generations to be computed is NG, the mutation rate of generation ng is calculated as:

$$Mu_{ng} = MuUB - \frac{MuUB - MuLB}{NG} * ng \qquad (5.4\text{-e})$$

The changing mutation rate helps GA explore the solution space for better schedules in the early stage and accelerate the convergence in the later stage.

To determine the loci for mutation, a total of $\lceil Mu_{ng} * totalWZ \rceil$ random numbers are generated between $[1, totalWZ]$. Each of these random numbers represent the locus where the mutation happens. For each of these loci, the repair start date of the lane will be an integer number randomly generated between the first day of the planning horizon and the latest possible repair start date for the lane. All the offspring generated from the crossover operation will go through this mutation process before becoming members in the next generation. To retain the best schedule obtained so far, the member with the best fitness in current generation will be directly put into the next generation without mutation.

Stopping Criteria for GA

The GA for MS-MMN will stop if the pre-specified maximum number of generations have been computed, or the best schedule hasn't changed for the past 10 consecutive generations.

The combination of the GA and the iterative UE-SO flow assignment algorithm completes the solution approach for MS-MMN. The overall computation procedure to solve MS-MMN is described below:

Step 1: Initial population is randomly generated

Step 2: Evaluate the members in current generation

Step 2.1: For a member, on each day in planning horizon, calculate the link available capacities, and perform the iterative UE-SO assignment algorithm to obtain the UE-SO flow travel time

Step 2.2: Sum the travel time over the planning horizon to obtain the total travel time associated with the member

Step 3: If the number of generations computed reach the pre-specified limit, or the best member hasn't changed for the last 10 consecutive generations, exit the solution procedure. Otherwise continue to Step 4

Step 4: Calculate the probability for each member to be selected for crossover

Step 5: Repetitively select two members to perform multi-point crossover, until the number of offspring is $N-1$

Step 6: Perform mutation on the $N-1$ offspring produced

Step 7: Add the member with the best fitness in the parent generation to the offspring generation, and go back to Step 2

5.5 Computational Experiments

The solution approach developed for MS-MMN is programmed in C++ and tested with three problem instances based on the square network shown in Figure 4.3.1-ii in Chapter 4. In the first problem scenario, 10% of the links are randomly selected to be the links that need repair. And the percentage of links to repair are 20% and 30% respectively in the other two problem scenarios. All three scenarios have the same OD demand and the same SO flow percentage of 10%, which means 10% of the demand for each OD pair will route through the network to achieve system optimum, and the rest 90% of the demand will route through the network to reach user equilibrium. All the maintenance works are due in 18 days for all the three scenarios. Since the square network is a specially designed network that can have severe Braess Paradox effect, for each scenario, five test cases are created to make sure the aggregated test results align with commonsense, that is, in general the more links need to be repaired during the same period of time, the higher the total travel time would be because of the network capacity is reduced. The detailed information of these test cases can be found in Appendix C.

Setting the upper bound of mutation rate 20% and the lower bound 10% for the GA, and using a personal computer with 3.7 GHz CPU and 24 GB memory for the computation work, the results of the three repair scenarios are summarized in Table 5.5-i, Table 5.5-ii and Table 5.5-iii respectively. As it can be observed from these three tables, the average computation time gets longer as more links need to be repaired. Also, as more links with lanes closed for maintenance during the same period of time, the total travel time of all flows gets longer since the available capacity of links are less, which leads to longer link travel times and longer travel times in general.

Table 5.5-i: Results of Five Test Cases for Square Network with 10% of Links to Repair

	Case I	Case II	Case III	Case IV	Case V	Average
Total Travel Time	198287018	228115451	198006312	199017203	200042655	204693727.8
Computation Time (in hours)	1.74	5.85	2.36	2.24	3.28	3.09
Number of Generations Computed	27	46	26	24	18	28

Table 5.5-ii: Results of Five Test Cases for Square Network with 20% of Links to Repair

	Case I	Case II	Case III	Case IV	Case V	Average
Total Travel Time	199755215	253964706	207756792	201070121	200019555	212513277.8
Computation Time (in hours)	6.13	6.11	4.76	1.42	4.12	4.56
Number of Generations Computed	24	24	26	21	34	26

Table 5.5-iii: Results of Five Test Cases for Square Network with 30% of Links to Repair

	Case I	Case II	Case III	Case IV	Case V	Average
Total Travel Time	219824173	207171576	211923696	235200590	199572821	214738571.2
Computation Time (in hours)	5.53	7.11	10.29	12.15	4.08	7.83
Number of Generations Computed	16	14	47	29	32	28

Because of the randomness of GA, to show the performance of the solution method developed, a test case is selected from each scenario and is solved five times and the computation results are averaged over the five runs. The results of the test cases selected are summarized in Table 5.5-iv, 5.5-v, 5.5-vi respectively. It is obvious that as more links are required to be repaired during the same period of time, GA takes longer to solve the problem instance.

Table 5.5-iv: Five Runs of Test Case I in 10% of Links to Repair Scenario

	Run 1	Run 2	Run 3	Run 4	Run 5	Average
Total Travel Time	198287958	198318282	198287018	198277517	198263262	198286807
Computation Time (in hours)	1.79	0.70	1.74	1.74	2.35	1.66
Number of Generations Computed	26	12	27	28	40	26.6

Table 5.5-v: Five Runs of Test Case I in 20% of Links to Repair Scenario

	Run 1	Run 2	Run 3	Run 4	Run 5	Average
Total Travel Time	199585929	199885561	199755215	199719781	199591233	199707543.8
Computation Time (in hours)	6.93	5.32	6.13	8.59	3.8	6.15
Number of Generations Computed	31	22	24	37	15	26

Table 5.5-vi: Five Runs of Test Case I in 10% of Links to Repair Scenario

	Run 1	Run 2	Run 3	Run 4	Run 5	Average
Total Travel Time	219824173	220572965	219586053	218746178	219424194	219630712
Computation Time (in hours)	5.53	10.82	18.78	10.04	9.4	10.91
Number of Generations Computed	16	26	49	24	24	28

The solution approach is also tested with two problem instances generated based on the Sioux Falls network shown in Figure 3.4.3-iii in Chapter 3. The percentage of the links that need repair in these two problem instances are 10% and 20% respectively. All maintenance works are due in 21 days and the SO flow percentage is 10% for both problem instances. The detailed information of these two test cases can be found in the Appendix C, and the total demand of UE and SO flows for each OD pair is the same as the Sioux Falls network test case, which can be found online. With the same mutation rate settings and the same computer for the computation work, the results are summarized in Table 5.5-vii and Table 5.5-viii.

Table 5.5-vii: Sioux Falls Network with 10% of Links to Repair

	Run 1	Run 2	Run 3	Run 4	Run 5	Average
Total Travel Time	173595710	174214938	173513915	174169416	174244662	173947728.2
Computation Time (in hours)	26.42	12.51	24.94	12.94	17.80	18.92
Number of Generations Computed	40	19	43	19	27	30

Table 5.5-viii: Sioux Falls Network with 20% of Links to Repair

	Run 1	Run 2	Run 3	Run 4	Run 5	Average
Total Travel Time	232500828	226601969	229547278	226966956	224486683	228020742
Computation Time (in hours)	31.93	33.56	48.43	43.03	35.24	38.44
Number of Generations Computed	21	20	30	25	23	24

The test cases generated based on the Sioux Falls network take much longer to solve than those generated based on the square network. And the reason is because Sioux Falls network is larger and requires longer computation time for the UE-SO flow assignment to obtain the converged UE-SO flow. Also, the longer planning horizon means the UE-SO flow assignment needs to be performed for more days for a schedule. And the larger problem size generally requires larger population, which means more schedules must be evaluated in a generation. From the five problem instances tested, it can be perceived that in general the MS-MMN takes a long time to solve. This is because the iterative UE-SO assignment algorithm needs to be performed repetitively for each day in the planning horizon and for all the schedules generated in GA.

5.6 Conclusion

With the fast-evolving technologies of self-driving cars, people will start traveling with these new transportation modes in the near future. Thus, the traffic flows in the road network would become more multi-modal flow, where travelers driving human-operated cars choosing the routes that minimize individual travel times, and travelers with self-driving cars selecting routes that minimize the total travel time of all the travelers. This multi-modal traffic flow essentially is a mixture of UE flows and SO flows.

This chapter investigates the maintenance scheduling problem in multi-modal networks (MS-MMN), where a set of links need to be repaired before a common due date, each lane of these links is an independent work zone to be scheduled, and there are mixed UE-SO flows routing through the network every day based on the link available capacities. A bi-level mixed-integer nonlinear program is formulated for this problem with the upper level to find schedules, and the lower level to obtain the converged UE-SO flows for the schedules obtained in the upper level.

The existence of the converged UE-SO flow is proved, and this converged flow can be obtained by the iterative UE-SO assignment algorithm developed in this chapter. Given link available capacities and OD demand, the iterative UE-SO assignment algorithm iteratively fixes the UE flows and solves the SO assignment problem, and fixes the SO flows and solves the UE assignment problem. This iterative procedure stops when the UE flows are optimal to the UE assignment problem and at the same time the SO flows are optimal to the SO assignment problem.

Since the MS-MMN is a challenging non-convex optimization problem, GA is applied to find good schedules that will result in less total travel time over the planning horizon. However, in general the MS-MMN takes a long time solve since the UE-SO flow assignment need to be performed for each day in the planning horizon and for each schedule in the generation. One possible way to reduce the computation time is to use parallel computing techniques for GA. Since most computers nowadays are equipped with a multi-core CPU and each core has two threads that can work on different tasks independently, by assigning each member in a generation to one of the available threads, the computation of total travel times associated with the members can be done parallelly. Then, these threads will perform the UE-SO assignment for each day in the planning horizon for the member assigned, and return the associated total travel time. Once all members in the generation have been evaluated, the crossover and mutation can also be done parallelly in the same fashion.

A direction for future research is to further differentiate the autonomous vehicle flows and the connected vehicle flows in MS-MMN. Since travelers using connected vehicles still are the decision makers on route choices, the connected vehicle flow most likely will not be the exact SO flow, but a flow pattern that is somewhere between the UE flow pattern and the SO flow pattern. Thus, future research topics include (a) how to model the connected vehicle flow, (b) whether there exists a converged multi-modal flow of these three travel modes (i.e., human-operated cars, self-driving cars, and connected vehicles), (c) how to obtain the converged multi-modal flow if it exists, and (d) how this multi-modal flow will react to the work zone schedules.

Chapter 6

CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

In transportation networks, both non-recurring events (e.g., road maintenance) and recurring events (i.e., demand surges during rush hours) can cause traffic congestion. To alleviate the traffic congestion caused by these two types of events, this dissertation develops solution from the supply side with a network-wide perspective. It builds optimization models to manage mandatory network capacity change to minimize the congestion caused by road maintenance activities, and designs the mechanism to manage optional network capacity change to reduce the congestion caused by inefficient routing in normal time.

The research on maintenance planning for various types of physical networks has been mostly focused on the long-term planning and the short-term planning. The long-term maintenance planning addresses the research question of how to maintain the network for a certain level of reliability or service quality with minimum maintenance cost. And the short-term planning schedules maintenance activities on a link to minimize the flow disruptions locally. Although maintenance work changes the network layout temporally and will impact the routing of OD flows, the long-term maintenance planning omits this effect because the planning horizon is much longer than the period when the network is under maintenance. And the short-term maintenance planning does not consider the flow diverted from the link being repaired to the neighborhood links since the scope of the problem is limited to the link being repaired. However, more often than not maintenance work needs to be performed on a set of links that are close to each other in a relatively short period of time (medium term). In these situations, the scheduling and coordination of these maintenance works are critical to the network capability on serving the flows. And this is particularly true for transportation network since each unit of flows (i.e., vehicles) can change its route on its own in response to changed network layouts.

The medium-term maintenance planning hasn't drawn much attention from researchers until last decade. Among the handful research that has investigated the medium-term maintenance planning with the consideration of network-wide OD flow diversions, most research did not consider partial link closures or assumed links under maintenance would have 50% of capacity decrease. Chapter 3 and Chapter 5 fill this blank and investigate the lane-based maintenance scheduling problem, where there are a set of links to repair before a common due date, and each lane of these links is an independent work zone to be scheduled.

Considering the exacerbation of traffic mobility and safety caused by the combination of work zones and service vehicles (e.g., trucks), Chapter 3 develops a mathematical model to optimize maintenance schedules particularly for service vehicle flows. These service vehicles are assumed to route through the network based on available link capacities every day to achieve system optimum (SO). The link travel cost function is designed to be piece-wise linear to approximate the nonlinear relation between the travel cost and the number of trucks traveling on the link. Because of the introduction of piece-wise linear link travel cost function, the problem of maintenance scheduling in networks of service vehicles (MS-NSV) is formulated as a mixed-integer linear program (MIP). Although there are commercial solvers available for MIPs, they are not able to solve MS-NSV instances within a tolerable amount of time because the solution space explodes as the problem size gets larger. Fortunately, this issue can been handled well by the randomized fix-and-optimize (RFO) heuristic developed. With a feasible schedule, RFO will randomly decompose the links that need repair into groups and optimize the work zone schedules for one group with schedules of other work zones fixed. RFO is an effective mechanism to limit the number of integer variables to be solved at a time. Computational experiments on various test cases show that RFO is able to obtain good quality solutions within much less time than solving the problem instances solely by CPLEX.

Chapter 5 extends the work in Chapter 3 to study the maintenance scheduling in networks with multi-modal traffic flows (MS-MMN). The travel modes considered in Chapter 5 include private cars and autonomous vehicles. Every traveler that drives a private car will take the route that minimizes his/her own travel time to reach user equilibrium (UE), and the travelers riding autonomous vehicles will choose the routes that minimize the total travel time of all the travelers to achieve system optimum (SO). Since flows of different travel modes share the road network, they compete for the limited capacity on the links. MS-MMN is formulated as a bi-level mixed-integer nonlinear program. The upper level of MS-MMN searches for the schedule that minimizes the total travel time of all travelers over the planning horizon, and the lower level finds the mixed UE-SO flow assignment for each day in the planning horizon based on a feasible schedule.

The lower level of MS-MMN contains two optimization problems: the UE assignment problem for travelers using private cars and the SO assignment problem for travelers riding autonomous vehicles. The optimal solution for the lower level is the UE-SO flow assignment where UE flows satisfy the UE condition and SO flows minimize the total travel time of all flows at the same time. Given the link available capacities and OD demand for UE flows and SO flows on a certain day, the existence of the optimal solution for the lower level UE-SO assignment problem is proved. The iterative UE-SO assignment algorithm is developed solve the lower level problem. It iteratively fixes the UE flows and solves the SO assignment problem, and fixes the SO flows and solves the UE assignment problem, until the total travel time between two iterations are the same. With the Bureau of Public Road (BPR) function adopted as the link travel time function, the non-convexity of MS-MMN is shown and the upper level scheduling problem is solved by the genetic algorithm with multi-point crossover. Since for each schedule evaluation the iterative UE-SO assignment has to be performed for each day in the planning horizon, it takes a long time to solve MS-MMN instances in moderate-size.

