The Dissertation Committee for Ehsan Jafari
certifies that this is the approved version of the following dissertation:

Network Modeling and Design: A Distributed Problem Solving Approach

Committee:

Stephen D. Boyles, Supervisor

Mark Hickman

Randy Machemehl

Christian Claudel

Avinash Unnikrishnan

Network Modeling and Design: A Distributed Problem Solving Approach

by

Ehsan Jafari

DISSERTATION

Presented to the Faculty of the Graduate School of
The University of Texas at Austin
in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

August 2017

Dedicated to Sayideh, Roza and Ava

Acknowledgments

I would like to express my deep appreciation and gratitude to my advisor, Dr. Stephen D. Boyles, for the patient guidance, motivation, immense knowledge, and mentorship he provided to me, all the way from when I started my PhD at The University of Texas at Austin, through to completion of this degree. I am truly fortunate to have had the opportunity to work with him. I must also express my sincere appreciation to Dr. Mark Hickman who introduced me to the field of transportation, and mentored me through my masters degree at the University of Arizona. I am also honored to have him as a member of my dissertation committee.

I would like to thank the rest of my thesis committee: Drs. Randy Machemehl, Christian Claudel, and Avinash Unnikrishnan for their insightful comments and encouragement, which incented me to widen my research from various perspectives. I am also grateful for the support by the National Science Foundation, and the Data-Supported Transportation Planning and Operations University Transportation Center.

Getting through my dissertation required more than academic support, and I have many, many people to thank for listening to and, at times, having to tolerate me over the past four years. A very special gratitude goes out to all the current and former members of the SPARTA Lab — Tarun, Michael, Venktesh, Sudesh, Rachel, John, Cesar, Dongxu, and Rahul. I learned a lot from all of you. I am also grateful to the administrative assistants who make sure nobody misses a deadline — Lisa Macias and Velma Vela have been particularly helpful to me in my time here.

My parents, brother and two sisters have been unwavering in their personal support. Thank you for supporting me spiritually throughout writing this dissertation and my life in general.

Finally, I'd be remiss if I didn't acknowledge the innumerable sacrifices made by my wife, Sayideh, in shouldering far more than her fair share of the parenting and household burdens standing by me through the good times and bad. This accomplishment would not have been possible without you. Thank you.

Abstract

Ehsan Jafari, Ph.D.
The University of Texas at Austin, 2017

Supervisor: Stephen D. Boyles

This dissertation is concerned with developing new solution algorithms for network modeling and design problems using a distributed problem solving approach. Network modeling and design are fundamental problems in the field of transportation science, and numerous transportation applications such as urban travel demand forecasting, congestion pricing, defining optimal toll values, and scheduling traffic lights all involve some form of network modeling or network design.

The first part of this dissertation focuses on developing a distributed scheme for the static traffic assignment problem, based on a spatial decomposition. The objective of the traffic assignment problem is to estimate traffic flows on a network and the resulting congestion considering the mutual interactions between travelers. A traffic assignment model takes as input the network topology, link performance functions, and a demand matrix indicating the traffic volume between each pair of origin-destination nodes. There are efficient algorithms to solve the traffic assignment problem, but, as computational hardware and algorithms advance, attention shifts to more demanding applications of the traffic assignment problem (bilevel programs whose solution often requires the solution of many traffic assignment problem instances as subproblems, accounting for forecasting errors with Monte Carlo simulation of input parameters, and broadening the geographic scope of models to the statewide or national levels.)

In Chapter 2, we propose a network contraction technique based on the theory of equilibrium sensitivity analysis. In the proposed algorithm, we replace the routes between each origin-destination (OD) pair with a single artificial link. These artificial links model the travel time between the origin and destination nodes of each OD pair as a function of network demands. The network contraction method can be advantageous in network design applications where many equilibrium problems must be solved for different design scenarios. The network contraction procedure can also be used to increase the accuracy of subnetwork analysis. The accuracy and complexity of the proposed methodology are evaluated using the network of Barcelona, Spain. Further, numerical experiments on the Austin, Texas regional network

validate its performance for subnetwork analysis applications.

Using this network contraction technique, we then develop a decentralized (distributed) algorithm for static traffic assignment in Chapter 3. In this scheme, which we term a decentralized approach to the static traffic assignment problem (DSTAP), the complete network is divided into smaller networks, and the algorithm alternates between equilibrating these networks as subproblems, and master iterations using a simplified version of the full network. The simplified network used for the master iterations is based on linearizations to the equilibrium solution for each subnetwork obtained using sensitivity analysis techniques. We prove that the DSTAP method converges to the equilibrium solution on the complete network, and demonstrate computational savings of 35-70% on the Austin network. Natural applications of this method are statewide or national assignment problems, or cities with rivers or other geographic features where subnetworks can be easily defined.

The second part of this dissertation, found in Chapter 4, deals with network design problems. In a network design problem, the goal is to optimize an objective function (minimize the travel time, pollution, maximize safety, social welfare, etc.) by making investment decisions subject to budget and feasibility constraints. Network design is a bi-level problem where the leader chooses the design parameters, and travelers, as followers, react to the leader's decision by changing their route. These problems are hard to solve, and distributed problem solving approach can be used to develop an efficient framework for scaling these problems.

In the proposed distributed algorithm for network design problems, different planning agencies may have different objective functions and priorities, while a regional agent (state or federal officials) allocates the finding between the urban cities. In this model, the urban planning agencies do their own planning and design independently while capturing the system-level effects of their local decisions and plans. The regional agent has limited and indirect authorities over the subnetworks through budget allocation. In addition to computational advantages for traditional bi-level network design problems, the proposed algorithm can be used to model the linkage between different entities for multi-resolution applications. We develop a solution algorithm based on a sensitivity-analysis heuristic, and test our algorithm on two case studies: a hypothetical network composed of two copies of Sioux Falls network, and the Austin regional network. We evaluate the correctness of the decentralized algorithm, and discuss the benefits of the algorithm in modeling the global impacts of local decisions. Furthermore, the implementation of distributed algorithm on Austin regional network demonstrates a computational saving of 22%.

Contents

List of figures

List of tables

1 Introduction

1.1 Distributed Problem Solving: Introduction

Complex and large-scale systems are usually made of multiple sub-systems working and interacting with each other to accomplish a global task. Such systems face the following basic challenge: how to implement and incorporate new demands, extensions and planning questions considering the limited amount of resource, capability and nonlinear behavior of entities. Traditional approaches have a *unitary* view: formulate the problem as a single problem and solve it using a single solver. Such approaches, however, have some major limitations: need a large amount of computational resources, especially for large scale problems; modeling the system may require knowledge and information from different fields; the system components may be essentially distributed, and due to data transmission and conversion costs, a unitary approach can be costly; extending the system may ask for substantial changes and modifications to the current system structure and entities, etc.

Distributed problem solving recommends partitioning the problem into smaller problems, called *subproblems (subtasks)*, and introducing multiple solvers, referred to as *local solvers (agents)*, to deal with the problem. In such a system, there is no central processor or central controller and tasks are divided between the local solvers. The local solvers, working on the subproblems, have limited access to local information, mainly from the assigned task, and none of them is equipped with global information or knowledge. The subproblems must be *cooperative* in the sense that, due to lack of sufficient information, a mechanism

should be implemented to share information between them. In addition, the local solvers should be *loosely coupled*: the local solvers spend most of their time on solving the assigned task rather than communicating with other solvers [Davis and Smith, 1983].

As mentioned by Davis and Smith [1983] and discussed further in Oates et al. [1997], distributed problem solving is not the same as *distributed processing*. Distributed problem solving involves solving a *single* problem by using multiple cooperative solvers with some degree of dependency. In distributed processing, however, we are facing with multiple problems which are usually independent and can be easily distributed over different solvers.

To the best of our knowledge, the term was first proposed and investigated in the field of artificial intelligence, called *distributed artificial intelligence* (DAI), and has found its way to applications such as cooperative information gathering [Oates et al., 1997], distributed scheduling of meetings [Zivan et al., 2014], channel-allocation problem in a wireless networks [Yeoh and Yokoo, 2012], and resource allocation in a disaster evacuation plan [Lass et al., 2008]. The benefits, implementation and requirements of a distrusted problem solver are discussed later, at the end of this chapter.

1.2 Problem Description

This dissertation is concerned with developing and applying distributed problem solving techniques for transportation network modeling and design problems. To this end, the work is partitioned into two parts.

Part I - Distributed problem solving and network modeling

The first part of this dissertation focuses on developing and using distributed problem solving for transportation network modeling. Formally, we consider traffic assignment problem on large-scale transportation networks comprised of a large number of links, nodes and zones. As a motivating example, consider a *statewide model*, where smaller subnetworks and urban areas compose the statewide network, as a case study. Such a network is geographically distributed which means data is decentralized, and also the modeling process may not be synchronized across urban models.

Two major approaches have been proposed to solve traffic assignment on large-scale networks: *exact* and *heuristic* modeling techniques. The former one tackles the problem by modeling all network components in detail. This task requires network details and demand data from all subnetworks, urban areas in case of statewide modeling. These pieces of information, after being converted to a consistent format, are stitched together to set up the regional network. This approach has two major shortcomings. First, collecting, transmitting and converting data on such a distributed and large-scale network can be costly. Second,

performing the assignment task on the resultant model requires a tremendous amount of computational power and time (traffic assignment on a simplified version of Texas statewide network takes three days).

The heuristic approaches use the ideas of network simplifications to answer modeling questions. These techniques simplify the subproblems or urban areas presented in the statewide model by removing links and nodes which are believed to be of less importance. The simplification process is done either by cutting out the minor links and nodes or representing them by a single link or node. As the name implies, this is a heuristic and generally the solution may not match the solution of the exact model.

On the other hand, due to lack of a consistent model and global information, the subnetworks (urban areas) develop their own model and evaluate the future plans by ignoring interdependency between subnetworks. Local decision and modifications, however, may not be optimal from a global perspective. For example, in a small scale, consider a plan to improve the signal timing plans in a small neighborhood as a solution to traffic congestion. The improved design may attract travelers from other parts of the network, resulting in a network state different that the one forecasted.

Here we propose a distributed problem solving approach to address the shortcomings of the exact and heuristic algorithms. In this scheme, which we term decentralized static traffic assignment (DSTAP), the network is divided into smaller networks, and the algorithm alternates between equilibrating these networks as subproblems, and master iterations using a simplified version of the full network. The simplified network used for the master iterations is based on linearizations to the equilibrium solution for each subnetwork obtained using sensitivity analysis techniques. We prove that the DSTAP method converges to the equilibrium solution on the full network, and demonstrate computational savings of $35 - 70\%$ on the Austin regional network. Natural applications of this method are statewide or national assignment problems, or cities with rivers or other geographic features where subnetworks can be easily defined.

The proposed distributed problem solving approach for traffic assignment task does not own the issues of exact and heuristic approaches: the proposed model benefits from decentralizing and parallelizing the assignment task over a large-scale network; also the decomposition, design and cooperation suggested by the algorithm ensures convergence to the correct solution.

DSTAP benefits from parallel computing, which is a general technique for reducing the running time of algorithms, by identifying problem components which can be solved independently, and brought together at a later point in time. Many algorithms for TAP naturally lend themselves to parallelization [Chen and Meyer, 1988, Karakitsiou et al., 2004]. For instance, the classic Frank-Wolfe algorithm can be parallelized by origin or destination when finding shortest paths and building the all-or-nothing link flow vector used in the search direction, and by link when determining the step size. The new way of parallelizing traffic assignment proposed in this dissertation is by geographic region rather than by origin.

Part II - Distributed problem solving and network design

In Part II, we focus on developing a distributed problem solver for transportation network design problems. The network design problem is concerned with making investment decisions to maximize a system objective function subject to budget and feasibility constraints. Network design can be formulated as a bi-level problem where the system manager selects the improvement plans, and users react by modifying their trip characteristics such as destination, mode, and route.

In this study, we consider a regional network composed of several subnetworks. The agents managing the subnetworks do their own network planning and design independently without taking into account that their local plans and investments may have broader impact than their subnetwork jurisdiction. This happens mainly because the internal concerns are their first priority. The agent managing the regional network (state or federal officials) is responsible for allocating funding to subnetworks and have limited and indirect authorities over the subnetworks. The transportation funds allocation to different subnetworks within a state has impacts at multiple levels. While such decisions are intensely political, well-grounded engineering models can help quantify the impacts of these policies, and guide allocation decisions. This can be modeled as a distributed network design problem, which can answer important questions about whether projects in different areas complement each other, measuring the impact of projects in one region on another, and so forth.

The proposed model for decentralized network design problem partitions the design problem into a design problem on a simplified version of the complete network and several smaller design problems (subnetwork design problems). The critical component of the proposed model is capturing the interactions between the partitioned design problems. This makes it possible to choose design variables while capturing the interactions between entities and understanding the system-level effects. The problem is formulated as a four-level network design problem, and a solution algorithm based on a sensitivity analysis heuristic is developed to solve the problem.

The proposed distributed network design model has the following main advantages: first, it can simplify the traditional bi-level network design problem; second, it increases awareness of local agents by representing the effects of local improvements on the regional level; third, it helps in allocating the federal or state funds to subnetworks by estimating the benefits from both local and global scales; and fourth, the model makes it possible to solve a design problem over a region with different and even conflicting objectives.

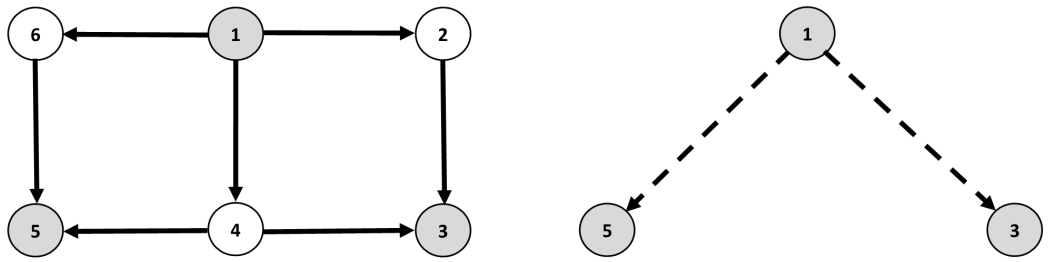


Figure 1.1: The complete network (left) has 6 nodes and 7 links, while the contracted network (right) has just 3 nodes representing the zones, and 2 artificial links representing each OD pair.

1.3 Informal Review

In this section we briefly introduce the problems discussed in this dissertation.

Chapter 2: *Network Contraction*

In this chapter, we review the general techniques developed in the literature for replacing a transportation network with a simpler network with less detail. The proposed algorithms for network contraction are usually *link extraction* or *link abstraction*. The former refers to simplifying a network by simply cutting out some links and nodes, which are believed to be of less importance. The latter, however, tries to simplify the network by aggregating portions of the network and representing a group of links and nodes with a single link or node. Network contraction, in general, is a trade-off between the network complexity and accuracy: higher contraction degree results in a simpler network with lower computational complexity but higher error and vise versa.

In Chapter 2, we propose a network contraction technique based on the theory of equilibrium sensitivity analysis. In the proposed algorithm, we replace the routes for each origin-destination (OD) pair with a single artificial link. These artificial links model the travel time between the origin and destination nodes of each OD pair as a function of network demands. Figure 2.2 illustrates the essence of this network contraction technique on a small network with two OD pairs: 1–5 and 1–3.

Let Υ_w denote the travel time on artificial link created for OD pair w. Using $\hat{\mathbf{d}}$ to denote the demand vector at the equilibrium solution $\hat{\mathbf{x}}$, and $\tilde{\mathbf{d}}$ to denote the perturbed demand vector, we write Υ_w using the first-order Taylor expansion:

$$\Upsilon_w(\tilde{\mathbf{d}}) = \hat{T}_w + \langle \nabla \hat{T}_w, \tilde{\mathbf{d}} - \hat{\mathbf{d}} \rangle \tag{1.1}$$

where $\hat{T}_w$ is the equilibrium travel time between the origin-destination nodes of OD pair w at flow $\hat{d}$,$\nabla\hat{T}_w$ is the gradient vector of $\hat{T}_w$ with respect to OD flows evaluated at $\hat{\mathbf{x}}$, and $\langle x, y\rangle$ is the inner product of vectors x and y.

In equation (1.1), the only unknowns to be estimated for OD pair w are the components of the gradient vector $\nabla\hat{T}_w$. The entry of the gradient vector $\nabla\hat{T}_w$ associated with OD pair u, i.e. $\partial\hat{T}_w/\partial d_u$, shows the derivative of $\hat{T}_w$ with respect to d_u evaluated at $\hat{\mathbf{x}}$.

In this chapter, we prove that these interactions are symmetric under the assumption that OD paths remain unchanged: a small change in $\hat{d}_u$ has the same impact on $\hat{T}_w$ as a small change in $\hat{d}_w$ would have on $\hat{T}_u$, i.e.,

$$\frac{\partial\hat{T}_w}{\partial d_u} = \frac{\partial\hat{T}_u}{\partial d_w} \tag{1.2}$$

This symmetry property can be used to reduce the number of unknown parameters needed to be estimated.

Later, we reformulate the linear system of equations defining these sensitivities as the solution to a convex programming problem, which can be solved by making minor modifications to static user equilibrium algorithms. The proposed convex program is essentially a static traffic assignment problem on a modified network with linear cost functions which can be solved efficiently.

Chapter 3: *Static traffic Assignment: A Decentralized Approach*

The third chapter of this dissertation develops a decentralized (distributed) algorithm for static traffic assignment based on the idea of distributed problem solving. The proposed decentralized approach partitions the assignment problem of a regional network into smaller problems. More precisely, for a regional network with $|U|$ subnetworks, the decentralized approach divides it into $|U| + 1$ smaller problems: one *master problem* and $|U|$ *subproblems*. The master problem solves a simplified version of the regional network, called *aggregated regional network*, where subnetworks are replaced with *artificial regional links* to capture the dynamics of the urban networks in an aggregated fashion. These artificial links represent first-order Taylor series approximations of OD travel time based on the equilibrium sensitivity analysis developed in Chapter 2. Each subproblem performs traffic assignment on a modified version of one of the subnetworks where *artificial urban links*, similar to those created in aggregated regional network, are created to model the interactions between subnetworks. Figure 1.2 shows the Texas statewide network, and Figure 1.3 describes the structure of the proposed distributed solver.