As to the strategy for managing optional network capacity changes, Chapter 4 develops a mechanism that selectively reduce the capacity of some links to improve the overall efficiency of the UE flow pattern. The research work in Chapter 4 is inspired by the well-known Braess paradox, which describes the counter-intuitive phenomenon in networks with UE flows, that adding more links to the network could worsen the traffic congestion, and congestion could be alleviated by removing links from the network. Chapter 4 studies the generalized Braess paradox that reducing the capacity of some links could improve the efficiency of UE flows. Compared to the generalized Braess paradox, the original Braess paradox is a special case since removing a link is the same as reducing the link capacity to zero.

Chapter 4 develops a heuristic that identifies the links whose capacity reduction could decrease the total travel time at UE, and finds the desired amount of link capacity reduction. Assuming link travel time is a decreasing function of link capacity, the basic idea of the heuristic is to reduce the capacity of some links to increase the link travel time, so as to drive the UE flow pattern towards the more efficient SO flow pattern. To find the links to reduce capacity, the UE assignment problem and the SO assignment problem are solved for the same OD demand, and links are sorted with respect to the difference between the total UE flows and the total SO flows on the link. Links with more UE flows than SO flows are considered being over-used by the UE flows, and could be candidates for capacity reductions. If the total travel time at UE is less after the link capacity reduction, the capacity reduction on the link can increase and the UE assignment problem will be solved again; otherwise the link capacity reduction could be decreased. This process repeats for the link until the total travel time at UE cannot decrease further through greater link capacity reductions. The process effectively is a line search to set the best capacity reductions for the link. Once the best capacity reduction is found for a link, the UE assignment and SO assignment will be solved and links are sorted again based on the difference between the total UE flows and the total SO flows on every link. And then a new round of link capacity reduction trials is considered. If there is no effective capacity reduction for current selected link, the heuristic considers the next link in the list to do a line search for a capacity reduction.

The heuristic is implemented in both C++ and AMPL. In the C++ implementation, the UE assignment problem is solved by the Traffic Assignment with Paired Alternative Segments (TAPAS) algorithm developed by Bar-Gera (2010), and the SO assignment problem is solved by a Bi-conjugate Frank-Wolfe (BFW) type algorithm. For the AMPL implementation, both UE and SO assignment problems are solved by the nonlinear commercial solver MINOS. Experiments on real network test cases show that MINOS sometimes fails to give correct solutions to the UE and SO assignment problems because some test cases are too large for MINOS to handle. Experiments on real networks demonstrate the generalized Braess' paradox exists in reality, and the C++ implementation with TAPAS and BFW is more reliable than the AMPL implementation with MINOS.

In summary, this dissertation develops optimization methods to manage both mandatory and optional network capacity changes. The computational experiments on real network test cases indicate the solution methods developed are efficient and reliable.

6.2 Future Work

Since the problems studied in Chapter 3, Chapter 4 and Chapter 5 do not involve any uncertainties, investigating these problems in stochastic settings would be a major extension to this dissertation. Uncertainties can stem from all aspects of the problems studied. For example, instead of assuming travelers have perfect information about the path travel times, it is more realistic to model travelers' perception of the path travel times as the true path travel time plus a random perception error. With travelers' perception error modeled, the UE assignment problem in the lower levels of OCREC and MS-MMN evolve to the stochastic UE assignment problem, which has been well researched in the literature as reviewed in Section 2.1.5 in Chapter 2. Correspondingly, the SO assignment problem in MS-NSV and MS-MMN becomes the stochastic SO assignment problem, and can be solved by the methods developed in literature for the stochastic UE assignment with some alteration.

Another way to involve uncertainty is to consider stochastic OD demand. The OD demands are assumed to be known in this dissertation but actually they are random variables, whose distributions can be estimated from historical data. With stochastic OD demand modeled, the three problems studied can be formulated as typical two-stage stochastic programs (Shapiro et al., 2009), where the first stage is to decide the schedule of lane closures in MS-NSV and MS-MMN and the link capacity reductions in OCREC, and the second stage solves the flow assignment problems. Since MS-NSV is a mixed-integer linear program, it can be solved by a progressive hedging method (Watson and Woodruff, 2011), which is a solution approach based on scenario decomposition of the stochastic parameters. Although the progressive hedging method has been used to handle nonlinear stochastic programs, the non-convexity of OCREC and MS-MMN would require extra caution when progressive hedging is applied as a meta-heuristic to solve OCREC and MS-MMN.

The stochastic programs investigated in literature only involve uncertainties in the follower problem, and all attributes of the decisions in the leader problem are deterministic. For example, in the MS-NSV and MS-MMN with stochastic OD demands, the uncertainty is considered in the lower level flow assignment problems but there is no uncertainty involved in the upper level scheduling problem, that is, it is assumed that the maintenance work on a lane of link i will last exactly p_i days. However, it is common for a road maintenance project to finish either earlier or later than the planned completion date due to various reasons (e.g., unexpected good/severe weather condition, work zone accidents, addition/failure of machines, etc.). Hence, the number of days required to repair a lane is a random variable and its distribution can be estimated from historical data. The MS-NSV and MS-MMN that involve uncertainty in project durations introduce a new category of stochastic program, where some attributes of the decisions in the leader problem are random variables. How to address this new type of stochastic program would be another interesting and challenging future research problem.

BIBLIOGRAPHY

(FHWA), F. H. A. (2015). Urban Congestion Report April 2015 - June 2015. Retrieved from https://ops.fhwa.dot.gov/perf_measurement/ucr/reports/fy2015_q3.htm

(FHWA), F. H. A. (2017). Recurring Traffic Bottlenecks: A Primer. Retrieved from https://ops.fhwa.dot.gov/publications/fhwahop12012/sec1.htm#sec13

Abdel-Aty, M., Dhindsa, A., & Gayah, V. (2007). Considering various ALINEA ramp metering strategies for crash risk mitigation on freeways under congested regime. Transportation Research Part C-Emerging Technologies, 15(2), 113-134. doi:10.1016/j.trc.2007.02.003

Agrawal, S., Zheng, H., Peeta, S., & Kumar, A. (2016). Routing aspects of electric vehicle drivers and their effects on network performance. Transportation Research Part D-Transport and Environment, 46, 246-266. doi:10.1016/j.trd.2016.04.002

Ahmed, F., & Hawas, Y. E. (2015). An integrated real-time traffic signal system for transit signal priority, incident detection and congestion management. Transportation Research Part C-Emerging Technologies, 60, 52-76. doi:10.1016/j.trc.2015.08.004

Ahuja, R. K., Magnanti, T. L., & Orlin, J. B. (1993). Network Flows Theory, Algorithms, and Applications: Prentice-Hall, Inc.

Akamatsu, T. (1996). Cyclic flows, Markov process and stochastic traffic assignment. Transportation Research Part B-Methodological, 30(5), 369-386. doi:10.1016/0191-2615(96)00003-3

Askoura, Y., Lebacque, J. P., & Haj-Salem, H. (2011). Optimal sub-networks in traffic assignment problem and the Braess paradox. Computers & Industrial Engineering, 61(2), 382-390. doi:10.1016/j.cie.2010.11.011

Babonneau, F., & Vial, J. P. (2009). ACCPM with a nonlinear constraint and an active set strategy to solve nonlinear multicommodity flow problems. Mathematical Programming, 120(1), 179-210. doi:10.1007/s10107-007-0151-3

Bagloee, S. A., & Asadi, M. (2015). Prioritizing road extension projects with interdependent benefits under time constraint. Transportation Research Part a-Policy and Practice, 75, 196-216. doi:10.1016/j.tra.2015.03,016

Bagloee, S. A., Ceder, A., Tavana, M., & Bozic, C. (2014). A heuristic methodology to tackle the Braess Paradox detecting problem tailored for real road networks. Transportmetrica a-Transport Science, 10(5), 437-456. doi:10.1080/23249935.2013.787557

Baiocchi, A., Cuomo, F., De Felice, M., & Fusco, G. (2015). Vehicular Ad-Hoc Networks sampling protocols for traffic monitoring and incident detection in Intelligent Transportation Systems. Transportation Research Part C-Emerging Technologies, 56, 177-194. doi:10.1016/j.trc.2015.03.041

Balijepalli, N. C., Ngoduy, D., & Watling, D. P. (2014). The two-regime transmission model for network loading in dynamic traffic assignment problems. Transportmetrica a-Transport Science, 10(7), 563-584. doi:10.1080/18128602.2012.751680

Bao, Y., Gao, Z., Xu, M., Sun, H., & Yang, H. (2015). Travel mental budgeting under road toll: An investigation based on user equilibrium. Transportation Research Part a-Policy and Practice, 73, 1-17. doi:10.1016/j.tra.2014.12.010

Bar-Gera, H. (2010). Traffic assignment by paired alternative segments. Transportation Research Part B-Methodological, 44(8-9), 1022-1046. doi:10.1016/j.trb.2009.11.004

Bayraktar, M. E., & Hastak, M. (2009). A decision support system for selecting the optimal contracting strategy in highway work zone projects. Automation in Construction, 18(6), 834-843. doi:10.1016/j.autcon.2009.03.007

Bell, M. G. H., & Lida, Y. (1997). Transportation Network Analysis. Baffins Lane, Chichester, West Sussex PO19 1UD, England: John Wiley & Sons Ltd.

Bellei, G., Gentile, G., & Papola, N. (2005). A within-day dynamic traffic assignment model for urban road networks. Transportation Research Part B-Methodological, 39(1), 1-29. doi:10.1016/j.trb.2004.02.002

Bhouri, N., Haj-Salem, H., & Kauppila, J. (2013). Isolated versus coordinated ramp metering: Field evaluation results of travel time reliability and traffic impact.

Transportation Research Part C-Emerging Technologies, 28, 155-167. doi:10.1016/j.trc.2011.11.001

Bliemer, M. C. J., & Bovy, P. H. L. (2003). Quasi-variational inequality formulation of the multiclass dynamic traffic assignment problem. Transportation Research Part B-Methodological, 37(6), 501-519. doi:10.1016/s0191-2615(02)00025-5

Bliemer, M. C. J., & Trb. (2000). Quasi-variational inequality approach to multi-user-class dynamic traffic assignment. In Traffic Flow Theory and Highway Capacity 2000: Highway Operations, Capacity, and Traffic Control (pp. 11-19). Washington: Transportation Research Board Natl Research Council.

Bocchini, P., & Frangopol, D. M. (2011). A probabilistic computational framework for bridge network optimal maintenance scheduling. Reliability Engineering & System Safety, 96(2), 332-349. doi:10.1016/j.ress.2010.09.001

Bocchini, P., & Frangopol, D. M. (2013). Connectivity-Based Optimal Scheduling for Maintenance of Bridge Networks. Journal of Engineering Mechanics-Asce, 139(6), 760-769. doi:10.1061/(asce)em.1943-7889.0000271

Boland, N., Kalinowski, T., & Kaur, S. (2015). Scheduling network maintenance jobs with release dates and deadlines to maximize total flow over time: Bounds and solution strategies. Computers & Operations Research, 64, 113-129. doi:10.1016/j.cor.2015.05.011

Boland, N., Kalinowski, T., Waterer, H., & Zheng, L. (2014). Scheduling arc maintenance jobs in a network to maximize total flow over time. Discrete Applied Mathematics, 163, 34-52. doi:10.1016/j.dam.2012.05.027

Boyce, D. E., Janson, B. N., & Eash, R. W. (1981). THE EFFECT ON EQUILIBRIUM TRIP ASSIGNMENT OF DIFFERENT LINK CONGESTION FUNCTIONS. Transportation Research Part a-Policy and Practice, 15(3), 223-232. doi:10.1016/0191-2607(81)90003-0

Braess, D., Nagurney, A., & Wakolbinger, T. (2005). On a paradox of traffic planning. Transportation Science, 39(4), 446-450. doi:10.1287/trsc.1050.0127

Branston, D. (1976). LINK CAPACITY FUNCTIONS - REVIEW. Transportation Research, 10(4), 223-236. doi:10.1016/0041-1647(76)90055-1

Carey, M., & Subrahmanian, E. (2000). An approach to modelling time-varying flows on congested networks. Transportation Research Part B-Methodological, 34(3), 157-183. doi:10.1016/s0191-2615(99)00019-3

Carlson, R. C., Papamichail, I., & Papageorgiou, M. (2014). Integrated feedback ramp metering and mainstream traffic flow control on motorways using variable speed limits. Transportation Research Part C-Emerging Technologies, 46, 209-221. doi:10.1016/j.trc.2014.05.017

Carlson, R. C., Papamichail, I., Papageorgiou, M., & Messmer, A. (2010). Optimal Motorway Traffic Flow Control Involving Variable Speed Limits and Ramp Metering. Transportation Science, 44(2), 238-253. doi:10.1287/trsc.1090.0314

Castillo, E., Menendez, J. M., Jimenez, P., & Rivas, A. (2008). Closed form expressions for choice probabilities in the Weibull case. Transportation Research Part B-Methodological, 42(4), 373-380. doi:10.1016/j.trb.2007.08.002

Chai, G., Guo, J., Wu, L., & Miao, X. (2015). Improved bottleneck approach for heuristic systemwide responsive ramp metering on congested freeways. Ksce Journal of Civil Engineering, 19(1), 296-302. doi:10.1007/s12205-014-0613-0

Chanut, S., Buisson, C., & Trb, T. R. B. (2003). Macroscopic model and its numerical solution for two-flow mixed traffic with different speeds and lengths. Traffic Flow Theory and Highway Capacity 2003: Highway Operations, Capacity, and Traffic Control(1852), 209-219.

Chen, A., Pravinvongvuth, S., Xu, X., Ryu, S., & Chootinan, P. (2012). Examining the scaling effect and overlapping problem in logit-based stochastic user equilibrium models. Transportation Research Part a-Policy and Practice, 46(8), 1343-1358. doi:10.1016/j.tra.2012.04.003

Chen, H., Liu, Y., & Nie, Y. (2015). Solving the step-tolled bottleneck model with general user heterogeneity. Transportation Research Part B-Methodological, 81, 210-229. doi:10.1016/j.trb.2015.09.005

Chen, X. J., Diao, Z., & Hu, X. D. (2016). Network Characterizations for Excluding Braess's Paradox. Theory of Computing Systems, 59(4), 747-780. doi:10.1007/s00224-016-9710-4

Cheng, Y., Zhang, M., & Yang, D. (2015). Automatic Incident Detection for Urban Expressways Based on Segment Traffic Flow Density. Journal of Intelligent Transportation Systems, 19(2), 205-213. doi:10.1080/15472450.2014.977046

Chi, R., Hou, Z., Jin, S., Wang, D., & Hao, J. (2013). A Data-Driven Iterative Feedback Tuning Approach of ALINEA for Freeway Traffic Ramp Metering With PARAMICS Simulations. Ieee Transactions on Industrial Informatics, 9(4), 2310-2317. doi:10.1109/tii.2013.2238548

Chien, S. I. J., & Tang, Y. (2014). Scheduling highway work zones with genetic algorithm considering the impact of traffic diversion. Journal of Advanced Transportation, 48(4), 287-303. doi:10.1002/atr.213

Chow, A. H. F., & Li, Y. (2014). Robust Optimization of Dynamic Motorway Traffic via Ramp Metering. Ieee Transactions on Intelligent Transportation Systems, 15(3), 1374-1380. doi:10.1109/tits.2014.2310454

Chu, J. C., & Chen, Y. J. (2012). Optimal threshold-based network-level transportation infrastructure life-cycle management with heterogeneous maintenance actions. Transportation Research Part B-Methodological, 46(9), 1123-1143. doi:10.1016/j.trb.2012.05.002

Chung, Y. (2011). Quantification of Nonrecurrent Congestion Delay Caused by Freeway Accidents and Analysis of Causal Factors. Transportation Research Record(2229), 8-18. doi:10.3141/2229-02

Clark, S. D., & Watling, D. P. (2002). Sensitivity analysis of the probit-based stochastic user equilibrium assignment model. Transportation Research Part B-Methodological, 36(7), 617-635. doi:10.1016/s0191-2615(01)00021-2

Connors, R. D., Sumalee, A., & Watling, D. P. (2007). Sensitivity analysis of the variable demand probit stochastic user equilibrium with multiple user-classes. Transportation Research Part B-Methodological, 41(6), 593-615. doi:10.1016/j.trb.2006.11.003

Corthout, R., Tampere, C. M. J., & Immers, L. H. (2009). Marginal Incident Computation An Efficient Algorithm to Determine Congestion Spillback due to Incidents. Transportation Research Record(2099), 22-29. doi:10.3141/2099-03

Dafermos, S., & Sparrow, F. T. (1971). OPTIMAL RESOURCE ALLOCATION AND TOLL PATTERNS IN USER-OPTIMISED TRANSPORT NETWORKS. Journal of Transport Economics and Policy, 5(2), 184-200.

Daganzo, C. F. (1994). THE CELL TRANSMISSION MODEL - A DYNAMIC REPRESENTATION OF HIGHWAY TRAFFIC CONSISTENT WITH THE HYDRODYNAMIC THEORY. Transportation Research Part B-Methodological, 28(4), 269-287. doi:10.1016/0191-2615(94)90002-7

Daganzo, C. F. (1995). THE CELL TRANSMISSION MODEL .2. NETWORK TRAFFIC. Transportation Research Part B-Methodological, 29(2), 79-93. doi:10.1016/0191-2615(94)00022-r

Davis, L. C. (2007). Effect of adaptive cruise control systems on mixed traffic flow near an on-ramp. Physica a-Statistical Mechanics and Its Applications, 379(1), 274-290. doi:10.1016/j.physa.2006.12.017

Di, S., Pan, C., & Ran, B. (2008). Stochastic Multiclass Traffic Assignment with Consideration of Risk-Taking Behaviors. Transportation Research Record(2085), 111-123. doi:10.3141/2085-13

Di, X., He, X. Z., Guo, X. L., & Liu, H. X. (2014). Braess paradox under the boundedly rational user equilibria. Transportation Research Part B-Methodological, 67, 86-108. doi:10.1016/j.trb.2014.04.005

Dominguez Frejo, J. R., & Fernandez Camacho, E. (2012). Global Versus Local MPC Algorithms in Freeway Traffic Control With Ramp Metering and Variable Speed Limits. Ieee Transactions on Intelligent Transportation Systems, 13(4), 1556-1565. doi:10.1109/tits.2012.2195493

Elefteriadou, L., Kondyli, A., Brilon, W., Hall, F. L., Persaud, B., & Washburn, S. (2014). Enhancing Ramp Metering Algorithms with the Use of Probability of Breakdown Models. Journal of Transportation Engineering, 140(4). doi:10.1061/(asce)te.1943-5436.0000653

Even, S., Itai, A., Shamir, A. (Oct. 1975). On the complexity of time table and multi-commodity flow problems. Paper presented at the 16th Annual Symposium on Foundations of Computer Science, Washington, DC, USA.

Ewing, R., & Proffitt, D. (2016). Improving Decision Making for Transportation Capacity Expansion: Qualitative Analysis of Best Practices for Regional Transportation Plans. Transportation Research Record(2568), 1-8. doi:10.3141/2568-01

Fan, W., & Gurmu, Z. (2014). Combined Decision Making of Congestion Pricing and Capacity Expansion: Genetic Algorithm Approach. Journal of Transportation Engineering, 140(8). doi:10.1061/(asce)te.1943-5436.0000695

Faulkner, L., Dekker, F., Gyles, D., Papamichail, I., & Papageorgiou, M. (2014). Evaluation of HERO-Coordinated Ramp Metering Installation at M1 and M3 Freeways in Queensland, Australia. Transportation Research Record(2470), 13-23. doi:10.3141/2470-02

Ferrari, P. (2009). The effect of the competition between cars and trucks on the evolution of the motorway transport system. Transportation Research Part C-Emerging Technologies, 17(6), 558-570. doi:10.1016/j.trc.2009.05.002

Ferrari, P. (2011). The dynamics of the competition between cars and trucks on motorways. Transportation Research Part C-Emerging Technologies, 19(4), 579-592. doi:10.1016/j.trc.2010.09.012

Fontaine, P., & Minner, S. (2014). Benders Decomposition for Discrete-Continuous Linear Bilevel Problems with application to traffic network design. Transportation Research Part B-Methodological, 70, 163-172. doi:10.1016/j.trb.2014.09.007

Frangopol, D. M., & Liu, M. (2007). Bridge network maintenance optimization using stochastic dynamic programming. Journal of Structural Engineering, 133(12), 1772-1782. doi:10.1061/(asce)0733-9445(2007)133:12(1772)

Frank, M., & Wolfe, P. (1956). An Algorithm for Quadratic Programming. Naval Research Logistics, 3(1-2), 95-110.

FWHA. (September, 2013). Facts and Statistics - Work Zone Delay. Facts and Statistics. Retrieved from http://www.ops.fhwa.dot.gov/wz/resources/facts_stats/delay.htm#fn1

Gan, M., Chen, S., & Yan, Y. (2013). The effect of roadway capacity expansion on facility sitting. Applied Mathematics & Information Sciences, 7(2), 575-581.

Gang, W. M., & Yong, L. S. (2011). Highway Ramp Metering Technique for Solving Non-Recurrent Congestion according to Incident. [돌발상황에 따른 비 반복정체를 해소하기 위한 고속도로 램프미터링 기법]. Journal of Korean Institute of Intelligent Systems, 21, 186-191.

Gao, L., Xie, C., Zhang, Z. M., & Waller, S. T. (2011). Integrated Maintenance and Expansion Planning for Transportation Network Infrastructure. Transportation Research Record(2225), 56-64. doi:10.3141/2225-07

Gawron, C. (1998). An iterative algorithm to determine the dynamic user equilibrium in a traffic simulation model. International Journal of Modern Physics C, 9(3), 393-407. doi:10.1142/s0129183198000303

Geroliminis, N., Srivastava, A., & Michalopoulos, P. (2011). A Dynamic-Zone-Based Coordinated Ramp-Metering Algorithm With Queue Constraints for Minnesota's Freeways. Ieee Transactions on Intelligent Transportation Systems, 12(4), 1576-1586. doi:10.1109/tits.2011.2164792

Ghali, M. O., & Smith, M. J. (1995). A MODEL FOR THE DYNAMIC SYSTEM OPTIMUM TRAFFIC ASSIGNMENT PROBLEM. Transportation Research Part B-Methodological, 29(3), 155-170. doi:10.1016/0191-2615(94)00024-t

Goel, A., & Meisel, F. (2013). Workforce routing and scheduling for electricity network maintenance with downtime minimization. European Journal of Operational Research, 231(1), 210-228. doi:10.1016/j.ejor.2013.05.021

Goffin, J. L., Gondzio, J., Sarkissian, R., & Vial, J. P. (1997). Solving nonlinear multicommodity flow problems by the analytic center cutting plane method. Mathematical Programming, 76(1), 131-154. doi:10.1007/bf02614381

Golani, H., Waller, S. T., & Trb. (2004). Combinatorial approach for multiple-destination user optimal dynamic traffic assignment. Transportation Network Modeling 2004(1882), 70-78.

Gomes, G., & Horowitz, R. (2006). Optimal freeway ramp metering using the asymmetric cell transmission model. Transportation Research Part C-Emerging Technologies, 14(4), 244-262. doi:10.1016/j.trc.2006.08.001

Gomes, G., Horowitz, R., Kurzhanskiy, A. A., Varaiya, P., & Kwon, J. (2008). Behavior of the cell transmission model and effectiveness of ramp metering. Transportation Research Part C-Emerging Technologies, 16(4), 485-513. doi:10.1016/j.trc.2007.10.005

Gomes, N., Pinheiro, R., Vale, Z., & Ramos, C. (2007). Scheduling maintenance activities of electric power transmission networks using an hybrid constraint method. Engineering Intelligent Systems for Electrical Engineering and Communications, 15(3), 127-134.

Guo, R.-Y., Yang, H., Huang, H.-J., & Tan, Z. (2015). Link-based day-to-day network traffic dynamics and equilibria. Transportation Research Part B-Methodological, 71, 248-260. doi:10.1016/j.trb.2014.11.005

Guo, X. (2013). Toll sequence operation to realize target flow pattern under bounded rationality. Transportation Research Part B-Methodological, 56, 203-216. doi:10.1016/j.trb.2013.08.006

Hajibabai, L., Bai, Y., & Ouyang, Y. (2014). Joint optimization of freight facility location and pavement infrastructure rehabilitation under network traffic equilibrium. Transportation Research Part B-Methodological, 63, 38-52. doi:10.1016/j.trb.2014.02.003

Han, D., & Yang, H. (2008). The multi-class, multi-criterion traffic equilibrium and the efficiency of congestion pricing. Transportation Research Part E-Logistics and Transportation Review, 44(5), 753-773. doi:10.1016/j.tre.2007.07.011

Han, L., Ukkusuri, S., & Doan, K. (2011). Complementarity formulations for the cell transmission model based dynamic user equilibrium with departure time choice, elastic demand and user heterogeneity. Transportation Research Part B-Methodological, 45(10), 1749-1767. doi:10.1016/j.trb.2011.07.007

Han, S. J. (2003). Dynamic traffic modelling and dynamic stochastic user equilibrium assignment for general road networks. Transportation Research Part B-Methodological, 37(3), 225-249. doi:10.1016/s0191-2615(02)00009-7

Helber, S., & Sahling, F. (2010). A fix-and-optimize approach for the multi-level capacitated lot sizing problem. International Journal of Production Economics, 123(2), 247-256. doi:10.1016/j.ijpe.2009.08.022

Helbing, D., & Tilch, B. (2009). A power law for the duration of high-flow states and its interpretation from a heterogeneous traffic flow perspective. European Physical Journal B, 68(4), 577-586. doi:10.1140/epjb/e2009-00092-8

Heydecker, B. G., & Addison, J. D. (2005). Analysis of dynamic traffic equilibrium with departure time choice. Transportation Science, 39(1), 39-57. doi:10.1287/trsc.1030.0075

Holland, J. (1975). Adaptation in Natural and Artificial Systems. Ann Arbor, Michigan: University of Michigan Press.

Hosseininasab, S.-M., & Shetab-Boushehri, S.-N. (2015). Integration of selecting and scheduling urban road construction projects as a time-dependent discrete network design problem. European Journal of Operational Research, 246(3), 762-771. doi:10.1016/j.ejor.2015.05.039

Hou, L., Lao, Y., Wang, Y., Zhang, Z., Zhang, Y., & Li, Z. (2013). Modeling freeway incident response time: A mechanism-based approach. Transportation Research Part C-Emerging Technologies, 28, 87-100. doi:10.1016/j.trc.2012.12.005

Hu, X., Daganzo, C., & Madanat, S. (2015). A reliability-based optimization scheme for maintenance management in large-scale bridge networks. Transportation Research Part C-Emerging Technologies, 55, 166-178. doi:10.1016/j.trc.2015.01.008

Hu, X. B., Chiu, Y. C., & Shelton, J. (2017). Development of a behaviorally induced system optimal travel demand management system. Journal of Intelligent Transportation Systems, 21(1), 12-25. doi:10.1080/15472450.2016.1171151

Huang, B., & Pan, X. (2007). GIS coupled with traffic simulation and optimization for incident response. Computers Environment and Urban Systems, 31(2), 116-132. doi:10.1016/j.compenvurbsys.2006.06.001

Huang, H.-J., & Li, Z.-C. (2007). A multiclass, multicriteria logit-based traffic equilibrium assignment model under ATIS. European Journal of Operational Research, 176(3), 1464-1477. doi:10.1016/j.ejor.2005.09.035

Huang, H. J., & Lam, W. H. K. (2002a). Modeling and solving the dynamic user equilibrium route and departure time choice problem in network with queues.

Transportation Research Part B-Methodological, 36(3), 253-273. doi:10.1016/s0191-2615(00)00049-7

Huang, H. J., & Lam, W. H. K. (2002b). Quasi-continuous dynamic equilibrium assignment with departure time choice in congested unidirectional pedestrian networks. Journal of the Operational Research Society, 53(1), 97-107.