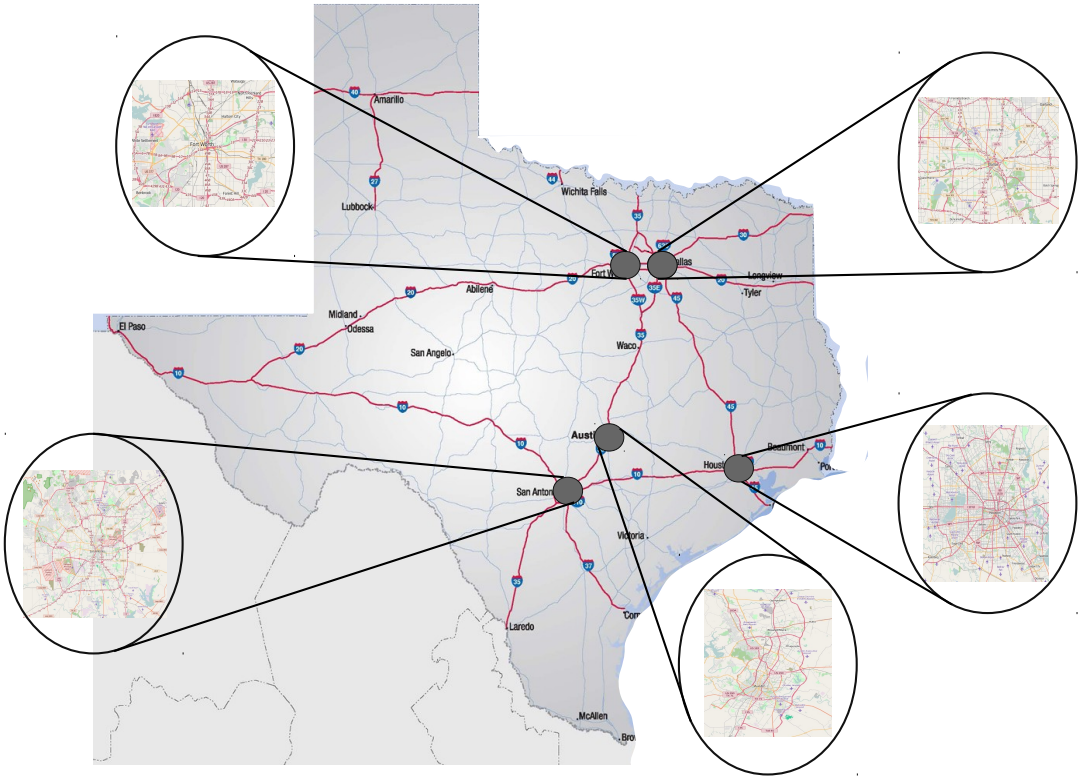


Figure 1.2: A regional network with 5 subnetworks. Solving this problem is practically and computationally challenging.

Each iteration of the proposed decentralized algorithm starts with solving the assignment problem of the aggregated regional network in the master problem. The assigned flow to each artificial regional link is the amount of flow attracted to the associated subnetwork based on the current traffic conditions of the regional and subnetworks. Each subproblem, then, solves the assignment problem on one of the subnetworks by incorporating the demand assigned to its artificial regional links in the master problem and artificial urban links in other subnetworks. As the last step at each iteration, the assignment solution of subproblems are used to update the parameters of the artificial regional and urban links. This process is repeated until a measure of convergence is satisfied.

Figure 1.4 describes the flowchart of the proposed decentralized traffic assignment method, called *DSTAP*, where subproblems are solved in parallel.

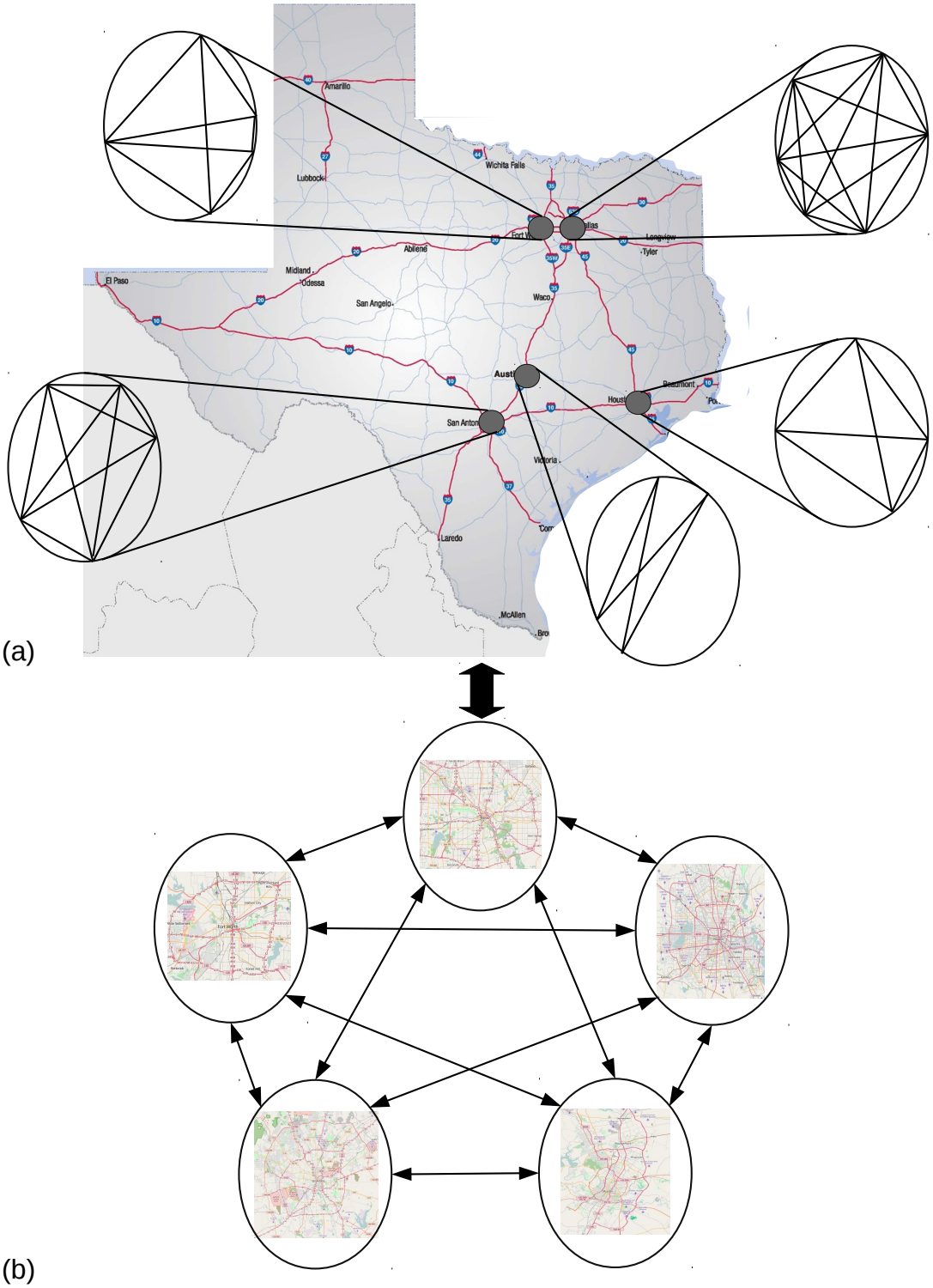


Figure 1.3: Distributed problem solving approach for statewide traffic assignment: (a) aggregated statewide network solved as the master problem; (b) subnetworks which are solved cooperatively in parallel.

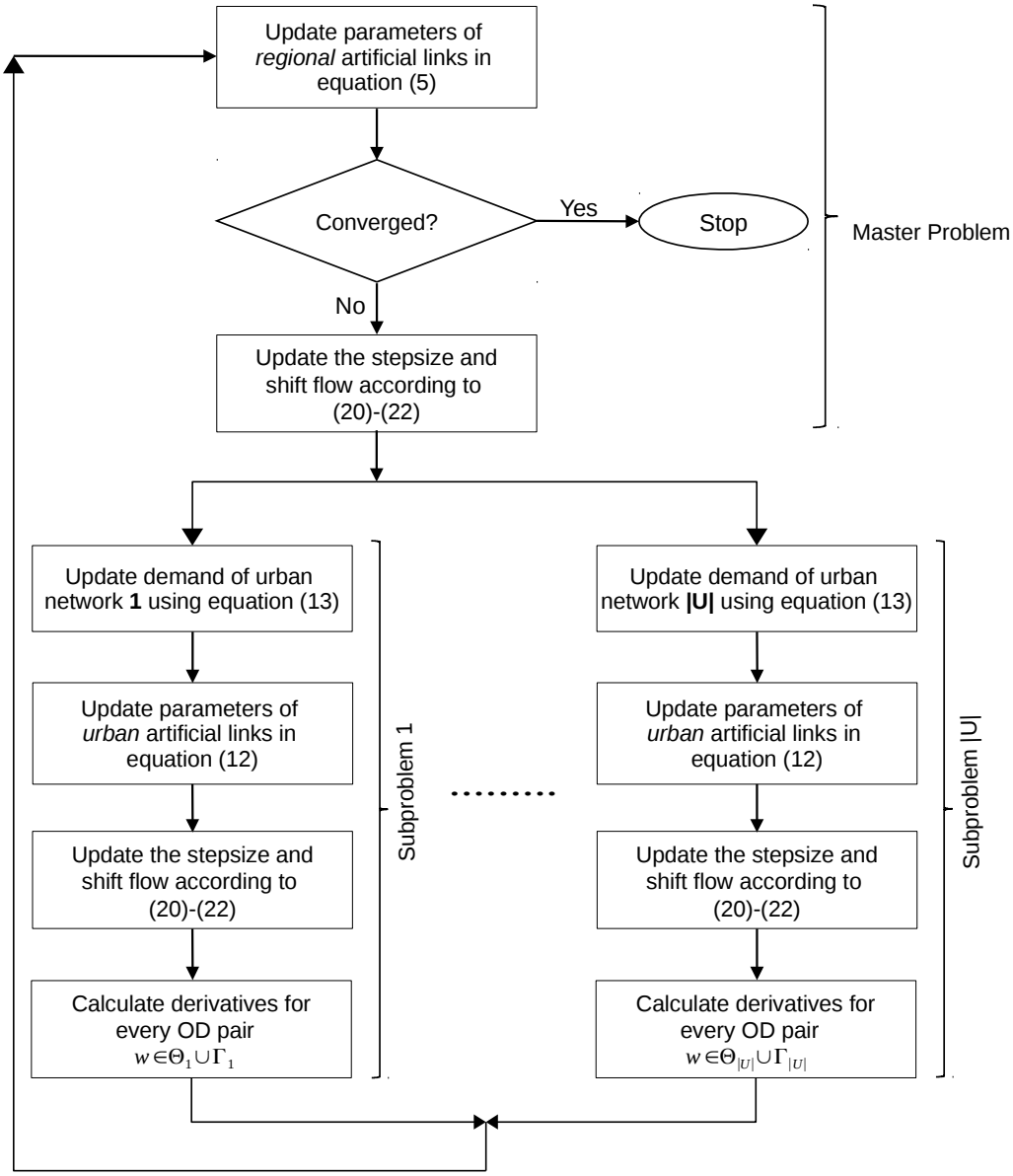


Figure 1.4: Flowchart of DSTAP in iteration $k+1$. The master problem starts by updating the parameters of the artificial regional links based on the solution of subproblems at iteration k. If the convergence condition is not satisfied, then aggregated regional network is solved and the assigned demand to artificial regional links are calculated. DSTAP continues by calling the $|U|$ subproblems. The subproblems are solved in parallel by incorporating assigned regional demand and demand from other urban networks into their OD pairs. Finally the parameters of the artificial regional and urban links are recalculated for iteration $k+2$.

1.3.1 Chapter 4: *Network Design Problem: A Decentralized Approach*

Chapter 4 of this dissertation deals with developing a decentralized algorithm for continuous network design problem (CNDP). Similar to the work discussed in Chapter 3, we consider a regional network composed of several subnetworks (urban cities). A regional agent has a budget to be distributed between the subnetworks and some regional projects in an equitable way. The urban cities are managed independently and the exact design plan at each urban city is not known to the regional agent.

The regional agent solves a CNDP on a simplified version of the regional network in which all urban cities are replaced with artificial links between the endpoints used by regional demand. The artificial links have a linear cost function stating the urban travel time as a function of regional flow and funding allocated to the urban cities. Each subproblem solves a CNDP on one subnetwork where the OD pairs with regional demand (those modeled in the regional network with artificial links) are modeled with elastic demand. This is due to the fact that regional demands traveling through urban cities have flexibility, and can change their entrance and/or exit points by evaluating the urban congestion.

The problem is formulated as a four-level NDP where the regional agent, deciding about the regional link improvements and funding allocation, forms the first level, a user equilibrium (UE) problem, modeling the route choice of regional demand as a function of first level decisions and urban improvements, form the second level, in the third level we have urban agents designing local plans using the funding allocated to them from the regional agent, and finally the last level models the route choice behavior of urban demands formulated as a UE problem with elastic demand.

We solve the problem using a sensitivity analysis based heuristic algorithm. Each iteration of the algorithm starts by updating the regional agent's decision variables. This step is performed using sensitivity analysis information from other levels. Next, the regional traffic is assigned to the regional network following the Wardrap's first principle. This step will also determine the external demand to each subnetwork. As the next step, the subnetwork design problems are solved using the current budget allocated to them from the regional agent. The solution algorithm for each subnetwork design problems is also based on a sensitivity analysis heuristic. Finally, a set of UE problems with elastic demand are solved on subnetworks.

1.4 Dissertation Structure

The rest of the dissertation is organized as follows.

Chapter 2 introduces the network contraction notion and reviews the proposed network contraction methods in the literature. Then the proposed network contraction technique in this dissertation, which is based on the equilibrium sensitivity analysis, is formulated. We prove a symmetry property for the sensitiv-

ity parameters, and reformulate the linear system of equations defining these sensitivities as the solution to a convex programming problem, which can be solved by making minor modifications to static user equilibrium algorithms. A heuristic is proposed to capture the interactions between the OD travel times and network flows. The accuracy and complexity of the proposed methodology are evaluated using the network of Barcelona, Spain. Further, numerical experiments on the Austin, Texas regional network validate its performance for subnetwork analysis applications.

Chapter 3 provides an introduction to static traffic assignment and its shortcomings when applied to a large scale network. Then a decentralized approach, which is based on the network contraction technique proposed in Chapter 2, is developed. The convergence of the decentralized approach is investigated using the *global convergence theorem*. Numerical examples on a regional network from Austin, Texas, evaluates the convergence properties and computational advantages of the proposed decentralized traffic assignment approach.

In Chapter 4, we mathematically introduce the network design problem and discuss different versions of the problem along with the most recent findings. Then the proposed decentralized scheme for network design problem is introduced and formulated. As the next section, we discuss the solution algorithm. Finally, the simulation results on a hypothetical network composed of two copied of the Sioux Falls network and also Austin regional network are presented.

Chapter 5 concludes this dissertation by summarizing the contributions of this work, and discusses possible extensions of the work for future research.

Part I:
Network Modeling: A Distributed Problem Solving Approach

2

Network Contraction

Calculating equilibrium sensitivity on a bush can be done very efficiently, and serve as the basis for a network contraction procedure. The contracted network (a simplified network with a few nodes and links) approximates the behavior of the full network but with less complexity. The network contraction method can be advantageous in network design applications where many equilibrium problems must be solved for different design scenarios. The network contraction procedure can also be used to increase the accuracy of subnetwork analysis. This method requires calculating travel time derivatives between two nodes, with respect to the demand between them, assuming that the flow distributes in a way that equilibrium is maintained. Previous research describes two methods for calculating these derivatives. This chapter presents a third method, which is simpler, faster, and just as accurate. The method presented in this chapter reformulates the linear system of equations defining these sensitivities as the solution to a convex programming problem, which can be solved by making minor modifications to static user equilibrium algorithms. In addition, the model is extended to capture the interactions between the path travel times and network flows, and a heuristic is proposed to compute these interactions. The accuracy and complexity of the proposed methodology are evaluated using the network of Barcelona, Spain. Further, numerical experiments on the Austin, Texas regional network validate its performance for subnetwork analysis applications.

2.1 Introduction

Static traffic assignment remains the most common network equilibrium model in practice: its favorable mathematical properties are well-known, practitioners are experienced at calibrating and interpreting the results, and it can be solved quickly. In particular, algorithms based on bushes — a concept dating to Dial [1970], and first applied to the static equilibrium problem in Dial [1999b] — exploit the acyclicity of the equilibrium flows to solve for equilibrium rapidly.

An important consequence of Beckmann's formulation [Beckmann et al., 1956] is that a meaningful sensitivity analysis can be undertaken. In the context of network equilibria, sensitivity analysis refers to determining a functional relationship between the travel times and demands without re-solving the network equilibrium problem. This functional relationship can be used to represent the network by connecting each origin-destination (OD) pair with a single artificial link and removing all the intermediate nodes and links.

The main reason to develop such network contraction techniques is reducing the computational burden of solving many network equilibrium problems [Friesz, 1985]. Still this computational motivation seems to be valid. As an example, consider a network design problem which is formulated as a bi-level problem. The master problem deals with computing some design parameters, and in the equilibrium subproblem, the travelers modify their route choice in response to design parameters set in the master problem. Such problems require solving hundreds or thousands of the network equilibrium problem as a subproblem which can be computationally expensive even for modern algorithms such as Algorithm B [Dial, 2006], TAPAS [Bar-Gera, 2010], or LUCE [Gentile, 2014].

As another application, consider the case of studying a number of alternatives, such as a new signal timing plan or converting a one-way local street to two ways, in a region of interest. The impacts of these policies are expected to be local, and solving the whole regional network may be unnecessary. Subnetwork analysis is commonly used to avoid incurring the computational burden of regional modeling. In practice, subnetwork modeling usually involves extracting a small component of a regional network, allowing the boundary nodes of the subnetwork to serve as origins and destinations, and estimating the subnetwork trip table from an equilibrium solution on the regional network. Xie et al. [2010] use entropy-maximization to identify these trips between the boundary points of the subnetwork. Effectively, this forms a "fixed" boundary condition which neglects diversion effects due to changes in the subnetwork.

Boyles [2012] adopts a different approach, using bush-based sensitivity analysis to simplify the regional network, rather than delete it entirely. In this way, diverting flows can be approximated while still retaining most of the computational advantage of subnetwork modeling; the boundary is less rigidly enforced.

The chief advantage of this procedure is that it captures diversion in a behavioral manner, based on drivers choosing routes to minimize their travel time. This contrasts with the fixed-boundary approach, where a common question is "how large does the subnetwork need to be to capture diversion?" While natural, this question is a bit of a red herring. Consider a downtown area served by two roughly parallel freeways (Figure 2.1). Even at large distances from the downtown area, there are drivers whose origins are roughly equidistant from these two freeways. These drivers' choice between these freeways depends on travel patterns downtown, regardless of how distant their origin is: the fundamental issue is modeling route choice, not simply capturing a large enough area. By integrating one model into another, "smoothing" the boundary, much faster convergence and greater accuracy can be seen without needing a large subnetwork.

Creating such a model requires estimation of a number of sensitivity parameters. Boyles [2012] provided two methods to calculate these parameters: one is reminiscent of resistive circuit analysis, and only applies when the equilibrium bushes are planar. The other is based on iterative solution of a linear system, exploiting the underlying network structure to avoid inverting any large matrices. Both of these methods, however, have undesirable aspects. In modern regional networks, planar bushes are rare because freeway interchanges and overpasses are modeled in detail, rather than representing the entire interchange with a single node (as was done in past decades to reduce the number of nodes and arcs). For example, in the Chicago regional network [Bar-Gera, 2013], none of the 1790 origin bushes are planar at equilibrium. While the second method is applicable in general networks, it requires careful implementation to avoid numerical instabilities due to repeated matrix reinversion.

Also, these previous methods assume that the travel time between the origin and destination nodes of each OD pair is a function of its own demand, and independent of other OD demands. In case of a congested network, this assumption may degrade the accuracy of the method. Due to overlapping paths, a small change to the flow on one path affects the travel time experienced by other paths. The set of impacted paths and OD pairs extends further when some travelers decide to reroute to other paths in response to travel time increase on their current path. In this chapter, the formulation is revised to model these interactions: the dependency of each OD travel time on the demand of other OD pairs. In fact, modeling OD travel time as a function of its demand can be viewed as a special case of the model proposed in this chapter, and further a heuristic is proposed to address the question of which OD pairs should be included in modeling the travel time of each OD pair.

In addition, this chapter describes a third method for calculating the necessary parameters, by reformulating the linear system of equations as the solution to a modified network equilibrium problem. This method is faster than the previous methods, makes no planarity assumptions, and can be solved by making only minor adjustments to existing equilibrium algorithms. In summary, the contributions are as follows:

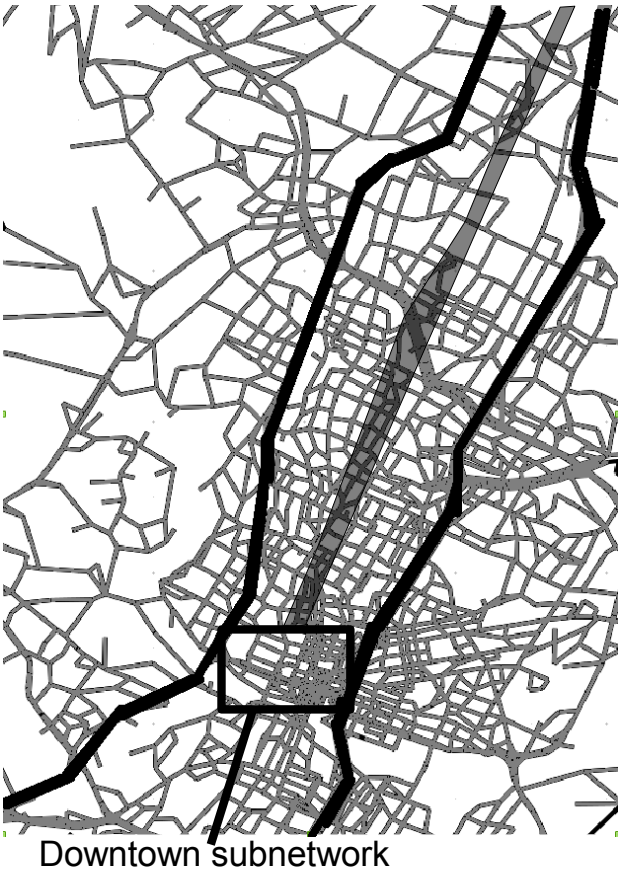


Figure 2.1: Portion of regional network with equidistant region shaded. Route choice from this region to downtown is sensitive to changes in the subnetwork regardless of distance between the origin and downtown.

- The contracted model is extended to include the interactions between different OD pairs.
- A symmetry result is found in the sensitivity of travel times and demands across different OD pairs: the impact of increased OD flow on another OD pair's cost is the same as the impact of increased flow of the second OD pair on the first OD pair's cost.
- A heuristic is proposed to approximate the most prominent interactions.
- A formulation similar to the network equilibrium problem is developed to carry out the sensitivity analysis and compute the derivatives.

The remainder of the chapter is organized as follows: Section 2.2 discusses related research on network contraction, sensitivity analysis, and bush-based algorithms. Section 2.3 briefly reviews the context and objectives of the problem at hand, and Section 2.4 presents the novel method based on reformulation as an equilibrium problem. Section 2.5 describes a heuristic for estimating the interactions. Section 2.6 demonstrates the accuracy and computational performance of this procedure using two case studies. The Barcelona network is used to measure the complexity and to evaluate the accuracy of the proposed model in response to different demand perturbations. The advantages of simplifying the regional network rather than deleting it in subnetwork analysis is illustrated on a regional network representing Austin, TX. Section 2.7 concludes the work.

2.2 Literature Review

The idea of simplifying networks to reduce computation time has existed for several decades, receiving considerable attention prior to the rise of efficient path-based and bush-based equilibrium algorithms since the 1990's. In case of an uncongested network, networks with constant link travel cost, the problem is easy to handle by using aggregated nodes instead of a group of the network nodes. Zipkin [1980] derives bounds for such a contraction technique for a class of linear minimum cost flow problems. As noted by Friesz [1985], the network contraction problem becomes complicated and computationally intensive for congested networks.

Different contraction methods have been studied in the presence of congestion. The early studies on network contraction used the idea of link extractions [Haghani and Daskin, 1983]. Link extraction refers to removing unimportant links and nodes from the network. Chan [1976] proposes such a network contraction approach for networks with constant link cost. This study shows that the derived traffic from the excluded links cause an unpredictable flow pattern through the network. Hearn [1984] proposes an

analytical method for link extraction where the subnetwork of interest is extracted and then a transfer decomposition is applied to partition the original traffic assignment problem into a master problem and a subproblem. The master problem deals with a modified version of the original network where the subnetwork is replaced with some pseudo-links, and the subproblem solves the traffic assignment problem over the extracted subnetwork.

Some efforts have been done to combine the link extraction with demand aggregation. Modeling the transportation network with full demand resolution requires detailed data which is expensive and time consuming. Zonal aggregation can reduce the computational expense of detailed zoning by dividing the space into discrete zones called traffic analysis zones. It is assumed that zone activity is concentrated at the zone centroid and demand are loaded on and of the network through connectors [Jafari et al., 2015]. By evaluating two levels of network details and 11 zoning structures, Chang et al. [2002] show that smaller zones are more beneficial in travel demand modeling, while larger zones result in a better performance for less detailed network. Jeon et al. [2010] construct three networks with different levels of contraction: a fine, a medium (collectors are excluded), and a coarse model (collectors and minor streets are deleted). Zones are also aggregated based on the method proposed by Bovy and Jansen [1983]. The simulation results show that the reduced capacity is hard to estimate, and traffic volumes are overestimated on the remaining links as a result of this reduced capacity. Also studies by Eash et al. [1983] and Sbayti et al. [2002] show that due to inconsistency between the extracted network and the complete network, the results from these methods are not reliable.

Another contraction approach is link abstraction where a set of links and nodes between two nodes are replaced with a single aggregated link. Eash et al. [1983] and Boyce et al. [1985] propose rules to aggregate series and parallel links for the purpose of the sketch planning. This idea is used by Wright et al. [2010] to reduce the computational time and increase the efficiency of the algorithm for large-scale networks. Connors and Watling [2014] employ the idea of link abstraction for stochastic user equilibrium (SUE) problem where the route choice is a function of the path utility. In SUE there is no notion of equilibrium travel time, and the authors propose the composite cost as a unique measure for each OD pair. Similar to Boyles [2012], sensitivity analysis is used to model the OD composite cost as function of the OD demands.

The methodology proposed in this chapter can be classified as link abstraction method based on the equilibrium sensitivity analysis and is a sequel to work done by Boyles [2012]. Explicit sensitivity analysis for the static equilibrium problem has primarily been based on the implicit function theorem [Tobin and Friesz, 1988, Cho et al., 2000, Yang and Bell, 2007a] or sensitivity results of variational inequalities [Lu, 2008]. Recently, Bar-Gera et al. [2013] formulated the sensitivity of link flows with respect to network design parameters as a solution to a quadratic programming and evaluated the precision of the computed

derivatives. Other researchers have established general regularity properties on the solution of network equilibrium problems as the input data vary, identifying conditions under which the equilibrium solution (expressed either in terms of link or path flows) is continuous or differentiable [Qiu and Magnanti, 1989, Yen, 1995, Patriksson, 2004, Lu and Nie, 2010a]. Particularly relevant to this problem, equilibrium bushes are stable and link flows are analytic with respect to demand perturbations under generally nonrestrictive regularity assumptions [Boyles, 2012].

In static traffic assignment, there is always an equilibrium flow solution in which the links used by each origin form an acyclic subgraph [Bar-Gera, 2002]. Due to the existence of a topological order [Ahuja et al., 1993], network algorithms on acyclic graphs are usually very fast. To the author's knowledge, this notion of a bush (an acyclic subgraph) dates to Dial [1970], where it was applied in a stochastic network loading procedure. The same concept was used in a network pricing problem [Dial, 1999a], and independently rediscovered in the setting of communications networks [Gallager, 1977], before being applied to the static equilibrium problem as the basis for a number of algorithms [Dial, 1999b, Bar-Gera, 2002, Dial, 2006, Nie, 2010, Gentile, 2014].

2.3 Problem Statement

Consider a directed network $G = (N, A, Z)$ with node and arc sets N and A of cardinality n and m, respectively, and a set of zones $Z \subseteq N$ of cardinality z. Let $W \subseteq Z \times Z$ denote the set of K OD pairs. The travel demand between OD pair w is indicated by d_w. We group the OD demands into a K-dimensional vector $\mathbf{d}$. Let p_w denote the set of all paths connecting OD pair w and $\mathbf{p}$ be the entire set of paths in the network. The delay on each link a is given by $t_a(x_a)$, as a function of its flow x_a which is strictly positive, strictly increasing, and differentiable. The deterministic user equilibrium problem seeks the vector of link flows $\hat{\mathbf{x}}$ minimizing the following convex optimization problem:

$$\text{minimize} \quad \sum_{a \in A} \int_0^{x_a} t_a(x) dx \tag{2.1}$$

$$\text{subject to} \quad \sum_{\pi \in p_w} h_\pi = d_w, \qquad \forall w \in W \tag{2.2}$$

$$\sum_{w \in W} \sum_{\pi \in p_w} h_\pi \delta_{a\pi} = x_a, \qquad \forall a \in A \tag{2.3}$$

$$h_\pi \geq 0, \qquad \pi \in \mathbf{p} \tag{2.4}$$

Table 2.1: Table of notation

N	$\triangleq$	set of network nodes of cardinality n
A	$\triangleq$	set of network links of cardinality m
Z	$\triangleq$	set of network zones of cardinality z
W	$\triangleq$	set of network OD pairs of cardinality K
$w \in W$	$\triangleq$	an OD pair
$\mathbf{d} = [d_w]$	$\triangleq$	vector of OD demands
$\mathbf{p} = [p_w]$	$\triangleq$	vector of OD paths
$\hat{p}_w$	$\triangleq$	set of paths with positive flow between OD pair w
T_w	$\triangleq$	travel time between OD pair w
h_π	$\triangleq$	flow of path π
C_π	$\triangleq$	cost of path π
x_a	$\triangleq$	flow on link a
$t_a(x_a)$	$\triangleq$	cost of traveling on link a
$\delta_{a\pi}$	$\triangleq$	1 if link a is on path π, otherwise 0
$\mathbb{B}_r$	$\triangleq$	equilibrium bush rooted at origin r
$\mathbb{N}_r$	$\triangleq$	set of nodes visited by some demand from origin r
$\mathbb{A}_r$	$\triangleq$	set of links visited by some demand from origin r
$\mathcal{B}_w$	$\triangleq$	stem of OD pair w
$\mathcal{N}_w$	$\triangleq$	nodes utilized by d_w
$\mathcal{A}_w$	$\triangleq$	links utilized by d_w
N_c	$\triangleq$	set of nodes in contracted network
A_c	$\triangleq$	set of artificial links in contracted network
Z_c	$\triangleq$	set of zones in contracted network
l_w	$\triangleq$	artificial link between OD pair w in contracted network
Υ_w	$\triangleq$	cost on artificial link between OD pair w
$G(w)$	$\triangleq$	set of OD pairs selected to model the travel time between origin and destination nodes of w
α_a^w	$\triangleq$	derivative of x_a with respect to d_w
β_π^w	$\triangleq$	derivative of h_π with respect to d_w
τ_u^w	$\triangleq$	derivative of T_u with respect to d_w
OD_a	$\triangleq$	set of OD pairs with demand on link a

where h_π is the flow on path π, and $\delta_{a\pi}$ is 1 if path π utilizes link a, and 0 otherwise. Assume that the unique link flow vector $\hat{\mathbf{x}}$ satisfying the above problem is known, and let $\hat{T}_w$ be the minimum travel time between OD pair w at equilibrium flow $\hat{\mathbf{x}}$. Furthermore, let $\hat{p}_w$ denote the set of paths with positive flow for w corresponding to $\hat{\mathbf{x}}$. These might represent the entropy-maximizing path flows, or some other path-based solution. The route flow is assumed to be strictly complementary in the sense that all minimum-cost routes have positive flow. This assumption is common in the sensitivity analysis literature [Friesz et al., 1990, Cho et al., 2000, Patriksson, 2004, Josefsson and Patriksson, 2007, Yang and Bell, 2007a]. In practice, it can be difficult to determine whether a solution violates strict complementarity, because equilibria are only solved to finite precision (cf. Bar-Gera [2006]), but noncomplementary solutions are a zero-measure set [Patriksson, 2004] so we feel this assumption is viable for practical applications such as the ones considered in this chapter.

The equilibrium flow from each origin to all destinations form a *bush*, a connected and acyclic subnetwork connecting origin to each node. Define the *equilibrium bush rooted at origin* r to be $\mathbb{B}_r = (\mathbb{N}_r, \mathbb{A}_r)$, where $\mathbb{N}_r \subseteq N$ and $\mathbb{A}_r \subseteq A$, respectively, are sets of nodes and links visited by some demand from r. For each OD pair $w \in W$ with origin r, *stem* $\mathcal{B}_w = (\mathcal{N}_w, \mathcal{A}_w)$ is defined as a subgraph of equilibrium bush $\mathbb{B}_r$ comprised of links and nodes utilized by d_w at equilibrium flow $\hat{\mathbf{x}}$; $\mathcal{A}_w$ and $\mathcal{N}_w$ are stem links and nodes, respectively. More precisely, the stem $\mathcal{B}_w$ is a subnetwork formed by the union of links and nodes visited by at least one path in $\hat{p}_w$.

We seek to estimate a simple equation to model the relation between the OD travel time and network flows. By doing so, network G is replaced by a "contracted" network $G_c = (N_c, A_c, Z_c)$ with the following properties: (a) the contracted network has the same set of zones as the original network and does not contain any intermediate node, i.e. $N_c = Z_c = Z$; (b) the number of links in the contracted network is the same as number of OD pairs in G. More precisely, there is one artificial link between each OD pair in the contracted network. If network G has n nodes, m links, z zones, and K OD pairs, the contracted version includes z nodes, z zones and K links. Figure 2.2 shows this transformation on a small network. The complete network on the left panel has 1 origin node, node 1, and 2 destination nodes, nodes 3 and 5, while the associated contracted network, right panel, has just 3 nodes and 2 links; one node for each origin/destination node and one artificial link between each OD pair.