Hwang, M. C., & Cho, H. J. (2016). The Classical Braess Paradox Problem Revisited: A Generalized Inverse Method on Non-Unique Path Flow Cases. Networks & Spatial Economics, 16(2), 605-622. doi:10.1007/s11067-015-9290-x

Iryo, T. (2011). Multiple equilibria in a dynamic traffic network. Transportation Research Part B-Methodological, 45(6), 867-879. doi:10.1016/j.trb.2011.02.010

Iryo, T. (2015). Investigating Factors for Existence of Multiple Equilibria in Dynamic Traffic Network. Networks & Spatial Economics, 15(3), 599-616. doi:10.1007/s11067-013-9206-6

Jaber, X., & O'Mahony, M. (2009). Mixed Stochastic User Equilibrium Behavior under Traveler Information Provision Services with Heterogeneous Multiclass, Multicriteria Decision Making. Journal of Intelligent Transportation Systems, 13(4), 188-198. doi:10.1080/15472450903299554

Jahn, O., Mohring, R. H., Schulz, A. S., & Stier-Moses, N. E. (2005). System-optimal routing of traffic flows with user constraints in networks with congestion. Operations Research, 53(4), 600-616. doi:10.1287/opre.1040.0197

Janson, B. N. (1991). DYNAMIC TRAFFIC ASSIGNMENT FOR URBAN ROAD NETWORKS. Transportation Research Part B-Methodological, 25(2-3), 143-161. doi:10.1016/0191-2615(91)90020-j

Jeong, y., Hanseon, & Joo-young, K. (2010). An Analytical Procedure to Estimate Non-recurrent Congestion caused by Freeway Accidents. [고속도로 교통사고로 인한 비반복 혼잡 추정 연구]. Korean Society of Transportation, 28(2), 45-52.

Jiang, L., Mahmassani, H. S., & Zhang, K. (2011). Congestion Pricing, Heterogeneous Users, and Travel Time Reliability Multicriterion Dynamic User Equilibrium

Model and Efficient Implementation for Large-Scale Networks. Transportation Research Record(2254), 58-67. doi:10.3141/2254-07

Jiang, N., & Xie, C. (2014). Computing and Analyzing Mixed Equilibrium Network Flows with Gasoline and Electric Vehicles. Computer-Aided Civil and Infrastructure Engineering, 29(8), 626-641. doi:10.1111/mice.12082

Jiang, R., & Chung, E. (2015). A Ramp Metering Strategy for Rapid Congestion Recovery. Computer-Aided Civil and Infrastructure Engineering, 30(3), 202-216. doi:10.1111/mice.12101

Jiang, X. M., & Adeli, H. (2003). Freeway work zone traffic delay and cost optimization model. Journal of Transportation Engineering-Asce, 129(3), 230-241. doi:10.1061/(asce)0733-947x(2003)129:3(230)

Jin, S., Hou, Z., Chi, R., & Hao, J. (2014). A Data-Driven Control Design Approach for Freeway Traffic Ramp Metering with Virtual Reference Feedback Tuning. Mathematical Problems in Engineering. doi:10.1155/2014/936531

Joksimovic, D., Blierner, M. C. J., Bovy, P. H. L., & Trb. (2005). Optimal toll design problem in dynamic traffic networks with joint route and departure time choice. Network Modeling 2005(1923), 61-72.

Kachroo, P., & Ozbay, K. (1998). Solution to the user equilibrium dynamic traffic routing problem using feedback linearization. Transportation Research Part B-Methodological, 32(5), 343-360. doi:10.1016/s0191-2615(97)00031-3

Kachroo, P., & Ozbay, K. (2005). Feedback control solutions to network level user-equilibrium real-time dynamic traffic assignment problems. Networks & Spatial Economics, 5(3), 243-260. doi:10.1007/s11067-005-3034-2

Kaufman, D. E., Nonis, J., & Smith, R. L. (1998). A mixed integer linear programming model for dynamic route guidance. Transportation Research Part B-Methodological, 32(6), 431-440. doi:10.1016/s0191-2615(98)00013-7

Kerner, B. S., & Klenov, S. L. (2004). Spatial-temporal patterns in heterogeneous traffic flow with a variety of driver behavioural characteristics and vehicle parameters. Journal of Physics a-Mathematical and General, 37(37), 8753-8788. doi:10.1088/0305-4470/37/37/001

Kim, H., Kim, W., Chang, G.-L., & Rochon, S. M. (2014). Design of Emergency Response System to Minimize Incident Impacts Case Study for Maryland District 7 Network. Transportation Research Record(2470), 65-77. doi:10.3141/2470-07

Kinoshita, A., Takasu, A., & Adachi, J. (2015). Real-time traffic incident detection using a probabilistic topic model. Information Systems, 54, 169-188. doi:10.1016/j.is.2015.07.002

Kitthamkesorn, S., & Chen, A. (2013). A path-size weibit stochastic user equilibrium model. Transportation Research Part B-Methodological, 57, 378-397. doi:10.1016/j.trb.2013.06.001

Kitthamkesorn, S., & Chen, A. (2014). Unconstrained weibit stochastic user equilibrium model with extensions. Transportation Research Part B-Methodological, 59, 1-21. doi:10.1016/j.trb.2013.10.010

Kiwiel, K. C. (2011). An alternating linearization bundle method for convex optimization and nonlinear multicommodity flow problems. Mathematical Programming, 130(1), 59-84. doi:10.1007/s10107-009-0327-0

Kolata, G. (Dec, 1990). What if They Closed 42d Street and Nobody Noticed? The New York Times. Retrieved from http://www.nytimes.com/1990/12/25/health/what-if-they-closed-42d-street-and-nobody-noticed.html

Konishi, H. (2004). Uniqueness of user equilibrium in transportation networks with heterogeneous commuters. Transportation Science, 38(3), 315-330. doi:10.1287/trsc.1030.0047

Kotsialos, A., & Papageorgiou, M. (2004a). Efficiency and equity properties of freeway network-wide ramp metering with AMOC. Transportation Research Part C-Emerging Technologies, 12(6), 401-420. doi:10.1016/j.trc.2004.07.016

Kotsialos, A., & Papageorgiou, M. (2004b). Nonlinear optimal control applied to coordinated ramp metering. Ieee Transactions on Control Systems Technology, 12(6), 920-933. doi:10.1109/tcst.2004.833406

Lafortune, S., Sengupta, R., Kaufman, D. E., & Smith, R. L. (1993). DYNAMIC SYSTEM-OPTIMAL TRAFFIC ASSIGNMENT USING A STATE-SPACE MODEL.

Transportation Research Part B-Methodological, 27(6), 451-472. doi:10.1016/0191-2615(93)90017-5

Landman, R. L., Hegyi, A., & Hoogendoorn, S. P. (2016). Coordinated Ramp Metering Based on On-Ramp Saturation Time Synchronization. Transportation Research Record(2484), 50-59. doi:10.3141/2484-06

Lawphongpanich, S. (2000). Simplicial with truncated Dantzig-Wolfe decomposition for nonlinear multicommodity network flow problems with side constraints. Operations Research Letters, 26(1), 33-41. doi:10.1016/s0167-6377(99)00059-0

Lee, H. Y. (2009). Optimizing schedule for improving the traffic impact of work zone on roads. Automation in Construction, 18(8), 1034-1044. doi:10.1016/j.autcon.2009.05.004

Lee, S. (2008). Multimodal, multiclass stochastic dynamic traffic assignment for evaluating information provision strategies. Journal of Advanced Transportation, 42(1), 45-64.

Lei, C., Lin, W.-H., & Miao, L. (2015). A Stochastic Emergency Vehicle Redeployment Model for an Effective Response to Traffic Incidents. Ieee Transactions on Intelligent Transportation Systems, 16(2), 898-909. doi:10.1109/tits.2014.2345480

Levin, M. W., Boyles, S. D., & Nezamuddin. (2015). Warm-starting dynamic traffic assignment with static solutions. Transportmetrica B-Transport Dynamics, 3(2), 99-113. doi:10.1080/21680566.2014.937788

Levin, M. W., Pool, M., Owens, T., Juri, N. R., & Waller, S. T. (2015). Improving the Convergence of Simulation-based Dynamic Traffic Assignment Methodologies. Networks & Spatial Economics, 15(3), 655-676. doi:10.1007/s11067-014-9242-x

Levinson, D., & Zhang, L. (2006). Ramp meters on trial: Evidence from the Twin Cities metering holiday. Transportation Research Part a-Policy and Practice, 40(10), 810-828. doi:10.1016/j.tra.2004.12.004

Li, A. C. Y., Nozick, L., Davidson, R., Brown, N., Jones, D. A., & Wolshon, B. (2013). Approximate Solution Procedure for Dynamic Traffic Assignment. Journal of

Transportation Engineering, 139(8), 822-832. doi:10.1061/(asce)te.1943-5436.0000518

Li, D., Ranjitkar, P., & Ceder, A. (2014). Integrated Approach Combining Ramp Metering and Variable Speed Limits to Improve Motorway Performance. Transportation Research Record(2470), 86-94. doi:10.3141/2470-09

Li, S.-G. (2008). Genetic algorithm for solving dynamic simultaneous route and departure time equilibrium problem. Transport, 23(1), 73-77. doi:10.3846/1648-4142.2008.23.73-77

Li, X., Lam, W. H. K., & Tam, M. L. (2013). New Automatic Incident Detection Algorithm Based on Traffic Data Collected for Journey Time Estimation. Journal of Transportation Engineering, 139(8), 840-847. doi:10.1061/(asce)te.1943-5436.0000566

Li, Y., & Chow, A. H. F. (2015). Optimisation of motorway operations via ramp metering and variable speed limits. Transportation Planning and Technology, 38(1), 94-110. doi:10.1080/03081060.2014.976982

Li, Y., Chow, A. H. F., & Cassel, D. L. (2014). Optimal Control of Motorways by Ramp Metering, Variable Speed Limits, and Hard-Shoulder Running. Transportation Research Record(2470), 122-130. doi:10.3141/2470-13

Lighthill, M. J., & Whitham, G. B. (1955). ON KINEMATIC WAVES .2. A THEORY OF TRAFFIC FLOW ON LONG CROWDED ROADS. Proceedings of the Royal Society of London Series a-Mathematical and Physical Sciences, 229(1178), 317-345. doi:10.1098/rspa.1955.0089

Lim, Y., & Heydecker, B. (2005). Dynamic departure time and stochastic user equilibrium assignment. Transportation Research Part B-Methodological, 39(2), 97-118. doi:10.1016/j.trb.2003.08.003

Lin, H., Roughgarden, T., Tardos, E., & Walkover, A. (2005). Braess's Paradox, Fibonacci numbers, and exponential inapproximability. In L. Caires, G. E. Italiano, L. Monteiro, C. Palamidessi, & M. Yung (Eds.), Automata, Languages and Programming, Proceedings (Vol. 3580, pp. 497-512).

Lin, H., Roughgarden, T., Tardos, E., & Walkover, A. (2011). STRONGER BOUNDS ON BRAESS'S PARADOX AND THE MAXIMUM LATENCY OF SELFISH ROUTING. Siam Journal on Discrete Mathematics, 25(4), 1667-1686. doi:10.1137/090769600

Lin, S. Y., & Lin, C. H. (1997). A computationally efficient method for nonlinear multicommodity network flow problems. Networks, 29(4), 225-244. doi:10.1002/(sici)1097-0037(199707)29:4<225::aid-net6>3.0.co;2-h

Liu, L., Zhu, L., & Yang, D. (2016). Modeling and simulation of the car-truck heterogeneous traffic flow based on a nonlinear car-following model. Applied Mathematics and Computation, 273, 706-717. doi:10.1016/j.amc.2015.10.032

Liu, M., & Frangopol, D. M. (2005). Balancing Connectivity of Deteriorating Bridge Networks and Long-Term Maintenance Cost through Optimization. Journal of Bridge Engineering, 10(4), 468-481. doi:10.1061/(asce)1084-0702(2005)10:4(468)

Liu, M., & Frangopol, D. M. (2006). Optimizing bridge network maintenance management under uncertainty with conflicting criteria: Life-cycle maintenance, failure, and user costs. Journal of Structural Engineering-Asce, 132(11), 1835-1845. doi:10.1061/(asce)0733-9445(2006)132:11(1835)

Liu, Q., Lu, J., Chen, S., & Zhao, K. (2014). Multiple Naive Bayes Classifiers Ensemble for Traffic Incident Detection. Mathematical Problems in Engineering. doi:10.1155/2014/383671

Liu, Y., Li, P., Wehner, K., & Yu, J. (2013). A Generalized Integrated Corridor Diversion Control Model for Freeway Incident Management. Computer-Aided Civil and Infrastructure Engineering, 28(8), 604-620. doi:10.1111/mice.12024

Liu, Z., Wang, S., & Meng, Q. (2014). Toll pricing framework under logit-based stochastic user equilibrium constraints. Journal of Advanced Transportation, 48(8), 1121-1137. doi:10.1002/atr.1255

Logghe, S., & Immers, L. H. (2008). Multi-class kinematic wave theory of traffic flow. Transportation Research Part B-Methodological, 42(6), 523-541. doi:10.1016/j.trb.2007.11.001

Long, J., Gao, Z., Orenstein, P., & Ren, H. (2012). Control Strategies for Dispersing Incident-Based Traffic Jams in Two-Way Grid Networks. Ieee Transactions on Intelligent Transportation Systems, 13(2), 469-481. doi:10.1109/tits.2011.2171035

Long, J., Szeto, W. Y., Shi, Q., Gao, Z., & Huang, H.-J. (2015). A nonlinear equation system approach to the dynamic stochastic user equilibrium simultaneous route and departure time choice problem. Transportmetrica a-Transport Science, 11(5), 388-419. doi:10.1080/23249935.2014.1003112

Lotito, P. A. (2006). Issues in the implementation of the DSD algorithm for the traffic assignment problem. European Journal of Operational Research, 175(3), 1577-1587. doi:10.1016/j.ejor.2005.02.029

Lou, Y., Yin, Y., & Lawphongpanich, S. (2011). Freeway service patrol deployment planning for incident management and congestion mitigation. Transportation Research Part C-Emerging Technologies, 19(2), 283-295. doi:10.1016/j.trc.2010.05.014

Lu, C., Zhao, Y., & Gong, J. (2015). Intelligent Ramp Control for Incident Response Using Dyna-Q Architecture. Mathematical Problems in Engineering. doi:10.1155/2015/896943

Lu, C.-C., & Mahmassani, H. S. (2008). Modeling User Responses to Pricing Simultaneous Route and Departure Time Network Equilibrium with Heterogeneous Users. Transportation Research Record(2085), 124-135. doi:10.3141/2085-14

Lu, C.-C., & Mahmassani, H. S. (2009). Dynamic Pricing with Heterogeneous Users Gap-Driven Solution Approach for Bicriterion Dynamic User Equilibrium Problem. Transportation Research Record(2090), 75-85. doi:10.3141/2090-09

Lu, J., Chen, S., Wang, W., & Ran, B. (2012). Automatic traffic incident detection based on nFOIL. Expert Systems with Applications, 39(7), 6547-6556. doi:10.1016/j.eswa.2011.12.050

Lu, J., Chen, S., Wang, W., & van Zuylen, H. (2012). A hybrid model of partial least squares and neural network for traffic incident detection. Expert Systems with Applications, 39(5), 4775-4784. doi:10.1016/j.eswa.2011.09.158

Lu, X.-Y., Varaiya, P., Horowitz, R., Su, D., & Shladover, S. E. (2011). Novel Freeway Traffic Control with Variable Speed Limit and Coordinated Ramp Metering. Transportation Research Record(2229), 55-65. doi:10.3141/2229-07

Luo, Z.-Q., Pang, J.-S., & Ralph, D. (1996). Mathematical Programs with Equilibrium Constraints (1 edition ed.). Cambridge, United Kingdom: Cambridge University Press.