The artificial links in the contracted network model the travel time between each OD pair as a function of OD flows. Let l_w denote the artificial link between OD pair w in the contracted network and Υ_w denote the travel time on l_w. As discussed before, Υ_w estimates the cost of travel between OD pair w in the complete network G. Boyles [2012] assumed that Υ_w is a function of d_w alone, and independent of other OD demands, i.e. $\Upsilon_w = \Upsilon_w(d_w)$. In this chapter, however, this assumption is relaxed, and the

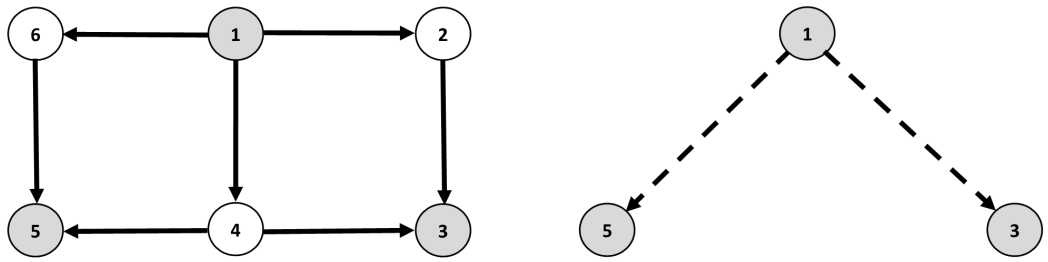


Figure 2.2: The complete network (left) has 6 nodes and 7 links, while the contracted network (right) has just 3 nodes representing the zones, and 2 artificial links representing each OD pair.

cost of travel on each artificial link is formulated as a multivariable function to capture the interactions between multiple OD flows.

Using $\hat{\mathbf{d}}$ to denote the demand vector at the equilibrium solution $\hat{\mathbf{x}}$, and $\tilde{\mathbf{d}}$ to denote the perturbed demand vector, we write Υ_w using the first-order Taylor expansion:

$$\Upsilon_w(\tilde{\mathbf{d}}) = \hat{T}_w + \langle \nabla \hat{T}_w, \tilde{\mathbf{d}} - \hat{\mathbf{d}} \rangle \tag{2.5}$$

where $\nabla \hat{T}_w$ is the K-dimensional gradient vector of $\hat{T}_w$ with respect to OD flows evaluated at $\hat{\mathbf{x}}$, and $\langle x, y \rangle$ is the inner product of vectors x and y. In (2.5), it is assumed that the travel time between OD pair w is a function of all demand vector entries. This assumption, however, requires estimating $\mathcal{O}(K^2)$ parameters which is not practically feasible. Hence we model the travel time between OD pair w as a function of a *subset* of OD demands. Let $G(w)$ (of cardinality g_w) denote the set of OD demands selected to model the travel time of OD pair w in the contracted network. Then, (2.5) can be written as:

$$\Upsilon_w(\tilde{\mathbf{e}}_\mathbf{w}) = \hat{T}_w + \langle \boldsymbol{\psi}_w, \tilde{\mathbf{e}}_w - \hat{\mathbf{e}}_w \rangle \tag{2.6}$$

where $\tilde{\mathbf{e}}_w$, $\hat{\mathbf{e}}_w$, and $\boldsymbol{\psi}_w$ are, respectively, g_w-dimensional subvectors of vectors $\tilde{\mathbf{d}}$, $\hat{\mathbf{d}}$, and $\nabla \hat{T}_w$ corresponding to OD pairs in $G(w)$. A small example in Section 2.6 describes this model approximation in more detail.

In general, g_w can be any number between 1 and K where $g_w = 1$ means that Υ_w is just a function of d_w (the assumption in Boyles [2012]), and for $g_w = K$, Υ_w is a function of all OD flows. The value of g_w is a trade-off between the accuracy of the contracted model and computational time required to estimate the contracted network, where a smaller g_w requires estimating fewer parameters, but such a model may

be prone to higher error rates since the effect of some OD demands are neglected. A heuristic for choosing this subset of OD pairs is provided below.

In equation (2.6), the only unknowns to be estimated for OD pair w are the g_w components of the vector $\boldsymbol{\psi}_w$. The entry of the vector $\boldsymbol{\psi}_w$ associated with OD pair u, i.e. $\partial \hat{T}_w / \partial d_u$, shows the derivative of $\hat{T}_w$ with respect to d_u evaluated at $\hat{\mathbf{x}}$. The theorem below shows that under the assumption that OD stems remain unchanged, these interactions are symmetric: a small change in $\hat{d}_u$ has the same impact on $\hat{T}_w$ as a small change in $\hat{d}_w$ would have on $\hat{T}_u$.

Theorem 1. *For any two OD pairs w and u:*

$$\frac{\partial \hat{T}_w}{\partial d_u} = \frac{\partial \hat{T}_u}{\partial d_w} \tag{2.7}$$

Proof. Tobin and Friesz [1988] showed that for a small demand perturbation, where the set of used paths remain intact, the change in equilibrium travel time would be:

$$\partial \mathbf{T} = \left[\Lambda (\nabla \mathbf{C})^{-1} \Lambda^t\right]^{-1} \partial \mathbf{d} \tag{2.8}$$

where $\partial \mathbf{T}$ and $\partial \mathbf{d}$ are, respectively, differential vectors of changes in OD travel time and demand, and $\nabla \mathbf{C}$ is Jacobian of path cost vector with respect to path flow:

$$\nabla \mathbf{C} = \begin{bmatrix} \frac{\partial C_1}{\partial h_1} & \frac{\partial C_1}{\partial h_2} & \cdots & \frac{\partial C_1}{\partial h_\rho} \\ \frac{\partial C_2}{\partial h_1} & \frac{\partial C_2}{\partial h_2} & \cdots & \frac{\partial C_2}{\partial h_\rho} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial C_\rho}{\partial h_1} & \frac{\partial C_\rho}{\partial h_2} & \cdots & \frac{\partial C_\rho}{\partial h_\rho} \end{bmatrix} \tag{2.9}$$

where ρ is the number of paths with positive demand, and C_π is travel time on path π. Λ is the OD/path incidence matrix with K rows and ρ columns and $\Lambda(w, \pi) = 1$ if path π is used by d_w, and 0 otherwise.

Noting that $\frac{\partial x_a}{\partial h_\pi} = 1$ if link a is utilized by path π, $\delta_{a\pi} = 1$, for two paths π and η we have:

$$\frac{\partial C_\pi}{\partial h_\eta} = \begin{cases} \sum_{a \in A} \dfrac{\partial t_a}{\partial x_a} \delta_{a\pi} & \text{if } \pi = \eta \\ \sum_{a \in A} \dfrac{\partial t_a}{\partial x_a} \delta_{a\pi} \delta_{a\eta} & \text{otherwise} \end{cases}$$

Using this equality, we can see that for two different paths π and η with some common links:

$$\frac{\partial C_\pi}{\partial h_\eta} = \frac{\partial C_\eta}{\partial h_\pi} \tag{2.10}$$

indicating that $\nabla \mathbf{C}$ (and thus its inverse) are symmetric:

$$(\nabla \mathbf{C})^{-1} = \left((\nabla \mathbf{C})^{-1}\right)^t \tag{2.11}$$

We can thus show that $\Lambda(\nabla \mathbf{C})^{-1}\Lambda^t$ is symmetric:

$$\left(\Lambda(\nabla \mathbf{C})^{-1}\Lambda^t\right)^t = \Lambda\left(\Lambda(\nabla \mathbf{C})^{-1}\right)^t = \Lambda\left((\nabla \mathbf{C})^{-1}\right)^t \Lambda^t = \Lambda(\nabla \mathbf{C})^{-1}\Lambda^t \tag{2.12}$$

which implies that $\left(\Lambda(\nabla \mathbf{C})^{-1}\Lambda^t\right)^{-1}$ is also symmetric. Let

$$\left(\Lambda(\nabla \mathbf{C})^{-1}\Lambda^t\right)^{-1} = \begin{bmatrix} \gamma_{11} & \gamma_{12} & \cdots & \gamma_{1K} \\ \gamma_{21} & \gamma_{22} & \cdots & \gamma_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{K1} & \gamma_{K2} & \cdots & \gamma_{KK} \end{bmatrix} \tag{2.13}$$

then

$$\begin{bmatrix} \partial \hat{T}_1 \\ \partial \hat{T}_2 \\ \vdots \\ \partial \hat{T}_K \end{bmatrix} = \begin{bmatrix} \gamma_{11} & \gamma_{12} & \cdots & \gamma_{1K} \\ \gamma_{21} & \gamma_{22} & \cdots & \gamma_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{K1} & \gamma_{K2} & \cdots & \gamma_{KK} \end{bmatrix} \begin{bmatrix} \partial d_1 \\ \partial d_2 \\ \vdots \\ \partial d_K \end{bmatrix} \tag{2.14}$$

It can be verified easily that change in equilibrium cost of OD pair w in response to a small change in demand of OD pair u is:

$$\frac{\partial \hat{T}_w}{\partial d_u} = \gamma_{wu} \tag{2.15}$$

and the same way:

$$\frac{\partial \hat{T}_u}{\partial d_w} = \gamma_{uw} \tag{2.16}$$

these two values are equal. □

Later, in Section 2.6, a numerical example on the small network described in Figure 2.2 clarifies this property and shows how to compute these derivatives and interactions.

This symmetry property can reduce the computational time required to estimate the contracted network by calculating just one derivative instead of doing two sensitivity analysis to compute $\partial \hat{T}_u / \partial d_w$ and $\partial \hat{T}_w / \partial d_u$. Consider a network with K OD pairs and assume that we want to model the contracted network with full interactions. Furthermore, assume that all K OD stems are of the same size. Without considering the symmetry property, we need to solve K problems each with K OD stems, a total of K^2 problems. With the symmetricity property, the problem size associated with the first OD pair is K, the second problem is $K-1$, and the last OD pair is 1. The happens because after solving the sensitivity problem for OD pair w_1 and computing the derivative for OD pair w_2, $\partial \hat{T}_{w_2} / \partial d_{w_1}$, we do not need to include $\mathcal{B}_{w_1}$ when doing sensitivity analysis for OD pair w_2. This results in a total of $\frac{K(K-1)}{2}$ problems, roughly half the number needed without considering the symmetricity property.

From here on, the focus of the chapter is the computation of these partial derivatives.

Assume that demand between OD pair w is perturbed by a small value such that the set of equilibrium paths remain fixed, and the goal is to estimate the change in the travel time of OD pairs in $G(w)$. Under the assumption that OD stems remain unchanged, flow shifts between the stem paths of each OD pair $u \in G(w)$ until a new equilibrium is achieved and all the stem paths have the same cost. This indicates that travel time is changed by the same amount on all paths. The above condition can be stated as:

$$\frac{\partial \hat{T}_w}{\partial d_u} = \frac{\partial \hat{T}_u}{\partial d_w} = \frac{\partial C_\pi}{\partial d_w}, \qquad \forall \pi \in \hat{p}_u, u \in G(w) \tag{2.17}$$

The changes in each path's travel time is equal to the sum of the changes in travel times of its links:

$$\frac{\partial C_\pi}{\partial d_w} = \sum_{a \in A_w} \frac{\partial t_a}{\partial d_w} \delta_{a\pi}, \qquad \forall \pi \in \hat{p}_u, u \in G(w) \tag{2.18}$$

where $\partial t_a / \partial d_w$ is the derivative of link a travel time with respect to d_w. Using the chain rule, the above

equation can be written as:

$$\frac{\partial C_\pi}{\partial d_w} = \sum_{a \in A_w} t'_a \alpha_a^w \delta_{a\pi}, \qquad \forall \pi \in \hat{p}_u, u \in G(w) \tag{2.19}$$

where $t'_a = dt_a/dx_a$ is the derivative of link travel time with respect to the link flow evaluated at $\hat{\mathbf{x}}$, and $\alpha_a^w = \partial x_a / \partial d_w$ is the derivative of link a flow with respect to d_w evaluated at $\hat{\mathbf{x}}$.

Let $\beta_\pi^w = \partial h_\pi / \partial d_w$ be the derivative of path π flow with respect to d_w and $\bar{\mathbf{A}}_w = \bigcup_{u \in G(w)} A_u$ be the union of the links on the stems of OD pairs in $G(w)$. Then we have:

$$\alpha_a^w = \sum_{u \in G(w)} \sum_{\pi \in \hat{p}_u} \beta_\pi^w \delta_{a\pi}, \qquad \forall a \in \bar{\mathbf{A}}_w \tag{2.20}$$

In addition, since d_w is the only independent variable, we can write:

$$\sum_{\pi \in \hat{p}_u} \beta_\pi^w = \begin{cases} 1 & w = u \\ 0 & w \neq u \end{cases}, \qquad \forall u \in G(w) \tag{2.21}$$

Equation (2.21) indicates that one unit of demand is added to d_w and the demand between the remaining OD pairs remain unchanged. The linear system of equations described by (2.17), (2.19), (2.20), and (2.21) can be solved for β_π^w and compute $\partial \hat{T}_u / \partial d_w$ for every OD pair $u \in G(w)$ which form the entries of $\boldsymbol{\psi}_w$. This procedure needs to be repeated for each OD pair $w \in W$.

The linear system (2.17), (2.19), (2.20), and (2.21) form the basis for the sensitivity analysis in this chapter. Boyles [2012] describes two methods for calculating these derivatives. The first approach is based on network transformations and has similarities with techniques used in analysis of resistive circuits. Four network transformations are proposed: two series links to one link; two parallel link to one link; $\boldsymbol{\Delta}$ structure (an undirected cycle of three nodes with an empty interior) to $\mathbf{Y}$, and $\mathbf{Y}$ structure to $\boldsymbol{\Delta}$. These four transformations are illustrated in Figure 2.3. Under the planarity assumption, these transformations can be used to successively replace each OD stem with a simpler stem. At the end, the stem is reduced to a single arc, and OD derivative can be easily calculated.

The second technique calculates these derivatives by directly solving equations (2.17), (2.19), (2.20), and (2.21) enforcing that the stem remains at equilibrium. The system of linear equation has $\mathcal{O}(n+m)$ equations and requires inverting a matrix of size $\mathcal{O}(n+m)$. Using a well-known block inversion formula, Boyles [2012] proposes a technique that iteratively solves this linear system of equations, ensuring that no

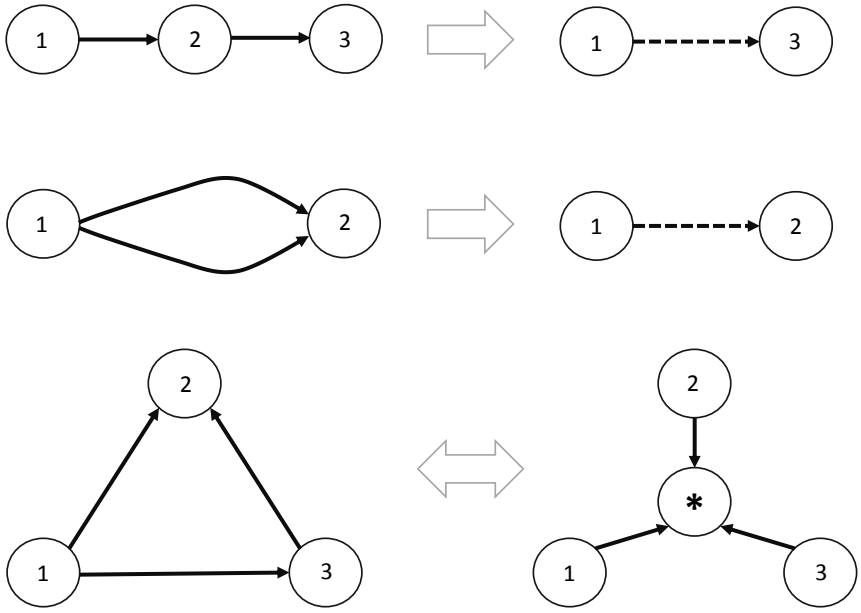


Figure 2.3: Schematic of series (top), parallel (middle), and $\mathbf{\Delta}$-$\mathbf{Y}$ (bottom) transformations

matrix needs to be inverted with dimension larger than the number of incident arcs to a single node.

In summary, the first technique is limited to planar bushes, which is a restrictive assumption. The second technique is more general and does not require any planarity assumption, but has higher computational time and needs careful implementation to avoid numerical instabilities from repeated matrix reinversion. In the next section, however, we propose a faster solution method based on formulating this system as the solution to a convex program.

Rather than solving (2.17), (2.19), (2.20), and (2.21) directly as a linear system, in this section we show that the solution to this system also solves the convex optimization problem:

$$\text{minimize} \sum_{a \in \bar{\mathbf{A}}_w} \int_0^{\alpha_a^w} t'_a \omega \, d\omega \tag{2.22}$$

$$\text{subject to} \sum_{\pi \in \hat{p}_u} \beta_\pi^w = 0, \qquad \forall u \in G(w), u \neq w \tag{2.23}$$

$$\sum_{\pi \in \hat{p}_w} \beta_\pi^w = 1 \tag{2.24}$$

$$\alpha_a^w = \sum_{u \in G(w)} \sum_{\pi \in \hat{p}_u} \beta_\pi^w, \qquad \forall a \in \bar{\mathbf{A}}_w \tag{2.25}$$

This is essentially a static equilibrium problem on the network comprised of all OD stems in $G(w)$, with α_a^w and β_π^w serving the role of link flows and path flows, respectively, linear cost functions of the form $t'_a \alpha_a^u$, and unit demand between OD pair u. There is one significant difference: there is no nonnegativity constraint on the β_π^u, reflecting the fact that not all path flows need increase with d_w. The Braess network [Braess, 1969] forms a counterexample: as the reader can verify, when the demand between the endpoints increases from 6, flow on the path utilizing the middle link *decreases*.

The objective function (2.22) is strictly convex, and the constraints (2.23)–(2.25) form a convex set, so a solution exists and is unique. Lagrangianizing the flow conservation constraints (2.23) and (2.24) with multipliers $\boldsymbol{\tau}^w$, the first-order conditions are

$$\sum_{a \in A_u} t'_a \alpha_a^w \delta_{a\pi} - \tau_u^w = 0, \qquad \forall \pi \in \hat{p}_u, u \in G(w) \tag{2.26}$$

$$\sum_{\pi \in \hat{p}_u} \beta_\pi^w = 0, \qquad \forall u \in G(w), u \neq w \tag{2.27}$$

$$\sum_{\pi \in \hat{p}_w} \beta_\pi^w = 1 \tag{2.28}$$

$$\sum_{u \in G(w)} \sum_{\pi \in \hat{p}_u} \beta_\pi^w = \alpha_a^w, \qquad \forall a \in \bar{\mathbf{A}}_w \tag{2.29}$$

with no complementarity conditions because there are no nonnegativity constraints on $\boldsymbol{\beta}$. Interpreting τ_u^w as $\partial \hat{T}_u / \partial d_w$, the system (2.26)–(2.29) is identical to (2.17), (2.19), (2.20), and (2.21).

This equivalence presents an easier method to identify the travel time derivatives needed to form cost functions on the artificial arcs of the contracted network, because the optimization problem (2.22)–(2.25) can be solved by making minor changes to bush-based algorithms for solving the traffic assignment problem.

The bush-based algorithms start from the shortest path tree for each origin and iterate between equilibrating and modifying the bush. At each iteration, the origin bush is fixed and the network is reduced to an acyclic network containing just the bush links and the assignment problem is restricted to this acyclic network. Bar-Gera [2002] and Dial [2006] prove that the equilibrium flow is attained after all bushes are equilibrated. The general procedure may be described as follows:

1. Initialize the origin bush by finding the shortest path tree for each origin and assign all flows to the origin bush.
2. Add new links to bush if travel time can be improved.
3. Solve the assignment problem over the acyclic network restricted to the bush.
4. Remove links with zero flow.
5. Stop if converged, otherwise go to step 2.

The interested reader is referred to Nie [2010] for an in-depth discussion on different bush-based algorithms and their computational power.

In a bush-based algorithm, the only change needed is to eliminate the zero-flow lower bound when equilibrating the bush, and skipping the bush updating steps since these links are fixed. Further, by solving this equilibrium problem to varying gap levels, one can more finely adapt the precision of the method to the computational resources available. As shown in the next section's computational results, it is not necessary to calculate these derivatives with a high level of accuracy, because other aspects of the approximation (e.g., assuming fixed bushes) tend to dominate the error in the overall procedure.

The optimization problem (2.22)–(2.25) is a special case of the quadratic approximations developed by Patriksson [2004] and Josefsson and Patriksson [2007], for the case where only the OD matrix is perturbed, link costs are separable, and there is no elasticity in demand. Patriksson [2004] obtained the model starting from results on the sensitivity of variational inequalities. This chapter presents an alternative derivation starting from equilibrium and flow conservation principles, showing that these equations form the KKT conditions for the quadratic approximation model given in the chapter. We believe that there is value in presenting such an alternative derivation, and that it may be more intuitive for some readers (albeit

for a special case of Patriksson's results.) Also the problem (2.22)–(2.25) is disaggregated, only considering the interactions between partial sets of OD pairs. This allows the sensitivity subproblem to be separated by OD pair, which may have computational advantages (including easier parallelization).

In this section, a formulation similar to classical user equilibrium problem was proposed to compute the sensitivity of travel time for every OD pair $u \in G(w)$ in response to a small change in d_w. The question still remained to be answered is which OD pairs should be included in $G(w), \forall w \in W$. The next section will address this question by proposing a simple heuristic.

2.5 Network Interactions

It is important to consider how the list of OD flows for modeling the cost on l_w in (2.6) is chosen for every $w \in W$. The goal of embedding the other flows in Υ_w is achieving an approximation with lower estimation error. Based on this, the list $G(w)$ for OD pair $w \in W$ should contain those OD pairs for which a small change in their flow has the highest impact on travel cost of w. Estimating the dependency of each OD travel time on other OD flows, however, requires doing sensitivity analysis for all OD pairs which is a computationally challenging and practically infeasible task. To alleviate this problem and still gain a good approximation, a simple heuristic is proposed here.

For each OD pair, the links with the highest sensitivity — defined as derivative of link travel time with respect to its flow — play the role of bottlenecks. For example, a link with derivative of 1 sec/vehicle has a small impact on route choice of users compared to a link with derivative of 10 sec/vehicle. It is clear that a small change in flow of such bottleneck links may encourage some travelers to search for cheaper routes. This is related to the concept of intrinsic sensitivity, defined in Boyles et al. [2011].

Let OD_a denote the list of OD pairs with demand on link a. Given g_w for every OD pair $w \in W$, Algorithm 1 calculates the list $G(w)$ in the following way. First, the link travel time derivatives for all links with positive demand are calculated. Then all OD stems are checked and w is added to the list OD_a if link a is part of the stem of OD pair w. These two steps are preprocessing steps, and can be implemented without any further computational effort while solving for the UE. Then for each OD pair $w \in W$, the first bottleneck link in A_w, the link with the highest sensitivity, is computed and the most dominant OD pair, defined as the OD pair with the highest share of flow on this bottleneck link, is added to $G(w)$. To make sure that the evaluated bottleneck link and selected OD pair are not considered again, they are removed from the associated lists. If the number of OD pairs inserted in $G(w)$ is less than g_w, the first bottleneck link among the present links in A_w is selected and the same process is repeated by selecting the dominant OD pair and adding it to $G(w)$. This process stops when g_w OD pairs are selected for OD

pair w or all links are evaluated where in the latter case, the cardinality of $G(w)$ can be less than g_w. The algorithm can easily be modified to make sure that the list $G(w)$ contains g_w OD pairs by selecting more than one OD pair from each bottleneck link. This process is repeated until all OD pairs $w \in W$ are evaluated. Note that g_w's are the input parameters in this heuristic, and can vary by OD pair.

The next section is devoted to evaluate the performance and computational time of the approximate model under different demand and modeling scenarios. Later the advantage of the proposed network contraction technique is described in subnetwork analysis where areas outside the subnetwork boundary are replaced by some artificial links to capture the impact of subnetwork modifications on demand in areas beyond the subnetwork boundary.

2.6 Demonstration

In this section, three test networks are used to implement the idea presented in this chapter. The first case study is the toy network sketched in Figure 2.2. This network can be solved manually and helps the reader to follow the discussion of the chapter. Then, the Barcelona network is used to illustrate the quality of the contracted network in capturing the behavior of the complete network. Also the effect of g_w on the accuracy of the model and time required for estimating the unknown parameters of the contracted network is evaluated. Finally, the regional Austin, Texas network is used for the purpose of subnetwork analysis. All experiments are implemented in Java and carried out on a PC with an Intel Core i7 1.8GHz CPU and main memory of 4GB.

2.6.1 Toy Network

Consider the toy network in Figure 2.2 with link cost functions described in Table 2.2. Assume demand of d_{15} and d_{13} from origin node 1 to destination nodes 5 and 3, respectively. At equilibrium, the path flows

Algorithm 1 Calculate Interactions

```
for all a in A do:                                   ▷ calculates derivative of each link a ∈ A
    t'_a = dt_a/dx_a |_{x̂_a}

for all w in W do:                                   ▷ calculates the list of OD pairs utilizing each link a ∈ A
    for all a in A_w do:
        add w to OD_a

for all w in W do:                                   ▷ finds the list of g_w OD pairs with the highest interaction ∀w ∈ W
    okay:=false
    while okay is false do:
        a :=getBottleneckLink(w)
        delete a from A_w
        u :=getOD(a)
        delete u from OD_a
        add u to G(w)
        if |G(w)| = g_w or A_w is empty then:
            okay:=true

function getBottleneckLink(w)                        ▷ returns the link with the highest derivative value for w
    d:=0
    for all a in A_w do:
        if t'_a > d then:
            d := t'_a
            selectedLink:=a
    return selectedLink

function getOD(a)                                    ▷ returns the OD pair with the highest share of demand on link a
    d := 0
    for all w in OD_a do:
        if d_w > d then:
            d := d_w
            selectedOD:=w
    return selectedOD
```

Table 2.2: The link cost functions of the network in Figure 2.2.

Link	$1-2$	$1-6$	$4-3$	$4-5$	$6-5$	$1-4$	$2-3$
Cost function	2	2	2	2	$3x_{15}$	x_{14}	$2x_{23}$

and OD travel times are as follows:

$$\begin{cases} \hat{h}_{1,2,3} = \frac{3}{11}(d_{13} + d_{15}), \\ \hat{h}_{1,4,3} = \frac{1}{11}(8d_{13} - 3d_{15}), \\ \hat{T}_{13} = 2 + \frac{6}{11}(d_{13} + d_{15}) \\ \hline \hat{h}_{1,6,5} = \frac{2}{11}(d_{15} + d_{13}), \\ \hat{h}_{1,4,5} = \frac{1}{11}(9d_{15} - 2d_{13}) \\ \hat{T}_{15} = 2 + \frac{6}{11}(d_{15} + d_{13}) \end{cases} \tag{2.30}$$

Let $\tilde{d}_{13}$ and $\tilde{d}_{15}$ denote the perturbed demand values and $\hat{d}_{13}$ and $\hat{d}_{15}$ denote the current demands. According to equation (2.6), the cost of traveling on the artificial links in the contracted network, Figure 2.2 right panel, would be:

$$\begin{cases} \Upsilon_{13}(\tilde{d}_{13}, \tilde{d}_{15}) & = \hat{T}_{13} + \frac{\partial T_{13}}{\partial d_{13}}(\tilde{d}_{13} - \hat{d}_{13}) + \frac{\partial T_{13}}{\partial d_{15}}(\tilde{d}_{15} - \hat{d}_{15}) \\ & = 2 + \frac{6}{11}(\hat{d}_{13} + \hat{d}_{15}) + \frac{6}{11}(\tilde{d}_{13} - \hat{d}_{13}) + \frac{6}{11}(\tilde{d}_{15} - \hat{d}_{15}) \\ & = 2 + \frac{6}{11}(\tilde{d}_{13} + \tilde{d}_{15}) \\ \hline \Upsilon_{15}(\tilde{d}_{13}, \tilde{d}_{15}) & = \hat{T}_{15} + \frac{\partial T_{15}}{\partial d_{15}}(\tilde{d}_{15} - \hat{d}_{15}) + \frac{\partial T_{15}}{\partial d_{13}}(\tilde{d}_{13} - \hat{d}_{13}) \\ & = 2 + \frac{6}{11}(\hat{d}_{15} + \hat{d}_{13}) + \frac{6}{11}(\tilde{d}_{15} - \hat{d}_{15}) + \frac{6}{11}(\tilde{d}_{13} - \hat{d}_{13}) \\ & = 2 + \frac{6}{11}(\tilde{d}_{15} + \tilde{d}_{13}) \end{cases} \tag{2.31}$$

where Υ_{13} and Υ_{15} are approximate OD travel times formulated as a linear function of perturbed demand values $\tilde{d}_{13}$ and $\tilde{d}_{15}$. Since all link cost functions are linear, the linear approximates match the correct equations. Let $\Delta T = \tilde{T} - \hat{T}$ and $\Delta h = \tilde{h} - \hat{h}$ denote, respectively, the travel time and path flow deviations as a result of demand deviation $\Delta d = \tilde{d} - \hat{d}$. Then, relations between travel time, demand, and path flow

deviations are as follows:

$$\begin{cases} \Delta T_{13} = \frac{6}{11}(\Delta d_{13} + \Delta d_{15}) \\ \Delta h_{1,2,3} = \frac{3}{11}(\Delta d_{13} + \Delta d_{15}) = \frac{\Delta T_{13}}{2} \\ \Delta h_{1,4,3} = \frac{1}{11}(8\Delta d_{13} - 3\Delta d_{15}) = \Delta d_{13} - \frac{\Delta T_{13}}{2} \\ \hline \Delta T_{15} = \frac{6}{11}(\Delta d_{13} + \Delta d_{15}) \\ \Delta h_{1,6,5} = \frac{2}{11}(\Delta d_{13} + \Delta d_{15}) = \frac{\Delta T_{15}}{3} \\ \Delta h_{1,4,5} = \frac{1}{11}(-2\Delta d_{13} + 9\Delta d_{15}) = \Delta d_{15} - \frac{\Delta T_{15}}{3} \end{cases} \tag{2.32}$$

These equations can be used easily to compute the amount of change in demand for each OD pair that results in a change ΔT in equilibrium travel times. For example, if the demand from 1 to 3 remains constant, i.e. $\Delta d_{13} = 0$, then $\Delta h_{1,4,3} = -\Delta T/2$; $\Delta h_{1,4,5} = 3\Delta T/2$; and $\Delta d_{1,5} = 11\Delta T/6$. The same way, if the demand from 1 to 5 remains constant, i.e. $\Delta d_{15} = 0$, then $\Delta h_{1,4,5} = -\Delta T/3$; $\Delta h_{1,4,3} = 4\Delta T/3$; and $\Delta d_{1,3} = 11\Delta T/6$.

Note that the equations in (2.31) are estimated with full interactions, and the OD travel times where interactions are not modeled would be as follows:

$$\begin{cases} \Upsilon_{13}(\tilde{d}_{13}) = 2 + \frac{2}{3}\tilde{d}_{13} \\ \hline \Upsilon_{15}(\tilde{d}_{15}) = 2 + \frac{3}{4}\tilde{d}_{15} \end{cases} \tag{2.33}$$

which is not as accurate as the case with full interactions. The travel time derivatives in equations (2.31) and (2.33) can easily be computed by solving the optimization problem (2.22)–(2.25).

2.6.2 Barcelona

The method presented in the previous section is applied to Barcelona network with 110 zone, 1020 nodes, and 2522 links [Bar-Gera, 2013]. As stated before, the travel cost between each artificial link l_w in the contracted network is a linear approximation of the travel cost between OD pair w in the complete network.

The complete network is solved to relative gap of 10^{-6} defined as:

$$\text{relative gap} = \frac{\sum_{w \in W} \sum_{\pi \in \hat{p}_w} h_\pi C_\pi - \sum_{w \in W} \sum_{\pi \in \hat{p}_w} h_\pi \kappa^w}{\sum_{w \in W} \sum_{\pi \in \hat{p}_w} h_\pi C_\pi} \tag{2.34}$$

where κ^w represents the time spent on the fastest path between OD pair w. As verified by Boyce et al. [2004], this relative gap is enough to ensure that traffic assignment is converged to a stable link flow solution.

The equilibrium formulation (2.22)–(2.25) is used to compute the OD travel time derivatives with respect to OD demands. The equilibrium travel times along with these derivatives are needed to set up the contracted network. To evaluate the accuracy of the proposed network approximation algorithm, the OD matrix is perturbed, and the new equilibrium travel times are compared against those estimated via (2.6). For convenience, it is assumed that all OD pairs are modeled with the same number of OD interactions, i.e. $g_w = g$. Figure 2.4 shows the simulation results for different g values and demand scenarios. Each line shows one demand scenario where $p\%$ means that all OD demands are perturbed randomly by p percentage from the base values. The horizontal axis represents the number of OD flows selected to model the travel cost of each OD pair, g, and the vertical axis shows the average error between the actual travel time in the complete network and estimated travel time calculated from the contracted network. The estimation error is this figure is calculated as:

$$\epsilon = \frac{1}{K} \sum_{w \in W} \frac{\left| \Upsilon_w(\tilde{\mathbf{e}}) - \hat{T}_w(\tilde{\mathbf{d}}) \right|}{\hat{T}_w(\tilde{\mathbf{d}})} \tag{2.35}$$

where ϵ is the average estimation error per OD pair, $\tilde{\mathbf{d}}$ is the perturbed demand vector, and $\tilde{\mathbf{e}}$ is a subset of $\tilde{\mathbf{d}}$ corresponding to the entries of $G(w)$. $\Upsilon_w(\tilde{\mathbf{e}})$ is the estimated travel time between OD pair w using the contracted network for perturbed demand $\tilde{\mathbf{d}}$, and $\hat{T}_w(\tilde{\mathbf{d}})$ is the actual travel time obtained by solving the complete network for $\tilde{\mathbf{d}}$.

As expected, the contracted network yields better results for demand scenarios with lower perturbation, and larger demand perturbations produces higher errors. This is due to the fact that the approximate model is a first order Taylor series calibrated for base demand and as demand deviates more from the calibration point, the accuracy of the model deteriorates. Also under each demand scenario, including more OD flows to model the travel time of each OD pair provides more accurate travel time estimates where improvement is more significant for a larger perturbation. For example, under 50% perturbation scenario, the error of 6.7% for the case where OD travel time is modeled as a function of only its own demand drops to 3.6% when 200 other OD flows are also included to model the interactions; an improvement of 3.1%. The difference for perturbation of 5%, however, is less than 0.4%. This indicates that for a larger demand disruption, the impact of g is more significant, but nevertheless the improvement is marginal, especially for small demand perturbations, and diminishes after $g \simeq 10$. Based on these observations, modeling OD

travel times based on their own demand should be enough for most applications.

Figure 2.5 shows the time required to approximate and set up the contracted network. The horizontal axis represents dimension of gradient vector in (2.6), g, and vertical axis shows the ratio of time required to set up the contracted network —computing the list of OD flows for modeling each OD pair based on Algorithm 1 and solving the convex program (2.22)–(2.25) for each OD pair— to the time required for solving the complete network. It can be seen that computational complexity of the problem is affected by g, because it takes more time to set up the contracted network for a larger g value. This is mainly because modeling more interactions increases the size of the problem in (2.22)–(2.25): for $g = 1$, we need to solve these system of equations for one stem, while modeling all interactions requires solving a problem with size of the complete network for each OD pair. The results described in Figure 2.5 show that in case of Barcelona network with $g \leq 150$, it takes less time to set up the contracted network than solving the complete network.

The next section illustrates the benefits of the proposed network contraction technique for subnetwork analysis where the network outside the subnetwork boundary is replaced with an aggregated version to reduce the computational time and still capture the important attraction and diversion effects as drivers (globally) change routes in response to (local) subnetwork changes.