Lund, A. S., & Pack, M. L. (2010). Dynamic Wide-Area Congestion and Incident Monitoring Using Probe Data. Transportation Research Record(2174), 1-9. doi:10.3141/2174-01

Luong, H. T., & Nagarur, N. N. (2005). Optimal maintenance policy and fund allocation in water distribution networks. Journal of Water Resources Planning and Management-Asce, 131(4), 299-306. doi:10.1061/(asce)0733-9496(2005)131:4(299)

Ma, W. T., Cheu, R. L., & Lee, D. H. (2004). Scheduling of lane closures using genetic algorithms with traffic assignments and distributed simulations. Journal of Transportation Engineering-Asce, 130(3), 322-329. doi:10.1061/(asce)0733-947x(2004)130:3(322)

Ma, Y., Zhang, W., Lu, J., & Yuan, L. (2014). Automated Generation of Traffic Incident Response Plan Based on Case-Based Reasoning and Bayesian Theory. Discrete Dynamics in Nature and Society. doi:10.1155/2014/920301

Ma, Y., Zhang, W., Xie, J., & Lu, J. (2014). Application of Real-Time Automated Traffic Incident Response Plan Management System: A Web Structure for the Regional Highway Network in China. Advances in Mechanical Engineering. doi:10.1155/2014/489623

Mahmassani, H., & Herman, R. (1984). DYNAMIC USER EQUILIBRIUM DEPARTURE TIME AND ROUTE CHOICE ON IDEALIZED TRAFFIC ARTERIALS. Transportation Science, 18(4), 362-384. doi:10.1287/trsc.18.4.362

Mahmassani, H. S. (2016). Autonomous Vehicles and Connected Vehicle Systems: Flow and Operations Considerations. Transportation Science, 50(4), 1140-1162. doi:10.1287/trsc.2016.0712

Mahmassani, H. S., & Mouskos, K. C. (1988). SOME COMPUTATIONAL EXPERIMENTS ON THE DIAGONALIZATION ALGORITHM FOR NETWORK ASSIGNMENT WITH ASYMMETRIC INTERACTIONS BETWEEN CARS AND TRUCKS. Transportation Research Part B-Methodological, 22(4), 275-290. doi:10.1016/0191-2615(88)90004-5

Mallikarjuna, C., & Rao, K. R. (2011). Heterogeneous traffic flow modelling: a complete methodology. Transportmetrica, 7(5), 321-345. doi:10.1080/18128601003706078

Marcotte, P., & Zhu, D. L. (2009). Existence and computation of optimal tolls in multiclass network equilibrium problems. Operations Research Letters, 37(3), 211-214. doi:10.1016/j.orl.2009.01.006

Marquez, A. C., Gomez, J. F., Leon, P. M. d., & Rosique, A. S. (2013). Modelling on-line reliability and risk to schedule the preventive maintenance of repairable assets in network utilities. Ima Journal of Management Mathematics, 24(4), 437-450. doi:10.1093/imaman/dps016

Marwali, M. K. C., & Shahidehpour, S. M. (1998). A deterministic approach to generation and transmission maintenance scheduling with network constraints. Electric Power Systems Research, 47(2), 101-113. doi:10.1016/s0378-7796(98)00055-8

Marwali, M. K. C., & Shahidehpour, S. M. (1999). Long-term transmission and generation maintenance scheduling with network, fuel and emission constraints. Ieee Transactions on Power Systems, 14(3), 1160-1165. doi:10.1109/59.780951

Mathew, T. V., & Sharma, S. (2009). Capacity Expansion Problem for Large Urban Transportation Networks. Journal of Transportation Engineering, 135(7), 406-415. doi:10.1061/(asce)0733-947x(2009)135:7(406)

Meng, Q., & Khoo, H. L. (2010). A Pareto-optimization approach for a fair ramp metering. Transportation Research Part C-Emerging Technologies, 18(4), 489-506. doi:10.1016/j.trc.2009.10.001

Meng, Q., & Khoo, H. L. (2012). A computational model for the probit-based dynamic stochastic user optimal traffic assignment problem. Journal of Advanced Transportation, 46(1), 80-94. doi:10.1002/atr.149

Meng, Q., & Liu, Z. (2012). Mathematical models and computational algorithms for probit-based asymmetric stochastic user equilibrium problem with elastic demand. Transportmetrica, 8(4), 261-290. doi:10.1080/18128601003736026

Meng, Q., & Wang, X. (2008). Sensitivity Analysis of Logit-Based Stochastic User Equilibrium Network Flows with Entry-Exit Toll Schemes. Computer-Aided Civil and Infrastructure Engineering, 23(2), 138-156.

Meng, Q., & Weng, J. X. (2013). Optimal subwork zone operational strategy for short-term work zone projects in four-lane two-way freeways. Journal of Advanced Transportation, 47(2), 151-169. doi:10.1002/atr.153

Mesa-Arango, R., & Ukkusuri, S. V. (2014). Modeling the Car-Truck Interaction in a System-Optimal Dynamic Traffic Assignment Model. Journal of Intelligent Transportation Systems, 18(4), 327-338. doi:10.1080/15472450.2013.806846

Meshkat, A., Zhi, M., Vrancken, J. L. M., Verbraeck, A., Yuan, Y., & Wang, Y. (2015). Coordinated ramp metering with priorities. Iet Intelligent Transport Systems, 9(6), 639-645. doi:10.1049/iet-its.2014.0207

Miwa, T., Okada, Y., & Morikawa, T. (2010). Applying a Structured Dispersion Parameter to Multiclass Stochastic User Equilibrium Assignment Model. Transportation Research Record(2196), 142-149. doi:10.3141/2196-15

Morcous, G., & Lounis, Z. (2005). Maintenance optimization of infrastructure networks using genetic algorithms. Automation in Construction, 14(1), 129-142. doi:10.1016/j.autcon.2004.08.014

Mounce, R. (2007). Existence of Equilibrium in a Continuous Dynamic Queueing Model for Traffic Networks. Paper presented at the 4th IMA International Conference on Mathematics in Transport, London, United Kingdom.

Msigwa, R. E., Lu, Y., Ge, Y. G., & Zhang, L. W. (2015). A smoothing approach for solving transportation problem with road toll pricing and capacity expansions. Journal of Inequalities and Applications. doi:10.1186/s13660-015-0759-4

Mun, J.-S. (2011). A solution algorithm for a dynamic deterministic user equilibrium assignment model with departure time choice. Transportation Planning and Technology, 34(5), 443-466. doi:10.1080/03081060.2011.586115

Nahapetyan, A., & Lawphongpanich, S. (2007). Discrete-time dynamic traffic assignment models with periodic planning horizon: system optimum. Journal of Global Optimization, 38(1), 41-60. doi:10.1007/s10898-006-9082-4

Nair, R., Mahmassani, H. S., & Miller-Hooks, E. (2011). A porous flow approach to modeling heterogeneous traffic in disordered systems. Transportation Research Part B-Methodological, 45(9), 1331-1345. doi:10.1016/j.trb.2011.05.009

Niazi, A. N., Badri, A., & Khoshnoud, S. (2015). Annual coordinated generation and transmission maintenance scheduling considering network constraints and energy not supplied. International Transactions on Electrical Energy Systems, 25(9), 1831-1847. doi:10.1002/etep.1935

Nie, Y. (2011). Multi-class percentile user equilibrium with flow-dependent stochasticity. Transportation Research Part B-Methodological, 45(10), 1641-1659. doi:10.1016/j.trb.2011.06.001

Orabi, W., & El-Rayes, K. (2012). Optimizing the Rehabilitation Efforts of Aging Transportation Networks. Journal of Construction Engineering and Management-Asce, 138(4), 529-539. doi:10.1061/(asce)co.1943-7862.0000445

Orcesi, A. D., & Cremona, C. F. (2010). A bridge network maintenance framework for Pareto optimization of stakeholders/users costs. Reliability Engineering & System Safety, 95(11), 1230-1243. doi:10.1016/j.ress.2010.06.013

Ortuzar, J. d. D., & Willumsen, L. G. (1990). Modeling Transport. England: John Wiley & Sons Ltd.

Osman, O., Codjoe, J., Ishak, S., Rodriguez, J., & Russell, M. (2015). Long-term evaluation of the operational performance of fixed time ramp metering control strategy: a freeway corridor study. Canadian Journal of Civil Engineering, 42(11), 910-918. doi:10.1139/cjce-2014-0489

Ouorou, A., & Mahey, P. (2000). A minimum mean cycle cancelling method for nonlinear multicommodity flow problems. European Journal of Operational Research, 121(3), 532-548. doi:10.1016/s0377-2217(99)00050-8

Pal, R., & Bose, I. (2009). An optimization based approach for deployment of roadway incident response vehicles with reliability constraints. European Journal of Operational Research, 198(2), 452-463. doi:10.1016/j.ejor.2008.09.010

Papageorgiou, M. (1990). DYNAMIC MODELING, ASSIGNMENT, AND ROUTE GUIDANCE IN TRAFFIC NETWORKS. Transportation Research Part B-Methodological, 24(6), 471-495. doi:10.1016/0191-2615(90)90041-v

Papageorgiou, M., & Kotsialos, A. (2002). Freeway ramp metering: An overview. Ieee Transactions on Intelligent Transportation Systems, 3(4), 271-281. doi:10.1109/tits.2002.806803

Papamichail, I., Kotsialos, A., Margonis, I., & Papageorgiou, M. (2010). Coordinated ramp metering for freeway networks - A model-predictive hierarchical control approach. Transportation Research Part C-Emerging Technologies, 18(3), 311-331. doi:10.1016/j.trc.2008.11.002

Pas, E. I., & Principio, S. L. (1997). Braess' paradox: Some new insights. Transportation Research Part B-Methodological, 31(3), 265-276. doi:10.1016/s0191-2615(96)00024-0

Patriksson, M. (1994). The Traffic Assignment Problem: Models and Methods. Utrecht, Netherlands: VSP.

Peng, F., & Ouyang, Y. (2012). Track maintenance production team scheduling in railroad networks. Transportation Research Part B-Methodological, 46(10), 1474-1488. doi:10.1016/j.trb.2012.07.004

Perrine, K. A., Lao, Y., Wang, J., & Wang, Y. (2015). Area-Wide Ramp Metering for Targeted Incidents: The Additive Increase, Multiplicative Decrease Method. Journal of Computing in Civil Engineering, 29(2). doi:10.1061/(asce)cp.1943-5487.0000321

Petersen, E. R. (1975). PRIMAL-DUAL TRAFFIC ASSIGNMENT ALGORITHM. Management Science, 22(1), 87-95. doi:10.1287/mnsc.22.1.87

Powell, W. B., & Sheffi, Y. (1982). THE CONVERGENCE OF EQUILIBRIUM ALGORITHMS WITH PREDETERMINED STEP SIZES. Transportation Science, 16(1), 45-55. doi:10.1287/trsc.16.1.45

Prashker, J. N., & Bekhor, S. (2004). Route choice models used in the stochastic user equilibrium problem: A review. Transport Reviews, 24(4), 437-463. doi:10.1080/0144164042000181707

Qian, Z., Shen, W., & Zhang, H. M. (2012). System-optimal dynamic traffic assignment with and without queue spillback: Its path-based formulation and solution via approximate path marginal cost. Transportation Research Part B-Methodological, 46(7), 874-893. doi:10.1016/j.trb.2012.02.008

Qian, Z., & Zhang, H. M. (2013). A Hybrid Route Choice Model for Dynamic Traffic Assignment. Networks & Spatial Economics, 13(2), 183-203. doi:10.1007/s11067-012-9177-z

Reilly, J., Samaranayake, S., Delle Monache, M. L., Krichene, W., Goatin, P., & Bayen, A. M. (2015). Adjoint-Based Optimization on a Network of Discretized Scalar Conservation Laws with Applications to Coordinated Ramp Metering. Journal of Optimization Theory and Applications, 167(2), 733-760. doi:10.1007/s10957-015-0749-1

Rezaee, K., Abdulhai, B., & Abdelgawad, H. (2013). Self-Learning Adaptive Ramp Metering Analysis of Design Parameters on a Test Case in Toronto, Canada. Transportation Research Record(2396), 10-18. doi:10.3141/2396-02

Richards, P. I. (1956). SHOCK-WAVES ON THE HIGHWAY. Operations Research, 4(1), 42-51. doi:10.1287/opre.4.1.42

Rosa, A., & Maher, M. (2002). Algorithms for Solving the Probit Path-Based SUE Traffic Assignment Problem with One or More User Classes. Paper presented at the Proceedings of the 15th ISTTT, Oxford: Pergamon.

Roughgarden, T., & Tardos, E. (2002). How bad is selfish routing? Journal of the Acm, 49(2), 236-259. doi:10.1145/506147.506153

Ryu, S., Chen, A., Xu, X., & Choi, K. (2014). Modeling Demand Elasticity and Route Overlapping in Stochastic User Equilibrium Through Paired Combinatorial Logit Model. Transportation Research Record(2429), 8-19. doi:10.3141/2429-02

Schroeder, B. J., & Rouphail, N. M. (2010). Estimating Operational Impacts of Freeway Work Zones on Extended Facilities. Transportation Research Record(2169), 70-80. doi:10.3141/2169-08

Schulz, A. S., & Stier-Moses, N. E. (2006). Efficiency and fairness of system-optimal routing with user constraints. Networks, 48(4), 223-234. doi:10.1002/net.20133

Scrimali, L. (2014). EVOLUTIONARY QUASI-VARIATIONAL INEQUALITIES AND THE DYNAMIC MULTICLASS NETWORK EQUILIBRIUM PROBLEM. Numerical Functional Analysis and Optimization, 35(7-9), 1225-1244. doi:10.1080/01630563.2014.895764

Shapiro, A., Dentcheva, D., & Ruszczynski, A. (2009). Lectures on Stochastic Programming Modeling and Theory. Philadelphia: The Society for Industrial and Applied Mathematics.

Sheffi, Y. (1984). Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods. Englewood Cliffs, N.J. 07632: Prentice-Hall, Inc.