2.6.3 Subnetwork Application

To enable direct comparison with the methods developed by Boyles [2012], the same experimental setting was adopted. The regional network represents the Austin, TX metropolitan area, and contains 7466 nodes; 18,718 links; and 1117 zones; the downtown subnetwork contains 143 nodes, 448 arcs, and 20 internal zones (Figure 2.6). In the original network, 7th Street is a downtown arterial which is one-way eastbound.

We consider the impact of converting 7th Street to two ways, dividing its capacity equally between the two directions, and compare three techniques for evaluating the performance of the modified network. The first technique is to simply re-solve the equilibrium problem on the entire regional network; as reported in Boyles [2012], this requires approximately 20 minutes to reduce the relative gap to the range 10^{-4}–10^{-6}, according to current recommendations Boyce et al. [2004]. The second technique is to solve the equilibrium problem on the subnetwork alone, using the route flow solution from the base case network to form the OD matrix between boundary nodes; this requires only a second or two of computation time. The third technique is based on network contraction for $g_w = 1$ by first creating artificial arcs as discussed in Section 2.3, then eliminating artificial arcs directly connecting an origin to a destination by introducing elastic demand and inverting the Gartner transformation [Boyles, 2012]. Since the equilibrium

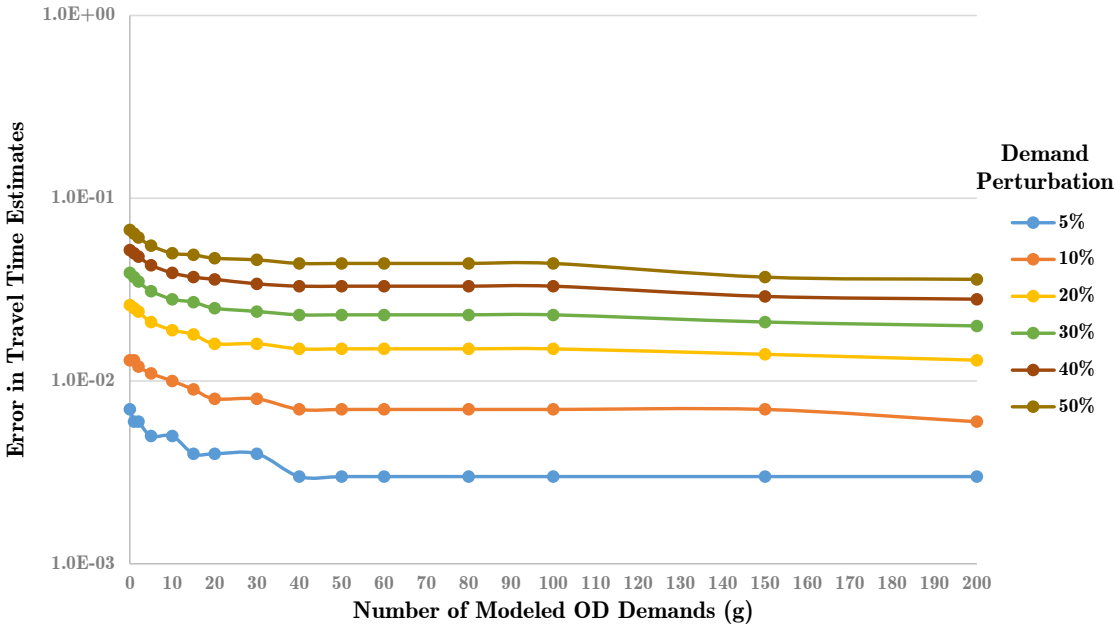


Figure 2.4: Total error between the travel times of the complete network and contracted network under different demand perturbation scenarios.

bushes are nonplanar, the artificial arc cost functions can be found either by solving linear system of the form (2.17), (2.19), (2.20), and (2.21) (which requires approximately 90 minutes), or by solving modified equilibrium problem of the form (2.22)–(2.25). Either method gives identical results when the equilibrium problems are solved to a very tight gap, so the focus on this section is comparing the computational requirements involved, and the tradeoffs between accuracy and computational time.

The remainder of this section investigates three main questions: first, the computation time required to solve (2.22)–(2.25) to varying gap levels; second, how the relative gap used in the the equilibrium problems in the third technique affects the accuracy of the approximation; and third, how the relative gap used in the elastic demand subnetwork problem in the third technique affects the accuracy of the approximation. This accuracy is measured in terms of link volumes, link travel times, and corridor travel times in each direction.

Each equilibrium problem (2.22)–(2.25) can be solved very rapidly, in far less than a second. However, creating the contracted network requires solving 212,432 such problems. Figure 2.7 shows how the total computation time required varies with the relative gap criterion used to terminate the equilibrium problems. Solving the equilibrium subproblems to a relative gap of 10^{-4} requires roughly an order of

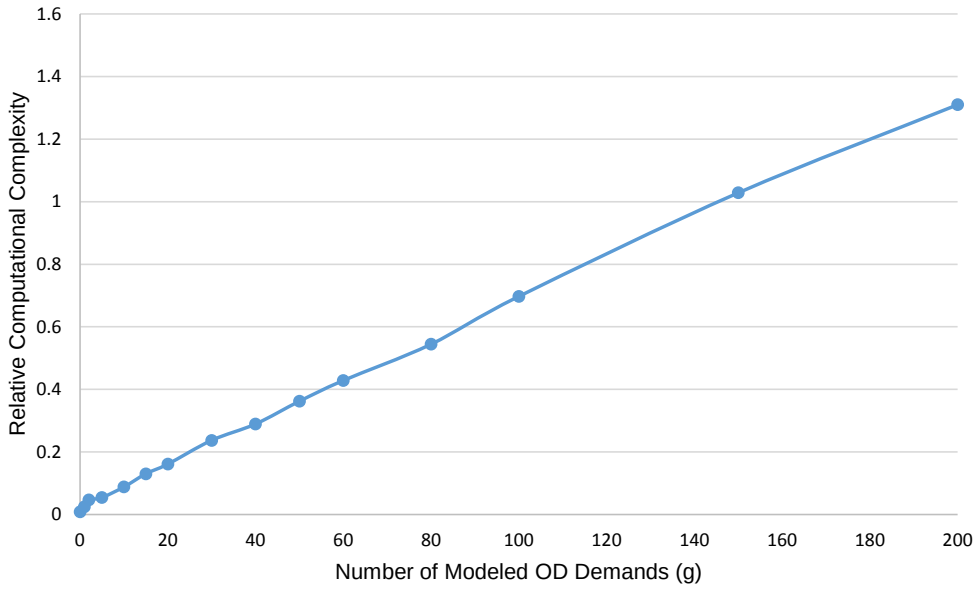


Figure 2.5: Ratio of time required to set up the contracted network to time of solving the complete network

magnitude less time than solving the linear systems directly. One may question the usefulness of this procedure, since the full regional network can be solved in only a little more time than is required to construct the approximation. The advantage comes in network design or other problems where subnetwork equilibrium must be solved many times, since the overhead involved in calculating these derivatives is only incurred once.

Table 2.3 shows how the root-mean square error (RMSE) link volume varies according to the relative gap used to calculate travel time derivatives, and the relative gap used when solving the elastic-demand equilibrium problem on the contracted network. These RMSE errors are calculated relative to the equilibrium solution on the regional network which serves as a baseline. By comparison, the RMSE from the fixed-boundary flows approach (the second technique) is 1216 vph. Notice that the accuracy of the third technique is not particularly sensitive to the gap values used. This indicates that the primary source of error in the approximation is due to the central assumptions made (fixed bushes and fixed flows outside each stem), rather than the accuracy to which the subproblems are solved. This result favors the method developed in this chapter, because there is no need to calculate the travel time derivatives exactly. Table 2.4 shows similar results when comparing average corridor travel times (in both directions, the contracted

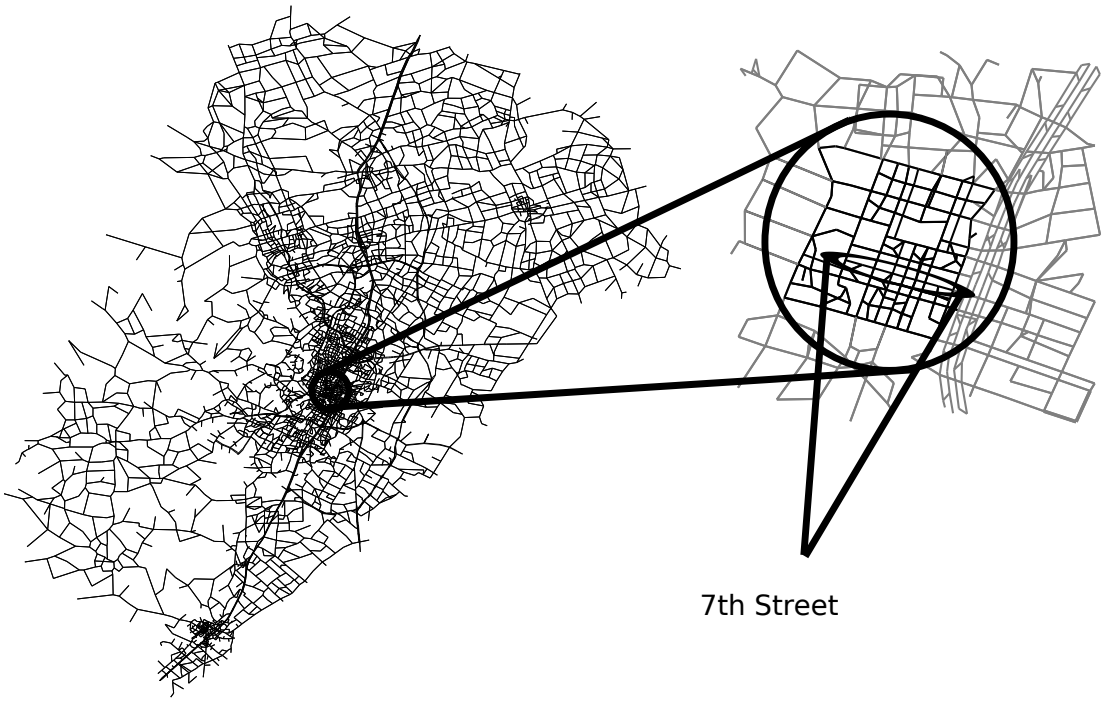


Figure 2.6: Austin regional network, subnetwork, and street modified from one-way to two-way.

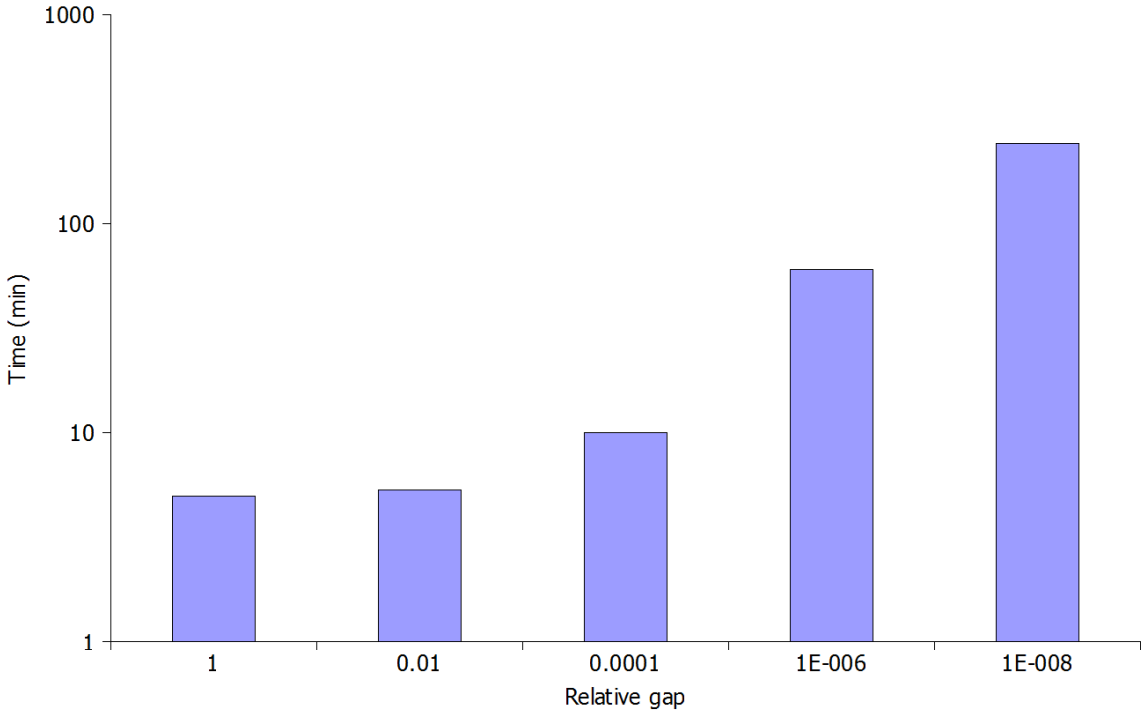


Figure 2.7: Computation time required for solving 212,432 modified equilibrium problems.

Table 2.3: Link-flow RMSE for contracted graph as subproblem accuracy varies.

Relative gap for generating	Relative gap for solving contracted graph			
contracted graph	10^{-2}	10^{-4}	10^{-6}	10^{-8}
1	791.150	778.596	778.596	778.536
10^{-2}	791.102	795.802	778.665	778.530
10^{-4}	791.104	793.825	778.758	778.530
10^{-6}	791.112	795.228	778.600	778.529

Table 2.4: Average corridor travel time error for contracted graph as subproblem accuracy varies.

Relative gap for generating	Relative gap for solving contracted graph			
contracted graph	10^{-2}	10^{-4}	10^{-6}	10^{-8}
1	−3.471%	−3.235%	−3.235%	−3.231%
10^{-2}	−5.817%	−5.982%	−5.439%	−5.435%
10^{-4}	−5.814%	−5.982%	−5.437%	−5.435%
10^{-6}	−5.814%	−5.976%	−5.441%	−5.435%

graph slightly underestimated the travel times).

2.7 Conclusion

This chapter presented a new method for calculating travel time sensitivities on equilibrium bushes which is superior to the two presented in earlier work by Boyles [2012]: it can model interactions between different OD pairs, it requires no planarity assumption, is more stable numerically, is easier to implement using existing code for traffic assignment, and the computation time can be controlled by adjusting a gap termination criterion. Further, simulation results on the Barcelona network indicates that the contracted network can approximate the behavior of the complete network: the error is less that 6.7% even for a demand perturbation as high as 50% and can be reduced to 3.6% by modeling the interaction between OD travel times and OD flows. The application of the proposed approach for subnetwork analysis on the Austin, TX regional network shows that this method achieves comparable accuracy to the other methods while requiring only a tenth of the computation time. It would be useful to validate this performance on other networks, or to develop tailored network design or second-best pricing algorithms based on these

approximations.

It should be noted that the contracted network only has to be constructed once and can be used to evaluate the network performance for different demand scenarios easily with a good approximation. The hybrid approach can be used to improve the accuracy of the model in a bi-level network design problem where hundreds or thousands of equilibrium subproblems are required to be solved. In the hybrid approach, the contracted network is approximated based on the current demand and is fixed. After several iterations of working with the contracted network, it can be updated by re-solving the complete network for the new demand vector. This reduces the estimation error by increasing the consistency between the contracted network and the complete network.

3

Static Traffic Assignment: A Decentralized Approach

This chapter describes a spatial parallelization scheme for the static traffic assignment problem. In this scheme, which we term decentralized static traffic assignment (DSTAP), the network is divided into smaller networks, and the algorithm alternates between equilibrating these networks as subproblems, and master iterations using a simplified version of the full network. The simplified network used for the master iterations is based on linearizations to the equilibrium solution for each subnetwork obtained using sensitivity analysis techniques. We prove that the DSTAP method converges to the equilibrium solution on the full network, and demonstrate computational savings of 35-70% on the Austin regional network. Natural applications of this method are statewide or national assignment problems, or cities with rivers or other geographic features where subnetworks can be easily defined.

3.1 Introduction

The traffic assignment problem (TAP) formulated by Beckmann et al. [1956] is used in transportation planning throughout the world, to predict drivers' route choice, and the resulting flows on roadway links [Patriksson, 2004]. Owing to its elegant formulation as a convex program with an underlying network structure, this problem can be efficiently solved to high precision on city-scale networks using any number of

modern algorithms [Dial, 2006, Bar-Gera, 2010, Gentile, 2014]. However, as computational hardware and algorithms advance, attention shifts to more demanding applications of the traffic assignment problem. These include bilevel programs, whose solution often requires the solution of many TAP instances as subproblems, accounting for forecasting errors with Monte Carlo simulation of input parameters, or broadening the geographic scope of models to the statewide or national levels.

Parallel computing is a general technique for reducing the running time of algorithms, by identifying problem components which can be solved independently, and brought together at a later point in time. Many algorithms for TAP naturally lend themselves to parallelization [Chen and Meyer, 1988, Karakitsiou et al., 2004]. For instance, the classic Frank-Wolfe algorithm can be parallelized by origin or destination when finding shortest paths and building the all-or-nothing link flow vector used in the search direction, and by link when determining the step size.

This paper introduces a new way of parallelizing traffic assignment, by geographic region rather than by origin. In this scheme, which we term the *decentralized static traffic assignment problem* (DSTAP) approach, the network (which we term the *original network* for clarity) is divided into a number of subnetworks. A *regional network* is also created as an abstraction of the original network. The DSTAP algorithm iterates between solving equilibrium on these subnetworks and on the master network, with the demand across the boundaries of the subnetworks obtained from the master network, and the structure of the master network updated based on the subnetwork equilibria. Any algorithm for TAP can be used to solve these equilibrium subproblems.