Shen, J.-B., & Chang, M.-S. (2007). Stochastic optimal-control approach to automatic incident-responsive coordinated ramp control. Ieee Transactions on Intelligent Transportation Systems, 8(2), 359-367. doi:10.1109/tits.2007.893327

Shen, W., & Zhang, H. M. (2008). What Do Different Traffic Flow Models Mean for System-Optimal Dynamic Traffic Assignment in a Many-to-One Network? Transportation Research Record(2088), 157-166. doi:10.3141/2088-17

Shen, W., & Zhang, H. M. (2010). Pareto-improving ramp metering strategies for reducing congestion in the morning commute. Transportation Research Part a-Policy and Practice, 44(9), 676-696. doi:10.1016/j.tra.2010.07.002

Shen, W., & Zhang, H. M. (2014). System optimal dynamic traffic assignment: Properties and solution procedures in the case of a many-to-one network. Transportation Research Part B-Methodological, 65, 1-17. doi:10.1016/j.trb.2014.02.002

Sherali, H. D., & Subramanian, S. (1999). Opportunity cost-based models for traffic incident response problems. Journal of Transportation Engineering-Asce, 125(3), 176-185. doi:10.1061/(asce)0733-947x(1999)125:3(176)

Sheu, J.-B. (2007). Stochastic modeling of the dynamics of incident-induced lane traffic states for incident-responsive local ramp control. Physica a-Statistical Mechanics and Its Applications, 386(1), 365-380. doi:10.1016/j.physa.2007.08.005

Sheu, J. B. (2002). A stochastic optimal control approach to real-time, incident-responsive traffic signal control at isolated intersections. Transportation Science, 36(4), 418-434. doi:10.1287/trsc.36.4.418.550

Sheu, J. B., Chou, Y. H., & Chen, A. (2004). Stochastic modeling and real-time prediction of incident effects on surface street traffic congestion. Applied Mathematical Modelling, 28(5), 445-468. doi:10.1016/j.apm.2003.10.004

Sheu, J. B., Chou, Y. H., & Shen, L. J. (2001). A stochastic estimation approach to real-time prediction of incident effects on freeway traffic congestion. Transportation Research Part B-Methodological, 35(6), 575-592. doi:10.1016/s0191-2615(00)00011-4

Sheu, J. B., Chou, Y. H., & Weng, M. C. (2003). Stochastic system modeling and optimal control of incident-induced traffic congestion. Mathematical and Computer Modelling, 38(5-6), 533-549. doi:10.1016/s0895-7177(03)00249-8

Smilowitz, K., & Madanat, S. (2000). Optimal inspection and maintenance policies for infrastructure networks. Computer-Aided Civil and Infrastructure Engineering, 15(1), 5-13. doi:10.1111/0885-9507.00166

Song, Z., Yin, Y., & Lawphongpanich, S. (2009). Nonnegative Pareto-Improving Tolls with Multiclass Network Equilibria. Transportation Research Record(2091), 70-78. doi:10.3141/2091-08

Steinberg, R., & Zangwill, W. I. (1983). THE PREVALENCE OF BRAESS PARADOX. Transportation Science, 17(3), 301-318. doi:10.1287/trsc.17.3.301

Sun, C., Cheng, L., Zhu, S., & Chu, Z. (2015). Multiclass Stochastic User Equilibrium Model with Elastic Demand Considering Systematic and Accidental Delays. Transportation Research Record(2497), 1-11. doi:10.3141/2497-01

Szeto, W. Y., & Lo, H. K. (2006). Dynamic traffic assignment: Properties and extensions. Transportmetrica, 2(1), 31-52.

Tang, T. Q., Huang, H. J., Zhao, S. G., & Shang, H. Y. (2009). A new dynamic model for heterogeneous traffic flow. Physics Letters A, 373(29), 2461-2466. doi:10.1016/j.physleta.2009.05.006

Tang, Y. M., & Chien, S. I. J. (2008). Scheduling Work Zones for Highway Maintenance Projects Considering a Discrete Time-Cost Relation. Transportation Research Record(2055), 21-30. doi:10.3141/2055-03

Tawarmalani, M., & Li, Y. (2011). Multi-period Maintenance Scheduling of Tree Networks with Minimum Flow Disruption. Naval Research Logistics, 58(5), 507-530. doi:10.1002/nav.20455

Tian, Y., & Chiu, Y.-C. (2014). A Variable Time-Discretization Strategies-Based, Time-Dependent Shortest Path Algorithm for Dynamic Traffic Assignment. Journal of Intelligent Transportation Systems, 18(4), 339-351. doi:10.1080/15472450.2013.806753

Tong, C. O., & Wong, S. C. (2010). Heuristic algorithms for simulation-based dynamic traffic assignment. Transportmetrica, 6(2), 97-120. doi:10.1080/18128600802630281

Torres Guardia, L. E., & Sant'Anna, A. P. (2009). A separable nonlinear model for the multicommodity flow problem. Optimization Methods & Software, 24(6), 973-987. doi:10.1080/10556780903027351

Transportation, N. H. D. o. (2012). Local Public Agency Manual for the Development of Projects. In Work Zone Traffic Control.

Usberti, F. L., Lyra, C., Cavellucci, C., & Vizcaino Gonzalez, J. F. (2015). Hierarchical multiple criteria optimization of maintenance activities on power distribution networks. Annals of Operations Research, 224(1), 171-192. doi:10.1007/s10479-012-1182-6

Valiant, G., & Roughgarden, T. (2010). Braess's Paradox in Large Random Graphs. Random Structures & Algorithms, 37(4), 495-515. doi:10.1002/rsa.20325

Varia, H. R., & Dhingra, S. L. (2004). Dynamic user equilibrium traffic assignment on congested multidestination network. Journal of Transportation Engineering-Asce, 130(2), 211-221. doi:10.1061/(asce)0733-947x(2004)130:2(211)

Verhoef, E. T. (2005). Second-best congestion pricing schemes in the monocentric city. Journal of Urban Economics, 58(3), 367-388. doi:10.1016/j.jue.2005.06.003

Waller, S. T., Fajardo, D., Duell, M., & Dixit, V. (2013). Linear Programming Formulation for Strategic Dynamic Traffic Assignment. Networks & Spatial Economics, 13(4), 427-443. doi:10.1007/s11067-013-9187-5

Waller, S. T., & Ziliaskopoulos, A. K. (2006). A combinatorial user optimal Dynamic Traffic Assignment algorithm. Annals of Operations Research, 144(1), 249-261. doi:10.1007/s10479-006-0013-z

Wang, J. Y. T., & Ehrgott, M. (2013). Modelling route choice behaviour in a tolled road network with a time surplus maximisation bi-objective user equilibrium model. Transportation Research Part B-Methodological, 57, 342-360. doi:10.1016/j.trb.2013.05.011

Wang, Y., Kosmatopoulos, E. B., Papageorgiou, M., & Papamichail, I. (2014). Local Ramp Metering in the Presence of a Distant Downstream Bottleneck: Theoretical Analysis and Simulation Study. Ieee Transactions on Intelligent Transportation Systems, 15(5), 2024-2039. doi:10.1109/tits.2014.2307884

Wang, Y., Messmer, A., & Papageorgiou, M. (2015). Real-Time Freeway Traffic State Estimation and Incident Detection based on Extended Kalman Filter: An Overview. At-Automatisierungstechnik, 63(4), 243-264. doi:10.1515/auto-2015-0006

Wang, Y., Papageorgiou, M., Gaffney, J., Papamichail, I., Rose, G., & Young, W. (2010). Local Ramp Metering in Random-Location Bottlenecks Downstream of Metered On-Ramp. Transportation Research Record(2178), 90-100. doi:10.3141/2178-10

Wardrop, J. G. (1952). Some Theoretical Aspects of Road Traffic Research. Proceedings of the Institution of Civil Engineers, 1(3), 325-362.

Wei, C., Asakura, Y., & Iryo, T. (2014). Formulating the within-day dynamic stochastic traffic assignment problem from a Bayesian perspective. Transportation Research Part B-Methodological, 59, 45-57. doi:10.1016/j.trb.2013.11.004

Wie, B. W. (1998). A convex control model of dynamic system optimal traffic assignment. Control Engineering Practice, 6(6), 745-753. doi:10.1016/s0967-0661(98)00080-x

Wie, B. W., Friesz, T. L., & Tobin, R. L. (1990). DYNAMIC USER OPTIMAL TRAFFIC ASSIGNMENT ON CONGESTED MULTIDESTINATION NETWORKS. Transportation Research Part B-Methodological, 24(6), 431-442. doi:10.1016/0191-2615(90)90038-z

Wie, B. W., Tobin, R. L., & Friesz, T. L. (1994). THE AUGMENTED LAGRANGIAN METHOD FOR SOLVING DYNAMIC NETWORK TRAFFIC ASSIGNMENT MODELS IN DISCRETE-TIME. Transportation Science, 28(3), 204-220. doi:10.1287/trsc.28.3.204

Wong, G. C. K., & Wong, S. C. (2002). A multi-class traffic flow model - an extension of LWR model with heterogeneous drivers. Transportation Research Part a-Policy and Practice, 36(9), 827-841. doi:10.1016/s0965-8564(01)00042-8

Woodruff, D., & Watson, J.-P. (2011). Progressive Hedging Innovations for a Class of Stochastic Mixed-integer Resource Allocation Problems Computational Management Science, 8(4), 355-370.

Wu, J. H., Florian, M., & He, S. G. (2006). An algorithm for multi-class network equilibrium problem in PCE of trucks: Application to the SCAG travel demand model. Transportmetrica, 2(1), 1-9.

Wu, X., & Nie, Y. (2013). Solving the Multiclass Percentile User Equilibrium Traffic Assignment Problem A Computational Study. Transportation Research Record(2334), 75-83. doi:10.3141/2334-08

Wynter, L. (1996). The traffic assignment problem: Models and methods - Patriksson,M. Transportation Science, 30(3), 271-272.

Xiao, J., Gao, X., Kong, Q.-J., & Liu, Y. (2014). More robust and better: a multiple kernel support vector machine ensemble approach for traffic incident detection. Journal of Advanced Transportation, 48(7), 858-875. doi:10.1002/atr.1231

Xiao, J., & Liu, Y. (2012). Traffic Incident Detection Using Multiple-Kernel Support Vector Machine. Transportation Research Record(2324), 44-52. doi:10.3141/2324-06

Xie, G., Hoeft, B., & Grayson, G. (2012). Ramp Meters Evaluation: Using ITS Archived Data. Journal of Transportation Engineering-Asce, 138(4), 447-454. doi:10.1061/(asce)te.1943-5436.0000332

Xu, M., Meng, Q., & Liu, K. (2017). Network user equilibrium problems for the mixed battery electric vehicles and gasoline vehicles subject to battery swapping stations and road grade constraints. Transportation Research Part B-Methodological, 99, 138-166. doi:10.1016/j.trb.2017.01.009

Xu, X., & Chen, A. (2013). C-logit stochastic user equilibrium model with elastic demand. Transportation Planning and Technology, 36(5), 463-478. doi:10.1080/03081060.2013.818275

Xu, X., Chen, A., Zhou, Z., & Cheng, L. (2014). A multi-class mean-excess traffic equilibrium model with elastic demand. Journal of Advanced Transportation, 48(3), 203-222. doi:10.1002/atr.205

Yai, T., Iwakura, S., & Morichi, S. (1997). Multinomial probit with structured covariance for route choice behavior. Transportation Research Part B-Methodological, 31(3), 195-207. doi:10.1016/s0191-2615(96)00025-2

Yang, H. (1999). System optimum, stochastic user equilibrium, and optimal link tolls. Transportation Science, 33(4), 354-360. doi:10.1287/trsc.33.4.354

Yang, H., & Huang, H. J. (2004). The multi-class, multi-criteria traffic network equilibrium and systems optimum problem. Transportation Research Part B-Methodological, 38(1), 1-15. doi:10.1016/s0191-2615(02)00074-7

Yang, H., Wang, X., & Yin, Y. (2012). The impact of speed limits on traffic equilibrium and system performance in networks. Transportation Research Part B-Methodological, 46(10), 1295-1307. doi:10.1016/j.trb.2012.08.002

Yang, H., Ye, H., Li, X., & Zhao, B. (2015). Speed limits, speed selection and network equilibrium. Transportation Research Part C-Emerging Technologies, 51, 260-273. doi:10.1016/j.trc.2014.12.002

Yang, H., & Zhang, X. (2008). Existence of anonymous link tolls for system optimum on networks with mixed equilibrium behaviors. Transportation Research Part B-Methodological, 42(2), 99-112. doi:10.1016/j.trb.2007.07.001

Yang, H., & Zhang, X. N. (2002). Multiclass network toll design problem with social and spatial equity constraints. Journal of Transportation Engineering-Asce, 128(5), 420-428. doi:10.1061/(asce)0733-947x(2002)128:5(420)

Yang, H., Zhang, X. N., & Trb. (2003). Optimal toll design in second-best link-based congestion pricing. In Transporation Network Modeling 2003: Plannning and Administration (pp. 85-92).

Yang, I., & Jayakrishnan, R. (2012). Gradient Projection Method for Simulation-Based Dynamic Traffic Assignment. Transportation Research Record(2284), 70-80. doi:10.3141/2284-09

Ye, X. (2010). Accelerated Procedure of Multiclass Highway Traffic Assignment for Maryland Statewide Transportation Model. Transportation Research Record(2176), 84-91. doi:10.3141/2176-10

Yin, Y. F., Yang, H., & Trb. (2004). Optimal tolls with a multiclass, bicriterion traffic network equilibrium. In Transportation Research Record (pp. 45-52).

Yu, Q., Fang, D., & Du, W. (2014). Solving the logit-based stochastic user equilibrium problem with elastic demand based on the extended traffic network model. European Journal of Operational Research, 239(1), 112-118. doi:10.1016/j.ejor.2014.04.009

Zhang, A., Gao, Z., & Ren, H. (2011). Incident-based traffic congestion control strategy. Science China-Technological Sciences, 54(5), 1338-1344. doi:10.1007/s11431-011-4337-9

Zhang, G., & Wang, Y. (2013). Optimizing Coordinated Ramp Metering: A Preemptive Hierarchical Control Approach. Computer-Aided Civil and Infrastructure Engineering, 28(1), 22-37. doi:10.1111/j.1467-8667.2012.00764.x

Zhang, H. M., Jin, W. L., & Trb, T. R. B. (2002). Kinematic wave traffic flow model for mixed traffic. In Traffic Flow Theory and Highway Capacity 2002: Highway Operations, Capacity, and Traffic Control (pp. 197-204).

Zhang, K., Mahmassani, H. S., & Lu, C.-C. (2008). Probit-Based Time-Dependent Stochastic User Equilibrium Traffic Assignment Model. Transportation Research Record(2085), 86-94. doi:10.3141/2085-10

Zhang, K., Mahmassani, H. S., & Lu, C.-C. (2013). Dynamic pricing, heterogeneous users and perception error: Probit-based bi-criterion dynamic stochastic user equilibrium assignment. Transportation Research Part C-Emerging Technologies, 27, 189-204. doi:10.1016/j.trc.2012.05.001

Zhang, L., & Levinson, D. (2010). Ramp metering and freeway bottleneck capacity. Transportation Research Part a-Policy and Practice, 44(4), 218-235. doi:10.1016/j.tra.2010.01.004

Zhang, S., Tang, J., Wang, H., & Wang, Y. (2015). Enhancing Traffic Incident Detection by Using Spatial Point Pattern Analysis on Social Media. Transportation Research Record(2528), 69-77. doi:10.3141/2528-08

Zhang, T., Andrews, J., & Wang, R. (2013). Optimal Scheduling of Track Maintenance on a Railway Network. Quality and Reliability Engineering International, 29(2), 285-297. doi:10.1002/qre.1381

Zhang, X., Yang, H., & Huang, H.-J. (2008). Multiclass multicriteria mixed equilibrium on networks and uniform link tolls for system optimum. European Journal of Operational Research, 189(1), 146-158. doi:10.1016/j.ejor.2007.05.004

Zhao, D., Bai, X., Wang, F.-Y., Xu, J., & Yu, W. (2011). DHP Method for Ramp Metering of Freeway Traffic. Ieee Transactions on Intelligent Transportation Systems, 12(4), 990-999. doi:10.1109/tits.2011.2122257

Zheng, C., Chen, S., Wang, W., & Lu, J. (2013). Using Principal Component Analysis to Solve a Class Imbalance Problem in Traffic Incident Detection. Mathematical Problems in Engineering. doi:10.1155/2013/524861

Zheng, H., & Chiu, Y.-C. (2011). A Network Flow Algorithm for the Cell-Based Single-Destination System Optimal Dynamic Traffic Assignment Problem. Transportation Science, 45(1), 121-137. doi:10.1287/trsc.1100.0343

Zheng, H., Nava, E., & Chiu, Y.-C. (2014). Measuring Networkwide Traffic Delay in Schedule Optimization for Work-Zone Planning in Urban Networks. Ieee Transactions on Intelligent Transportation Systems, 15(6), 2595-2604. doi:10.1109/tits.2014.2318299

Zhou, Z., Chen, A., & Bekhor, S. (2012). C-logit stochastic user equilibrium model: formulations and solution algorithm. Transportmetrica, 8(1), 17-41. doi:10.1080/18128600903489629

Zhu, S., Kim, W., & Chang, G.-L. (2012). Design and Benefit-Cost Analysis of Deploying Freeway Incident Response Units Case Study for Capital Beltway in Maryland. Transportation Research Record(2278), 104-114. doi:10.3141/2278-12

Ziliaskopoulos, A. K. (2000). A linear programming model for the single destination System Optimum Dynamic Traffic Assignment problem. Transportation Science, 34(1), 37-49. doi:10.1287/trsc.34.1.37.12281

Zografos, K. G., Androutsopoulos, K. N., & Vasilakis, G. M. (2002). A real-time decision support system for roadway network incident response logistics. Transportation Research Part C-Emerging Technologies, 10(1), 1-18. doi:10.1016/s0968-090x(00)00043-7

Zverovich, V., & Avineri, E. (2015). Braess' paradox in a generalised traffic network. Journal of Advanced Transportation, 49(1), 114-138. doi:10.1002/atr.1269

APPENDIX A

FIVE RUNS OF RFO FOR TESTCASES SOLVED IN MS-NSV

Table 1: Objective and Time Consumption of Five Runs by RFO for Each Test Case of the Radial Network

Completion Date (*T*)	Run 1		Run 2		Run 3		Run 4		Run 5	
	Objective Value	Solving Time	Objective Value	Solving Time	Objective Value	Solving Time	Objective Value	Solving Time	Objective Value	Solving Time
15	233166	1.6 min	233166	1.45 min	233166	1.58 min	233166	1.52 min	233166	1.5 min
16	170591	3.25 min	170591	2.85 min	170591	3.02 min	170591	4.4 min	170591	4.95 min
17	101516	8.35 min	101516	4.93 min	101516	6.12 min	101516	5.45 min	101516	5.8 min
18	25644.7	7.1 min	25644.7	6.5 min	26547.7	5.92 min	25677.7	6.1 min	25647.7	4.97 min
19	19668.1	4.92 min	19067.3	12.87 min	19067.3	6.52 min	19067.4	6.52 min	19067.4	4.32 min
20	10889.6	6.15 min	10389.2	6.56 min	9888.26	7.4 min	9888.26	9.42 min	9888.07	7.42 min

Table 2: Objective and Time Consumption of Five Runs by RFO for Each Test Case of the Grid Network

Completion Date (*T*)	Run 1		Run 2		Run 3		Run 4		Run 5	
	Objective Value	Solving Time	Objective Value	Solving Time	Objective Value	Solving Time	Objective Value	Solving Time	Objective Value	Solving Time
14	143230	4.9 min	144071	4.2 min	143630	2.78 min	143033	5.23 min	143429	5.33 min
15	105997	3 min	105997	6.5 min	105991	5.71 min	105989	3.75 min	106000	3.5 min
16	67527.2	3.05 min	68704.6	5.43 min	67711.7	2.23 min	66211.3	3.25 min	67711.7	4.02 min
17	51773.6	5.87 min	51773.7	4.1 min	51773.7	7.92 min	51772	7.28 min	51772	8.18 min
18	37350	9.95 min	37350.5	4.53 min	37348.3	3.95 min	37350.3	13 min	38602	7.98 min
19	26921.2	7.53 min	26671.2	6.56 min	26921.4	6.58 min	26672.5	7.32 min	26671.4	6.25 min
20	15989.2	6.68 min	15988.9	5.22 min	15989.5	3.62 min	15989.2	3.75 min	15989.3	6.5 min
21	7809.33	4.48 min	7810.41	7.67 min	7809.9	3.65 min	7810.32	4 min	7809.11	6.7 min
26	1915.95	2.13 min	1915.51	2.17 min	1914.49	2.15 min	1915.75	1.97 min	1915.97	2.02 min
36	2631.65	2.6 min	2630.52	2.6 min	2631.65	2.57 min	2628.9	2.93 min	2631.65	2.65 min

Table 3: Objective and Time Consumption of Five Runs by RFO for Each Test Case of the Sioux Falls Network (10%)

Completion Date (*T*)	Run 1		Run 2		Run 3		Run 4		Run 5	
	Objective Value	**Solving Time**	**Objective Value**	**Solving Time**	**Objective Value**	**Solving Time**	**Objective Value**	**Solving Time**	**Objective Value**	**Solving Time**
19	237459	2.68 min	237494	4.63 min	237515	5.12 min	237487	2.17 min	237542	2.05 min
20	242532	4.67 min	242542	2.2 min	242518	4.47 min	242532	3 min	242542	2.58 min
21	247317	2.17 min	247325	2.63 min	247325	2.47 min	247325	2.58 min	247325	2.35 min
22	252140	2.57 min	252201	2.38 min	252201	2.6 min	252203	3.62 min	252140	2.77 min
23	260386	3.05 min	260302	2.87 min	260302	5.6 min	260318	2.75 min	260302	3.05 min
24	268500	2.87 min	268570	2.97 min	268498	6.53 min	268603	3.25 min	268631	3.03 min
25	277223	6.87 min	277258	3.35 min	277170	7.15 min	277339	3.07 min	277216	7.28 min
26	285841	7.65 min	285791	3.13 min	285841	4.12 min	285841	7.63 min	285843	3.35 min
27	294744	7.07 min	294744	4.75 min	294744	6.72 min	294573	5.3 min	294591	7 min
28	302933	7.92 min	303279	8.27 min	303278	8.92 min	303529	3.93 min	303396	7.5 min
29	311643	7.37 min	311643	8.82 min	311691	4.77 min	311447	8.37 min	311723	8.23 min
30	320849	5.45 min	320798	8.7 min	320522	8.27 min	320756	8.28 min	320798	8.75 min
31	329556	9.12 min	329368	10.77 min	329476	10.78 min	329430	8.18 min	329436	8.93 min
32	338608	10.93 min	338334	10.35 min	338860	9.08 min	338829	10.62 min	338665	9.35 min
33	349090	8.18 min	349265	2.67 min	347910	6.83 min	347897	3.83 min	349015	6.48 min
34	357086	8.43 min	357090	8.63 min	357045	9.5 min	357064	9.1 min	256866	10.38 min
35	366209	11.55 min	366280	9.47 min	366242	9.22 min	366256	11.11 min	366335	9.8 min
36	375665	9.62 min	375585	9.38 min	375633	9.57 min	375407	9.52 min	375516	10.18 min
37	385649	10.08 min	385649	11.1 min	385649	12.07 min	385649	10.95 min	358649	9.5 min
38	395879	10.53 min	395879	11.17 min	395879	10.75 min	395743	10.05 min	395861	12.45 min

Table 4: Objective and Time Consumption of Five Runs by RFO for Each Test Case of the Sioux Falls Network (20%)

Completion Date (T)	Run 1		Run 2		Run 3		Run 4		Run 5	
	Objective Value	**Solving Time**	**Objective Value**	**Solving Time**	**Objective Value**	**Solving Time**	**Objective Value**	**Solving Time**	**Objective Value**	**Solving Time**
26	429644	20.63 min	419612	19.38 min	429673	23.15 min	430182	18.85 min	430182	18.33 min
27	441983	24.33 min	439772	28.43 min	440444	34.22 min	441998	27.45 min	441053	55.08 min
28	446741	1.03 hr.	446946	25.7 min	446316	38.73 min	444821	1.13 hr.	447709	26.4 min
29	451943	44.5 min	451798	1.06 hr.	452415	41.88 min	451972	1.17 hr.	451404	23.28 min
30	461263	37.18 min	462005	53 min	465685	42.33 min	463749	22.77 min	460272	49.8 min
31	468504	1.17 hr.	468968	52.93 min	468724	53.72 min	468724	1.33 hr.	467886	1.18 hr.
32	474658	1.2 hr.	475698	33.17 min	475486	1.14 hr.	474658	1.34 hr.	474848	48.28 min
33	486761	47.63 min	485102	1.11 hr.	485800	1.23 hr.	487476	24.58 min	486772	44.37 min
34	495069	1.3 hr.	496514	1.25 hr.	496356	1.24 hr.	495278	1.21 hr.	495502	1.29 hr.
35	502656	1.43 hr.	502775	1.48 hr.	502704	1.31 hr.	502616	1.47 hr.	502656	1.3 hr.
36	513889	1.5 hr.	513445	1.04 hr.	512527	1.25 hr.	512588	1.24 hr.	514482	1.14 hr.
37	523158	1.41 hr.	521056	1.27 hr.	523197	1.28 hr.	521984	1.14 hr.	523766	1.35 hr.
38	547711	30.32 min	550335	38.5 min	535769	17.73 min	554759	52.37 min	558797	44.03 min
39	543698	26.23 min	544046	40.72 min	553827	46.6 min	560203	33.07 min	542010	38.35 min
40	568072	56.52 min	563160	38.82 min	575092	56.82 min	556089	39.6 min	551897	19.53 min
41	563869	43.07 min	562826	31.43 min	561137	30.55 min	581717	1.16 hr.	573486	40.75 min
42	571838	49.55 min	594626	1.02 hr.	567858	30.08 min	585013	47.62 min	588380	1.07 hr.

APPENDIX B

NETWORK TESTCASES SOLVED IN OCREC

Table 5: Simple Four-Node Network

Link ID	Initial node	Terminal node	Capacity	Free Flow Time	Alpha	Beta
1	1	2	600	50	2.4	4
2	1	3	50	1	2.4	4
3	2	4	50	1	2.4	4
4	3	4	600	50	2.4	4
5	3	2	60	40	2.4	4

Table 6: OD Demand for the Simple Four-Node Network

OD	Origin Node	Destination Node	Demand
1	1	4	40
2	3	4	20

Table 7: Square Network

Link ID	Initial Node	Terminal Node	Free-Flow Travel Time	Alpha	Beta	Capacity
1	1	2	500	0.15	4	6000
2	1	6	10	0.15	4	500
3	6	2	400	0.15	4	600
4	2	3	500	0.15	4	6000
5	2	7	10	0.15	4	500
6	7	3	400	0.15	4	600
7	3	4	500	0.15	4	6000
8	3	8	10	0.15	4	500
9	8	4	400	0.15	4	600
10	4	5	500	0.15	4	6000
11	4	9	10	0.15	4	500
12	9	5	400	0.15	4	600
13	5	10	10	0.15	4	500
14	6	7	500	0.15	4	6000
15	6	11	10	0.15	4	500
16	11	7	400	0.15	4	600
17	7	8	500	0.15	4	6000
18	7	12	10	0.15	4	500
19	12	8	400	0.15	4	600
20	8	9	500	0.15	4	6000
21	8	13	10	0.15	4	500

Link ID	Initial Node	Terminal Node	Free-Flow Travel Time	Alpha	Beta	Capacity
22	13	9	400	0.15	4	600
23	9	10	500	0.15	4	6000
24	9	14	10	0.15	4	500
25	14	10	400	0.15	4	600
26	10	15	10	0.15	4	500
27	11	12	500	0.15	4	6000
28	11	16	10	0.15	4	500
29	16	12	400	0.15	4	600
30	12	13	500	0.15	4	6000
31	12	17	10	0.15	4	500
32	17	13	400	0.15	4	600
33	13	14	500	0.15	4	6000
34	13	18	10	0.15	4	500
35	18	14	400	0.15	4	600
36	14	15	500	0.15	4	6000
37	14	19	10	0.15	4	500
38	19	15	400	0.15	4	600
39	15	20	10	0.15	4	500
40	16	17	500	0.15	4	6000
41	16	21	10	0.15	4	500
42	21	17	400	0.15	4	600
43	17	18	500	0.15	4	6000
44	17	22	10	0.15	4	500
45	22	18	400	0.15	4	600
46	18	19	500	0.15	4	6000
47	18	23	10	0.15	4	500
48	23	19	400	0.15	4	600
49	19	20	500	0.15	4	6000
50	19	24	10	0.15	4	500
51	24	20	400	0.15	4	600
52	20	25	10	0.15	4	500
53	21	22	500	0.15	4	6000
54	22	23	500	0.15	4	6000
55	23	24	500	0.15	4	6000
56	24	25	500	0.15	4	6000

Table 8: The Original OD Demand for Square Network

OD	Origin Node	Destination Node	Demand
1	1	7	400
2	1	13	400
3	1	19	400
4	1	25	400
5	7	13	400
6	7	19	400
7	7	25	400
8	13	19	400
9	13	25	400
10	19	25	400
11	6	7	200
12	11	13	200
13	16	19	200
14	21	25	200