As is shown in this paper, the DSTAP algorithm converges to the same equilibrium solution as would be obtained for the original network. Numerical experiments on the Austin, TX regional network also show a substantial reduction in computation time (ranging from 35–70%). Although we show correctness on general networks, DSTAP is most obviously suited for assignment problems on networks which naturally divide into subnetworks. Examples include statewide or nationwide models, where clearly-defined urban areas are connected by sparser rural regions, or in cities partitioned by rivers or other geographic features. In this paper we do not consider how best to partition a network into subnetworks, although this is a highly interesting problem for future research.

The remainder of this chapter is organized as follows. Section 3.2 presents a review of current modeling methods for TAP, and research studies related to this modeling approach. Sections 3.3 defines terminology related to the spatial decomposition scheme. Section 3.4 overviews the proposed DSTAP approach, Section 3.5 describes the algorithm in detail, and Section 3.6 provides a proof of its convergence to the original network equilibrium. Section 3.7 presents numerical results when applying DSTAP to a regional network from Austin, TX, and Section 3.8 concludes the chapter.

3.2 Literature Review

This section provides an overview of the existing literature in the following areas: modeling approaches to solve TAP on large scale networks; a review of network aggregation techniques and their applications in the field of transportation planning; and methods to parallelize the solution of TAP.

Many solution methods have been developed to solve the traffic assignment problem, which can be broadly classified into link-based methods, path-based methods, and bush-based methods. Link-based methods require less operational memory and work in the space of link flows to solve the optimization problem [Frank and Wolfe, 1956, LeBlanc et al., 1975, Mitradjieva and Lindberg, 2013]. Path-based methods offer faster convergence compared to link based methods and act on the space of path flows; however, they have larger memory requirements [Florian et al., 2009, Jayakrishnan et al., 1994]. Bush-based methods exploit the fact that the set of used paths from each origin at equilibrium forms an acyclic network [Bar-Gera, 2002, Dial, 2006, Nie, 2010, Bar-Gera, 2010, Gentile, 2014]. These methods have improved the existing state-of-the-art of algorithms and are fast and memory efficient for networks with large scale. However, applying such algorithms to solve equilibrium on very large-scale networks may remain impractical. Even if not computation time is not an issue, more time-efficient methods for TAP require more memory consumption, and a parallelization scheme may be able to reduce these requirements.

Current statewide planning models still rely on aggregation of the networks within cities, capturing only the major freeways and demand using the freeways. For instance, the Texas Statewide Analysis Model (SAM) captures the lower-level transportation system using centroid connectors which serve as an abstract but aggregate representation of urban transportation networks in different cities [TXDOT, 2013]. Many planning models also utilize the aggregation of Traffic Analysis Zones (TAZ) to simplify the network at larger scale. Such techniques are employed by statewide planning models which aggregate the zones and links in MPO models in the statewide network representation [Horowitz, 2006].

Several methods have been proposed to combine the zones and links in a network to form an aggregate network. Link extraction methods remove the unimportant links and nodes from the network [Haghani and Daskin, 1983], but as shown in Chan [1976], such extraction might lead to unpredictable flow patterns on network. Other researchers have proposed link abstraction methods where set of links and nodes between two nodes are replaced with a single aggregated link. Methods proposed to aggregate series and parallel links for purposes of sketch planning are one form of link abstraction [Boyce et al., 1985, Eash et al., 1983].

Recent studies have proposed methods to abstract the links outside a subnetwork in form of an artificial link. Zhou et al. [2006] determines the OD matrices for subnetwork by capturing the behavior of

shift of travelers from the subnetwork using a virtual link. The split proportion for that link is determined using a proportional model based on travel times on paths inside and outside the subnetwork. The virtual link represents trips between boundary nodes that bypass the subnetwork (bypassing paths includes all paths with 50% of its links passing through the subnetwork but are not completely contained), but do not include the paths which are completely (or more than 50%) external to the subnetwork. Hearn [1984] introduces a "transfer decomposition method" to solve traffic assignment on a complete network by dividing it into master and subproblems, where the master problem solves an abstracted network. The approximations to link performance functions for the artificial links are proposed by generating a space of potential artificial link flows, and using the least squares minimization technique to determine the parameters of the polynomial function which minimizes the difference between original network travel times and predicted travel times. Such approximation of parameters for artificial links, however, is computationally taxing to evaluate, especially if there are multiple interactions between boundary nodes of the subnetwork and the origin nodes outside. Barton et al. [1989] later proved that transfer decomposition proposed in Hearn [1984] is equivalent to the generalized Bender's decomposition of the traffic assignment problem. He concludes that such a decomposition method may be efficient for models where the network outside the subnetwork is large but has a smaller number of boundary or "interface" nodes.

The newer network aggregation techniques involving link abstraction rely on sensitivity analysis to estimate changes in travel time between OD pairs with changes in the demand level. Boyles [2012] proposes a bush-based sensitivity analysis of the equilibrium to accurately perform network aggregation, and suggests a flexible boundary approach for modeling subnetwork, highlighting the inability to capture network level changes with the fixed-boundary approaches (e.g. Xie et al. [2010] and Xie et al. [2011]). Jafari and Boyles [2016] and Boyles [2013] present an improved methodology to calculate sensitivity parameters as a convex optimization problem whose structure is identical to the traffic assignment problem making it easier to perform network contraction. This latter method is employed in DSTAP to determine artificial link parameters (as made clear in Section 3.4).

These aggregation procedures are often used in coordination with disaggregation procedures which seek to find solution to the original problem from the solution of aggregated network. The DSTAP algorithm proposed in this paper is one form of an iterative aggregation-disaggregation (IAD) algorithm, where aggregation of subnetworks is done at master level to solve a simplified regional network, and the flow on "artificial links" in the subnetwork is disaggregated to find flows on actual links inside the subnetwork. Rogers et al. [1991] and Dubkin et al. [1987] present a comprehensive review of IAD techniques used in linear, non-linear, and integer optimization. They highlight that convergence of these iterative techniques depends on the aggregation and disaggregation procedure employed, and one primary issue

with these techniques is calculating a bound on computation time for the IAD algorithms to reach within some error range of the original problem solution.

Previous attempts at parallelizing the process of finding network equilibrium includes work by Chen and Meyer [1988], where they distribute finding shortest paths between different OD pairs as independent processes on different threads. Other authors have parallelized a simplified quadratic-knapsack problem for a disaggregate simplicial decomposition algorithm to achieve faster convergence [Lotito, 2006, Karakitsiou et al., 2004] which still hinges on separability of minimization problem in Beckman formulation wrt OD pairs. Work by Damberg and Migdalas [1997] uses similar ideas of parallelizing simplicial decomposition methods. Bar-Gera [2010] also describes parallelization of algorithms based on paired-alternative segments. An alternative way of parallelizing the traffic assignment process is described in Section 3.4.

3.3 Problem Statement

This section reviews the formulation of TAP, presents the problem statement specific to the DSTAP decomposition scheme, and introduces definitions used in the rest of this paper. Figure 3.1 provides an illustration of the definitions given below, for a full network with 2 subnetworks and the following origin-destination (OD) pairs: r-s, r-5, 1-11, 9-5, 2-4 and 9-8. The same network is used throughout the paper in explaining the various concepts associated with the DSTAP algorithm. A glossary of terms and table of notation are provided at the end of the paper, in Tables 3.2 and 3.3.

The *full network* $G = (N, A, W)$ contains all the nodes, links, and OD pairs under consideration, respectively defined by the sets N, A, and W. Let $u \in U$ index the subnetworks. Each subnetwork $G_u = (N_u, A_u, W_u)$ is a subset of the full network, where $N_u \subset N$, $A_u \subset A$, and $W_u \subset W$, with the additional stipulations that A_u contains exactly the links whose tail and head nodes are in N_u, and W_u contains exactly the OD pairs whose origin and destination are both nodes in N_u. These sets respectively contain the *subnetwork nodes*, *subnetwork links*, and *subnetwork OD pairs*, and trips corresponding to subnetwork OD pairs are referred to as *subnetwork demand*. Examples of subnetwork OD pairs are 2-4 and 9-8. The subnetworks do not overlap, so the sets N_u, A_u, and W_u are disjoint across subnetworks u.

The subnetworks also need not form complete partitions of N, A, and W; the sets N_r, A_r, and W_r denote the nodes, links, and OD pairs in the full network which are not part of any subnetwork. These sets include links and OD pairs whose tail and head lie in different subnetworks, or in no subnetwork. These sets are referred to as containing *regional nodes*, *regional links*, and *regional OD pairs*, respectively, and trips corresponding to regional OD pairs are referred to as *regional demand*. In Figure 3.1, the regional nodes N_r are shaded and the regional links A_r are colored, whereas the nodes and links corresponding to

the two subnetworks are not. Examples of regional OD pairs are r-s, r-5, 1-11 and 9-5. In general, we have $N = N_r \underset{u \in U}{\cup} N_u$, $A = A_r \underset{u \in U}{\cup} A_u$, and $W = W_r \underset{u \in U}{\cup} W_u$.

The *boundary nodes* of subnetwork u are denoted by the set $B_u \subseteq N_u$, consisting of subnetwork nodes which are the tail or head node of a regional link. In Figure 3.1, the boundary nodes of subnetwork 1 form the set $\{2, 4, 5, 7\}$, and the boundary nodes of subnetwork 2 form the set $\{8, 9\}$. A path is a *regional path* if its endpoints correspond to a regional OD pair, and a *subnetwork path* if its endpoints correspond to a subnetwork OD pair. A subnetwork path is *internal* if all nodes and links on the path belong to the same subnetwork (path $\pi = \{2, 3, 4\}$ in Figure 3.1), and is *external* if it uses links and nodes from more than one subnetwork (path $\pi = \{2, 5, 8, 12, 9, 7, 4\}$ in Figure 3.1). This paper only considers acyclic paths, and all references to paths will exclude paths which repeats a node.

Let d_w denote the travel demand between regional/subnetwork OD pair w, and let p_w denote the set of paths connecting the endpoints of OD pair w. The delay on each regional/subnetwork link a is given by $t_a(\cdot)$, as a function of its flow x_a which is strictly increasing and differentiable. The user equilibrium (UE) problem seeks the vector of regional and subnetwork link flows $\mathbf{x}$ minimizing the following convex optimization problem:

$$\text{minimize} \quad \sum_{a \in A} \int_0^{x_a} t_a(\omega)\, d\omega \tag{3.1}$$

$$\text{subject to} \quad \sum_{\pi \in p_w} f_\pi = d_w, \qquad \forall w \in W, \tag{3.2}$$

$$\sum_{w \in W} \sum_{\pi \in p_w} f_\pi \delta_{a\pi} = x_a, \qquad \forall a \in A \tag{3.3}$$

$$f_\pi \geq 0, \qquad \pi \in p_w, w \in W. \tag{3.4}$$

where f_π denotes the flow on regional/subnetwork path π and, and the indicator variable $\delta_{a\pi}$ is 1 if path π uses link a, and 0 otherwise.

The feasible region, defined by equations (3.2)–(3.4), forms a compact, convex polyhedral set. This equilibrium problem on the full network can be solved using any number of modern algorithms [Bar-Gera, 2002, Dial, 2006, Bar-Gera, 2010, Gentile, 2014], but we propose a parallelization scheme which may offer faster convergence.

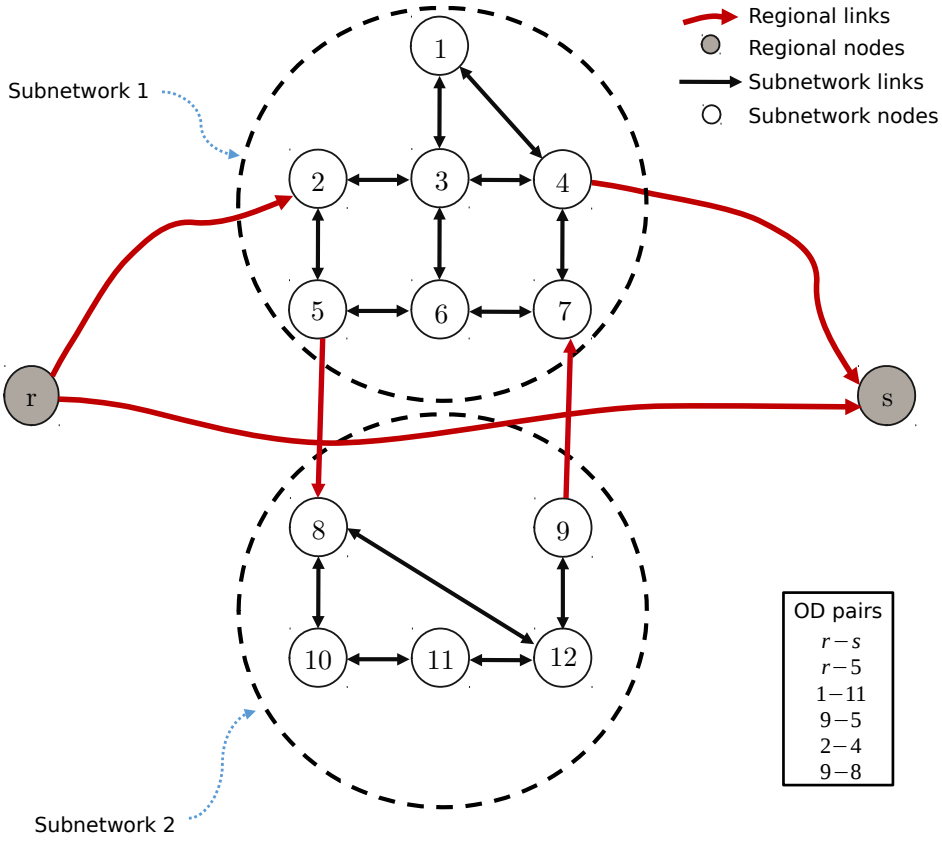


Figure 3.1: A full network with two subnetworks.

3.4 A Spatial Decomposition Algorithm: Overview

The traffic assignment problem is typically solved as one problem without regard to the subnetworks. In this paper, we call this a *centralized approach*. A *decomposition approach*, by contrast, divides the assignment by subnetwork, which may have computational benefits. This paper describes a new algorithm following a decomposition approach, which we term DSTAP (decomposed approach for the static traffic assignment problem). The DSTAP algorithm considers $|U|+1$ assignment problems for a network with a set U of subnetworks: there are $|U|$ subproblems, one corresponding to each subnetwork, and one master problem which is derived from the full network. Each iteration of DSTAP starts by *partially* solving the master problem, using the most recent information on travel times from the subproblems. The output of the master problem provides the regional demand for each subproblem ("external trips" in planning parlance). The subproblem flows and travel times are then updated and solved in a parallel fashion. As discussed later, the set-up of regional network and subnetworks rely on equilibrium sensitivity analysis based on path flow information. As the DSTAP algorithm we describe is path-based, this information is available at each iteration.

The master problem assigns regional demand to a simplified version of the full network, referred to as the *regional network* $G_a = (N_a, A_a, W_a)$, where subnetworks are aggregated and modeled by some artificial links, referred to as *artificial regional links*, alongside the regional nodes and links as defined in the previous section. The origin and destination nodes of the artificial regional links are subnetwork nodes, which are either the boundary nodes or nodes serving as origin or(and) destination of regional demand. These artificial links are created to represent the routes for regional demand which go through subnetworks. The regional demand assigned to each artificial regional link in the master problem represents the amount of regional demand which travels between the origin and destination nodes of the artificial link in the associated subnetwork. This demand is used to update the subnetwork OD trips before running the subproblems.

Each subproblem solves the assignment problem on one subnetwork augmented by some artificial links, referred to as *artificial subnetwork links*. These artificial subnetwork links are created to represent the routing of subnetwork demand which exit the boundary of subnetwork and go through other subnetworks (subnetwork external paths). Let l denote an artificial subnetwork link added to subnetwork $u_1 \in U$ representing routing through subnetwork u_2. Similar to artificial regional links, the demand assigned to l specifies the subnetwork demand from u_1 which travels between the origin and destination nodes of l in subnetwork u_2 and need to be included when solving subproblem of u_2. The artificial regional/subnetwork link parameters are dynamic and need to be updated each iteration.

For each OD pair in the regional network, we perform one shift from each used regional path to the shortest regional path connecting its OD pair. Next, OD demands in each subnetwork are updated to represent changes in external trips from the regional path shifts. Next, the subproblems are solved in parallel. For each OD pair in subnetwork u, we perform one shift from each used subnetwork path to the shortest subnetwork path connecting its OD pair. After all shifts are performed, the artificial regional/subnetwork links are updated. This finishes one iteration of the DSTAP algorithm. If a convergence criterion is not satisfied, these steps are repeated.

We propose a convergence criterion based on the maximum excess cost in the master problem. The proposed stopping condition needs to be checked at the beginning of each DSTAP iteration, after the artificial links are updated and before the master problem is solved. We prove the convergence of the DSTAP algorithm using Zangwill's global convergence theorem and show that, upon termination, the flow assignment in DSTAP corresponds to an equilibrium flow in the full network.

3.5 DSTAP Algorithm

In this section, we present a formal definition of the master problem and subproblems introduced in the DSTAP approach, and then discuss the algorithmic details.

3.5.1 Master Problem

The regional network, which forms the basis for the DSTAP master problem, includes all the regional links, regional nodes, boundary nodes, subnetwork nodes which are origins or destinations of regional demand, and some artificial regional links. The artificial regional links represent abstractions of the subnetworks, and aim to capture interactions between regional and subnetwork demands in a simplified manner. These artificial regional links are created between two nodes in the same subnetwork which can be used by regional demand, and must satisfy two conditions: if (i, j) is an artificial regional link, then (1) node j is reachable from i using only subnetwork links, and (2) both i and j correspond either to a subnetwork boundary node or an endpoint of a regional OD pair; and in this latter case, i must precede j in some regional path.

The artificial regional links in the regional network are created in three steps. First, for each subnetwork u, artificial links are created between the boundary nodes B_u. These boundary-boundary links model the routing of regional demand through subnetwork u. The second step constructs artificial links between the boundary nodes and subnetwork destinations with regional demand. These boundary-destination links are used by regional demand with destination in u. Finally, we create artificial links between the

subnetwork origins with regional demand and boundary nodes. The origin-boundary links created in subnetwork u will be used by regional demand starting from subnetwork u.