Table 9: Simple Four-Node Network to Show the Nonconvexity of OCREC

Link ID	Initial node	Terminal node	Capacity	Free Flow Time	Alpha	Beta
1	1	2	8	50	2.4	4
2	1	3	6	1	2.4	4
3	2	4	6	1	2.4	4
4	3	4	8	50	2.4	4
5	3	2	7	10	2.4	4

Table 10: OD Demand for the Simple Four-Node Network to Show the Nonconvexity of OCREC

OD	Origin Node	Destination Node	Demand
1	1	4	6

APPENDIX C

TEST CASES SOLVED IN MS-MMN

Table 11: Square Network Test Cases for 10% of Links to Repair

Initial node	Terminal node	Days Required to Repair a Lane	Need to Repair in Case I?	Need to Repair in Case II?	Need to Repair in Case III?	Need to Repair in Case IV?	Need to Repair in Case V?
1	2	10	0	0	0	0	0
1	6	4	0	0	0	0	1
2	3	10	0	0	0	0	0
2	7	4	1	0	0	0	0
3	4	10	0	0	0	0	0
3	8	4	0	0	0	0	0
4	5	10	0	0	0	0	1
4	9	4	0	0	0	0	0
5	10	4	0	0	1	1	0
6	2	6	0	0	0	0	0
6	7	10	0	1	0	0	0
6	11	4	0	0	0	0	0
7	3	6	0	0	0	0	0
7	8	10	0	0	1	0	0
7	12	4	0	0	0	0	0
8	4	6	0	0	0	1	0
8	9	10	0	0	0	0	0
8	13	4	0	0	0	0	0
9	5	6	0	0	0	0	0
9	10	10	1	0	0	0	0
9	14	4	0	0	0	1	0
10	15	4	0	1	0	0	0
11	7	6	0	0	0	0	1
11	12	10	0	0	0	0	0

Initial node	Terminal node	Days Required to Repair a Lane	Need to Repair in Case I?	Need to Repair in Case II?	Need to Repair in Case III?	Need to Repair in Case IV?	Need to Repair in Case V?
11	16	4	1	0	0	0	0
12	8	6	1	0	0	1	0
12	13	10	0	0	0	0	1
13	9	6	0	0	0	0	0
13	14	10	0	0	0	0	0
13	18	4	0	0	0	0	0
14	10	6	0	0	0	1	0
14	15	10	0	0	0	0	0
14	19	4	0	0	0	0	0
15	20	4	0	1	0	0	0
16	12	6	1	0	0	0	0
16	17	10	0	0	0	0	0
16	21	4	0	0	0	0	0
17	13	6	0	0	0	0	0
17	18	10	0	0	0	0	0
17	22	4	0	0	1	0	0
18	14	6	1	0	0	0	0
18	19	10	0	0	0	0	0
18	23	4	0	0	0	0	0
19	15	6	0	0	1	0	0
19	20	10	0	1	0	0	0
19	24	4	0	0	0	0	1
20	25	4	0	0	1	0	0
21	17	6	0	0	0	0	0
21	22	10	0	0	0	1	0
22	18	6	0	0	0	0	0

Initial node	Terminal node	Days Required to Repair a Lane	Need to Repair in Case I?	Need to Repair in Case II?	Need to Repair in Case III?	Need to Repair in Case IV?	Need to Repair in Case V?
22	23	10	0	0	0	0	0
23	19	6	0	0	0	0	0
23	24	10	0	0	0	1	0
24	20	6	0	0	0	0	0
24	25	10	1	1	0	0	0

Table 12: Square Network Test Cases for 20% of Links to Repair

Initial node	Terminal node	Days Required to Repair a Lane	Need to Repair in Case I?	Need to Repair in Case II?	Need to Repair in Case III?	Need to Repair in Case IV?	Need to Repair in Case V?
1	2	10	0	1	0	0	0
1	6	4	0	0	0	0	0
2	3	10	1	0	0	0	0
2	7	4	0	0	0	0	0
3	4	10	0	1	0	0	0
3	8	4	0	0	0	0	0
4	5	10	0	0	0	0	0
4	9	4	1	0	0	0	0
5	10	4	1	1	0	0	0
6	2	6	1	1	0	0	1
6	7	10	0	1	0	1	1
6	11	4	1	0	0	1	0
7	3	6	0	0	0	0	0
7	8	10	0	1	1	1	0
7	12	4	0	0	0	0	0
8	4	6	0	0	1	0	1
8	9	10	0	0	0	0	1
8	13	4	0	0	0	0	1
9	5	6	0	0	0	0	0
9	10	10	0	1	1	0	0
9	14	4	0	0	1	0	0
10	15	4	1	0	0	0	0
11	7	6	0	0	1	1	1
11	12	10	0	0	0	1	1

Initial node	Terminal node	Days Required to Repair a Lane	Need to Repair in Case I?	Need to Repair in Case II?	Need to Repair in Case III?	Need to Repair in Case IV?	Need to Repair in Case V?
11	16	4	1	1	0	0	0
12	8	6	0	0	1	0	0
12	13	10	0	0	1	0	0
13	9	6	0	0	0	0	0
13	14	10	1	0	0	1	0
13	18	4	0	0	0	1	1
14	10	6	0	1	0	0	0
14	15	10	0	0	0	0	1
14	19	4	0	0	0	0	0
15	20	4	0	0	0	0	1
16	12	6	0	0	0	0	0
16	17	10	0	0	0	0	0
16	21	4	0	0	1	1	0
17	13	6	0	0	0	0	0
17	18	10	0	1	0	0	0
17	22	4	1	0	0	0	0
18	14	6	0	0	0	0	0
18	19	10	0	0	0	0	0
18	23	4	0	0	1	1	0
19	15	6	0	0	1	1	0
19	20	10	1	0	0	0	0
19	24	4	0	0	0	1	1
20	25	4	0	0	0	0	0
21	17	6	0	0	0	0	0
21	22	10	0	0	0	0	0
22	18	6	0	0	0	1	0

Initial node	Terminal node	Days Required to Repair a Lane	Need to Repair in Case I?	Need to Repair in Case II?	Need to Repair in Case III?	Need to Repair in Case IV?	Need to Repair in Case V?
22	23	10	0	0	0	0	0
23	19	6	0	0	1	0	1
23	24	10	0	0	1	0	1
24	20	6	1	0	0	0	0
24	25	10	0	1	0	0	0

Table 13: Square Network Test Cases for 30% of Links to Repair

Initial node	Terminal node	Days Required to Repair a Lane	Need to Repair in Case I?	Need to Repair in Case II?	Need to Repair in Case III?	Need to Repair in Case IV?	Need to Repair in Case V?
1	2	10	1	0	0	0	0
1	6	4	0	1	0	0	0
2	3	10	0	1	1	1	0
2	7	4	0	1	0	1	1
3	4	10	0	1	0	0	1
3	8	4	0	1	0	0	0
4	5	10	0	0	0	1	0
4	9	4	0	1	1	1	1
5	10	4	1	1	0	1	0
6	2	6	1	0	0	0	0
6	7	10	0	0	1	0	0
6	11	4	0	0	0	1	1
7	3	6	1	0	0	0	0
7	8	10	1	0	0	0	0
7	12	4	0	0	1	1	0
8	4	6	0	0	0	0	0
8	9	10	0	0	0	1	0
8	13	4	0	1	1	0	0
9	5	6	1	0	0	1	1
9	10	10	0	0	0	0	1
9	14	4	0	0	0	0	0
10	15	4	1	0	0	1	0
11	7	6	0	0	0	0	0
11	12	10	0	1	0	0	0

Initial node	Terminal node	Days Required to Repair a Lane	Need to Repair in Case I?	Need to Repair in Case II?	Need to Repair in Case III?	Need to Repair in Case IV?	Need to Repair in Case V?
11	16	4	1	1	1	1	1
12	8	6	1	0	0	0	0
12	13	10	0	1	0	0	1
13	9	6	0	1	0	1	0
13	14	10	0	0	1	1	1
13	18	4	0	0	1	0	1
14	10	6	1	0	0	0	0
14	15	10	0	0	0	1	0
14	19	4	1	1	1	1	0
15	20	4	0	0	0	0	1
16	12	6	0	1	0	0	0
16	17	10	1	0	1	0	0
16	21	4	0	0	0	0	0
17	13	6	0	0	0	0	1
17	18	10	1	0	0	0	1
17	22	4	0	1	1	0	0
18	14	6	0	1	1	1	1
18	19	10	0	0	0	0	0
18	23	4	0	0	1	0	0
19	15	6	0	0	1	0	0
19	20	10	0	0	1	0	1
19	24	4	0	0	0	0	1
20	25	4	1	0	1	0	0
21	17	6	0	0	0	0	0
21	22	10	1	0	0	0	1
22	18	6	0	0	1	0	0

Initial node	Terminal node	Days Required to Repair a Lane	Need to Repair in Case I?	Need to Repair in Case II?	Need to Repair in Case III?	Need to Repair in Case IV?	Need to Repair in Case V?
22	23	10	0	1	0	0	0
23	19	6	0	1	1	0	0
23	24	10	1	0	1	1	0
24	20	6	0	0	1	0	1
24	25	10	0	0	0	1	0

Table 14: OD Demand for Square Network

OD	Origin Node	Destination Node	Total Demand of UE and SO Flows
1	1	7	400
2	1	13	400
3	1	19	400
4	1	25	400
5	7	13	400
6	7	19	400
7	7	25	400
8	13	19	400
9	13	25	400
10	19	25	400
11	6	7	200
12	11	13	200
13	16	19	200
14	21	25	200

Table 15: Sioux Falls Network Test Case

Initial node	Terminal node	Capacity	Number of Lanes	Free Flow Time	B	Power	Days Required to Repair a Lane	Need to Repair in 10% Repair Case?	Need to Repair in 20% Repair Case?
1	2	25900.2	4	6	0.15	4	11	10%	20%
1	3	23403.47	4	4	0.15	4	9	0	0
2	1	25900.2	4	6	0.15	4	11	0	0
2	6	4958.181	1	5	0.15	4	8	1	0
3	1	23403.47	4	4	0.15	4	9	0	0
3	4	17110.52	3	4	0.15	4	8	0	0
3	12	23403.47	4	4	0.15	4	9	0	1
4	3	17110.52	3	4	0.15	4	8	0	0
4	5	17782.79	4	2	0.15	4	6	0	0
4	11	4908.827	1	6	0.15	4	7	0	0
5	4	17782.79	4	2	0.15	4	6	0	1
5	6	4947.995	1	4	0.15	4	7	0	0
5	9	10000	2	5	0.15	4	6	0	1
6	2	4958.181	1	5	0.15	4	8	0	0
6	5	4947.995	1	4	0.15	4	7	1	0
6	8	4898.588	1	2	0.15	4	5	0	0
7	8	7841.811	2	3	0.15	4	6	0	1
7	18	23403.47	4	2	0.15	4	7	0	0
8	6	4898.588	1	2	0.15	4	5	1	0
8	7	7841.811	2	3	0.15	4	6	0	0
8	9	5050.193	1	10	0.15	4	8	0	0
8	16	5045.823	1	5	0.15	4	7	0	1
9	5	10000	2	5	0.15	4	6	0	0
9	8	5050.193	1	10	0.15	4	8	0	0
9	10	13915.79	3	3	0.15	4	4	0	0

Initial node	Terminal node	Capacity	Number of Lanes	Free Flow Time	B	Power	Days Required to Repair a Lane	Need to Repair in 10% Repair Case?	Need to Repair in 20% Repair Case?
10	9	13915.79	3	3	0.15	4	4	0	0
10	11	10000	2	5	0.15	4	5	0	0
10	15	13512	3	6	0.15	4	6	0	0
10	16	4854.918	1	4	0.15	4	6	0	0
10	17	4993.511	1	8	0.15	4	8	0	1
11	4	4908.827	1	6	0.15	4	7	0	0
11	10	10000	2	5	0.15	4	5	0	1
11	12	4908.827	1	6	0.15	4	8	0	0
11	14	4876.508	1	4	0.15	4	5	0	0
12	3	23403.47	4	4	0.15	4	9	0	0
12	11	4908.827	1	6	0.15	4	8	0	0
12	13	25900.2	4	3	0.15	4	8	0	0
13	12	25900.2	4	3	0.15	4	8	0	0
13	24	5091.256	1	4	0.15	4	7	0	1
14	11	4876.508	1	4	0.15	4	5	0	1
14	15	5127.526	1	5	0.15	4	6	0	0
14	23	4924.791	1	4	0.15	4	5	0	0
15	10	13512	3	6	0.15	4	6	0	0
15	14	5127.526	1	5	0.15	4	6	0	0
15	19	14564.75	3	3	0.15	4	6	0	0
15	22	9599.181	2	3	0.15	4	7	0	0
16	8	5045.823	1	5	0.15	4	7	0	0
16	10	4854.918	1	4	0.15	4	6	0	1
16	17	5229.91	1	2	0.15	4	4	0	1
16	18	19679.9	4	3	0.15	4	5	1	1
17	10	4993.511	1	8	0.15	4	8	0	0
17	16	5229.91	1	2	0.15	4	4	1	0

Initial node	Terminal node	Capacity	Number of Lanes	Free Flow Time	B	Power	Days Required to Repair a Lane	Need to Repair in 10% Repair Case?	Need to Repair in 20% Repair Case?
17	19	4823.951	1	2	0.15	4	5	0	0
18	7	23403.47	4	2	0.15	4	7	0	0
18	16	19679.9	4	3	0.15	4	5	0	0
18	20	23403.47	4	4	0.15	4	10	0	0
19	15	14564.75	3	3	0.15	4	6	0	0
19	17	4823.951	1	2	0.15	4	5	1	0
19	20	5002.608	1	4	0.15	4	6	0	0
20	18	23403.47	4	4	0.15	4	10	0	0
20	19	5002.608	1	4	0.15	4	6	0	1
20	21	5059.912	1	6	0.15	4	7	0	0
20	22	5075.697	1	5	0.15	4	7	0	0
21	20	5059.912	1	6	0.15	4	7	0	0
21	22	5229.91	1	2	0.15	4	5	0	0
21	24	4885.358	1	3	0.15	4	6	0	0
22	15	9599.181	2	3	0.15	4	7	0	0
22	20	5075.697	1	5	0.15	4	7	0	1
22	21	5229.91	1	2	0.15	4	5	0	1
22	23	5000	1	4	0.15	4	6	0	0
23	14	4924.791	1	4	0.15	4	5	0	0
23	22	5000	1	4	0.15	4	6	0	0
23	24	5078.508	1	2	0.15	4	4	1	1
24	13	5091.256	1	4	0.15	4	7	1	0
24	21	4885.358	1	3	0.15	4	6	0	0
24	23	5078.508	1	2	0.15	4	4	0	0