Figure 3.2 shows the regional network of the full network shown in Figure 3.1. In Figure 3.2, links $(2, 4)$, $(2, 5)$, $(7, 4)$, and $(7, 5)$ in subnetwork 1, and link $(8, 9)$ in subnetwork 2, are boundary-boundary links and created in the first step. We do not create artificial links from boundary nodes 2, 4 and 5 to boundary node 7 in subnetwork 1, because the second condition is violated: there is no demand destined to node 7, and regional demand can not leave subnetwork 1 at node 7. Nodes 5 and 11 are destination nodes with regional demand in subnetworks 1 and 2, respectively. In the second step, we create the boundary-destination links $(2, 5)$ and $(8, 11)$. Finally, the origin-boundary link $(1, 5)$ is created as the last step (origin node 9 in subnetwork 2 is a boundary node, and the only origin-boundary link we could create is $(9, 8)$ which violates condition 2.)

Note that the artificial links created in mentioned steps are not disjoint, and some of them may fall into more than one category. For example, link $(2, 5)$ is both a boundary-boundary and boundary-destination link. In such cases, only one link is created.

Now we investigate how the routing options in the full network correspond to routing options in the regional network in Figure 3.2.

1. OD pair r-s: in the full network of Figure 3.1, there are 3 types of routes available: using the direct regional link (r, s); entering subnetwork 1 at node 2 and leavnig it at node 4 using links and nodes which are internal to subnetwork 1; and entering subnetwork 1 at node 2 and leaving it at node 4, but first passing through subnetwork 2 using regional links $(5, 8)$ and $(9, 7)$. All these routing options are present in the regional network of Figure 3.2: in the first case, the direct link (r, s) is present; in the second case, option 2 is modeled by the artificial link $(2, 4)$; and in the third case, through the artificial links $(2, 5)$, $(8, 9)$ and $(7, 4)$.

2. OD pair r-5: all routes for this OD pair enter subnetwork 1 at node 2 and finish the trip at node 5. This option is modeled in the regional network by the artificial link $(2, 5)$. Note that the artificial link $(2, 5)$ is not a copy of the subnetwork link $(2, 5)$, but rather represents *all* used paths from 2 to 5 in subnetwork 1.

3. OD pair 1-11: all routes exit subnetwork 1 at node 5, enter subnetwork 2 at node 8, and finish the trip at node 11. The artificial links $(1, 5)$ and $(8, 11)$ model this routing option.

4. OD pair 9-5: all routes enter subnetwork 1 at node 7 and travel to node 5 in subnetwork 1. The artificial link $(7, 5)$ provides this option.

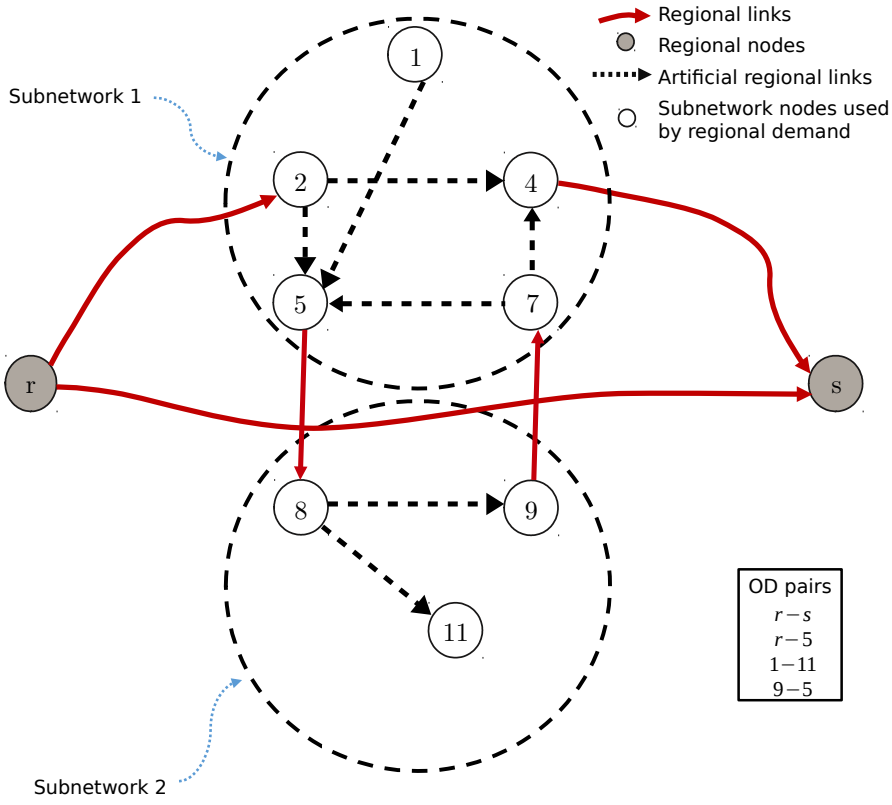


Figure 3.2: Regional network solved in the DSTAP master problem, containing all regional links and regional nodes, and artificial regional links.

Each artificial regional link represents all used paths in the subnetwork connecting its tail to its head, and will be equipped with cost functions which represent the equilibrium travel time between these nodes in the subnetwork, as a function of the regional demand between these points. These cost functions are derived from a linear approximation of the subnetwork equilibrium solutions, obtained through a sensitivity analysis procedure described in Section 3.5.4. Let Θ_u denote the set of artificial regional links created for subnetwork u in the regional network. As the flow on each artificial regional link corresponds to external demand for subnetwork u, we will also use Θ_u to denote the associated set of OD pairs in subnetwork u. At iteration $k+1$ of the DSTAP algorithm, these artificial regional links have the following cost function:

$$t_\theta^{k+1}(x_{\theta,r}^{k+1}) = \mu_\theta^k + \psi_\theta^k(x_{\theta,r}^{k+1} - x_{\theta,r}^k), \qquad \forall \theta \in \Theta_u, u \in U \tag{3.5}$$

where the superscript is the iteration number, $t_\theta^{k+1}(\cdot)$ denotes the travel time variable, $x_{\theta,r}^{k+1}$ is the amount of regional demand on the artificial regional link θ at iteration $k+1$, $x_{\theta,r}^k$ is the regional demand using the link at the previous iteration, which is fixed, and μ_θ^k and ψ_θ^k respectively denote the average travel time of the paths represented by θ (weighted by flow), and the derivative of this average travel time for the artificial regional link θ. Equation (4.23) has three parameters from the previous iteration k: $x_{\theta,r}^k$, given by the master problem, and μ_θ^k and ψ_θ^k, which are obtained from the subproblems. In the above formulation, the derivative ϕ_θ^k only exists if the solution is strictly complementary in the sense that all minimum-cost routes have positive flow. This assumption is common in the sensitivity analysis literature [Tobin and Friesz, 1988, Cho et al., 2000, Patriksson, 2004, Josefsson and Patriksson, 2007, Yang and Bell, 2007b]. In practice, it can be difficult to determine whether a solution violates strict complementarity, because equilibria are only solved to finite precision (cf. Bar-Gera [2006]). However, since non-complementary solutions are a zero-measure set [Patriksson, 2004], and the sensitivity analysis procedure described below still produces a directional derivative in noncomplementary cases, we feel this assumption is not limiting in practice. Furthermore, the parameters ϕ_θ^k are only used to help determine the step size using Newton's method, and the proof of convergence in Section 3.6 does not require the equilibrium cost to be differentiable at each iteration.

The cost functions on regional links also need to be modified, to account for subnetwork demand which uses regional links. The resulting *biased cost functions* $\tilde{t}_a^{k+1}(\cdot)$ are defined as follows. Let $x_{a,s}^k$ denote the subnetwork demand assigned to the regional link a as a result of solving the subproblems at iteration k. The biased link cost function is given by:

$$\tilde{t}_a^{k+1}(x_{a,r}^{k+1}) = t_a(x_{a,s}^k + x_{a,r}^{k+1}), \qquad \forall a \in A_r \tag{3.6}$$

where $x_{a,s}^k$ is fixed and $x_{a,r}^{k+1}$ is the regional link flow variable. More discussion on these bias terms and their values is found in the following subsection, when the subproblems and the structure of artificial subnetwork links are presented.

The master problem at iteration $k+1$ is then given by:

$$\text{minimize} \quad \sum_{a \in A_r} \int_0^{x_{a,r}^{k+1}} \tilde{t}_a^{k+1}(\omega)\, d\omega + \sum_{u \in U} \sum_{\theta \in \Theta_u} \int_0^{x_{\theta,r}^{k+1}} t_\theta^{k+1}(\omega)\, d\omega \tag{3.7}$$

$$\text{subject to} \quad \sum_{\pi \in p_w} f_\pi^{k+1} = d_w, \qquad \forall w \in W_r \tag{3.8}$$

$$\sum_{v \in W_r} \sum_{\pi \in p_v} f_\pi^{k+1} \delta_{a\pi} = x_{a,r}^{k+1}, \qquad \forall a \in A_r \tag{3.9}$$

$$\sum_{v \in W_r} \sum_{\pi \in p_v} f_\pi^{k+1} \delta_{\theta\pi} = x_{\theta,r}^{k+1}, \qquad \forall \theta \in \Theta_u, u \in U \tag{3.10}$$

$$f_\pi^{k+1} \geq 0, \qquad \forall \pi \in p_{w,R}, w \in W_r \tag{3.11}$$

The first term of the objective function includes regional links with their biased cost functions and the second term sums over the artificial regional links with cost defined in (4.23). The constraint set only includes regional OD pairs, regional links and artificial regional links; notice that the allowable path set in the regional network is $p_{w,R}$, which is distinct from the path set in the full network p_w due to the aggregation of subnetworks. The solution to this problem specifies the flow assigned to regional links, $x_{a,r}^{k+1}$, and artificial regional links, $x_{\theta,r}^{k+1}$, at iteration $k+1$. The flow on artificial links will be used to update the demand of the associated subnetwork OD pairs.

3.5.2 Subproblems

Each of the $|U|$ subproblems finds equilibrium on one of the subnetworks while approximating interactions with the regional network and other subnetworks. To represent regional demand from the master problem, the set of subnetwork OD pairs W_u is augmented with an OD pair for each artificial regional link in the set Θ_u where the origin and destination correspond to the tail and head nodes of each link $\theta \in \Theta_u$. These OD pairs represent external trips from regional demand. For example, after solving the network in Figure 3.2, the demand between OD pair 2-4 in subnetwork 1 needs to be adjusted based on the regional demand assigned to artificial regional link $(2,4)$ ($d_{24} \leftarrow d_{24} + x_{24,r}^{k+1}$), and for artificial regional link $(8,9)$, we first create OD pair 8-9, if not already created in previous iterations, and then set its demand equal to the regional flow assigned to artificial regional link $(8,9)$, i.e., $d_{89} \leftarrow x_{89,r}^{k+1}$. A similar procedure needs to

be implemented for the remaining 5 artificial regional links and their associated subnetwork OD pairs.

Furthermore, the set of subnetwork links A_u is augmented by artificial subnetwork links between boundary nodes. The artificial subnetwork links represent the external subpaths: possibility of subnetwork demand to leave the subnetwork, route through regional links and perhaps other subnetworks, and re-enter the subnetwork at a later point. Consider the subproblems of the full network in Figure 3.1, as shown in Figure 3.5. For subnetwork 1, we create one artificial link between boundary nodes 5 and 7 because subnetwork demand can leave the subnetwork at node 5 and re-enter at node 7 by traveling through regional links $(5, 8)$, subnetwork 2, and $(9, 7)$. Similarly, artificial link $(9, 8)$ added to subnetwork 2 represents the opportunity for subnetwork demand from subnetwork 2 to travel through subnetwork 1. OD pairs are also added to correspond to the artificial links created in the regional network (compare with Figure 3.2).

Let $x_{\gamma,u}^{k+1}$ denote the flow assigned to artificial subnetwork link γ added to subnetwork u to represent routing through subnetwork ν between the end points of OD pair $w \in W_\nu$ at iteration $k+1$, and let Γ_u denote the set of artificial subnetwork links added to subnetwork u. The cost function of these artificial subnetwork links at iteration $k+1$, denoted by $t_\gamma^{k+1}(\cdot)$, is similar to (4.23):

$$t_\gamma^{k+1}(x_{\gamma,u}^{k+1}) = \lambda_\gamma^k + \phi_\gamma^k(x_{\gamma,u}^{k+1} - x_{\gamma,u}^k), \qquad \forall \gamma \in \Gamma_u, u \in U \tag{3.12}$$

where $x_{\gamma,u}^{k+1}$ denotes the flow variable, $x_{\gamma,u}^k$ is the flow assigned to the artificial subnetwork link at iteration k, λ_γ^k is the average travel time between the endpoints of OD pair w in subnetwork ν at iteration k (weighted by current flows), and ϕ_γ^k is the derivative of this average travel time.

Th reader may skip this discussion without loss of continuity.

Discussion

The external paths, in general, may traverse multiple regional links and subnetworks, and in such a case, to represent all these external paths, the subnetwork can be equipped with regional links and some artificial links representing the paths between the boundary nodes of other subnetwork. The resulting network resembles the network used in the master problem, but retains the full subnetwork and only assigns subnetwork demand.

As an example, consider the network shown in Figure 3.3, which has 3 subnetworks (for simplicity, this figure only shows the boundary nodes and their connections with other subnetworks). Assume that flow exiting subnetwork 1 at node a_1 may travel through subnetwork 2 and subnetwork 3 before returning

back to subnetworks 1 at node a_2. Then the set of alternative external subpaths for subnetwork 1 can be modeled by adding the regional links and artificial links for the other subnetworks. Figure 3.4 shows subnetwork 1 augmented with regional links (continuous line) and artificial links (dashed line). Similarly, one can modify subnetworks 2 and 3 to represent the external subnetwork paths.

If there are multiple regional segments directly connecting nodes a_1 and a_2 without going through other subnetworks, the regional segments can be directly added to the subnetwork. If this is computationally prohibitive (e.g., if there are many such routse), one may perform a sensitivity analysis on the regional network to simplify all these segments between a_1 and a_2 with one artificial link. The procedure for performing sensitivity analysis on the regional network for the purpose of estimating the parameters of the new artificial link would be the same as the one discussed for the case of subnetworks.

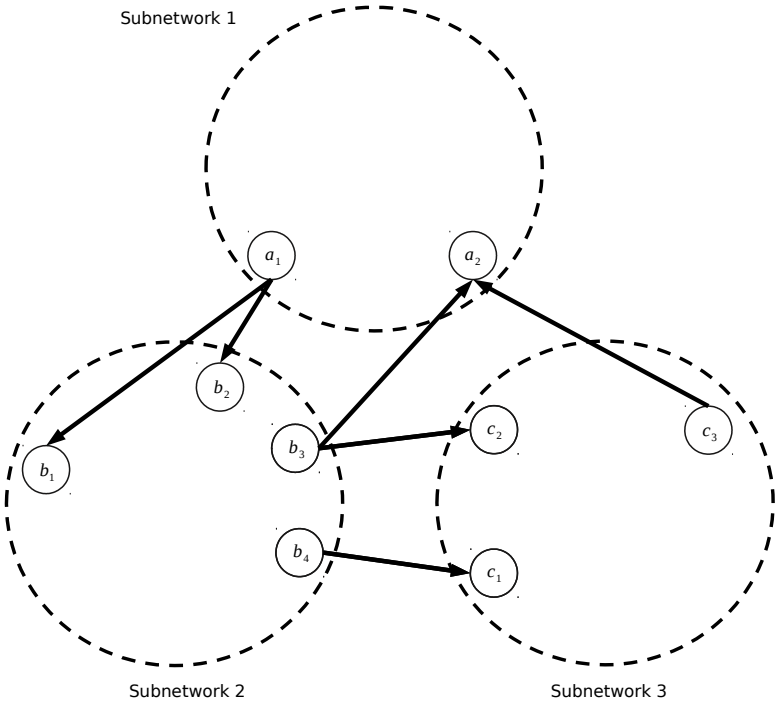


Figure 3.3: A regional network with three subnetworks.

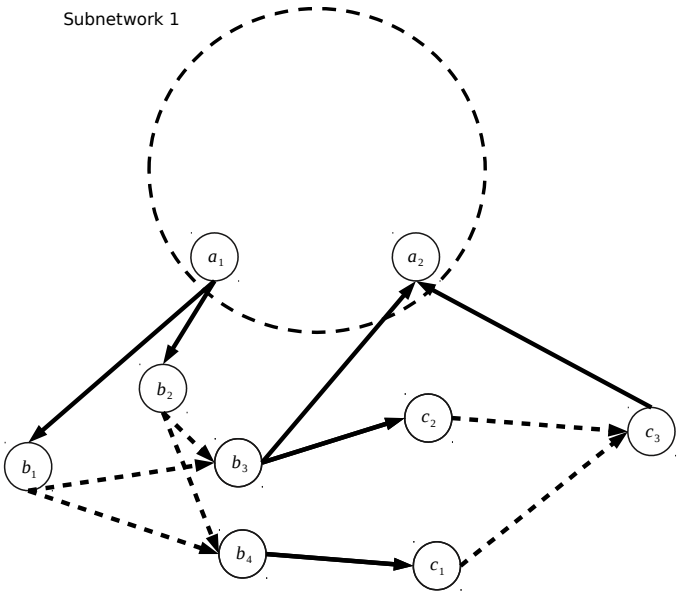


Figure 3.4: Subnetwork 1 augmented with multiple regional and artificial links to model external paths.

The demand in subnetwork u needs to be updated considering the flow assigned from other subnetworks' artificial links in previous iteration. Let $d_w^k(\nu)$ denote the flow from subnetwork ν assigned to travel between endpoints of OD pair w in subnetwork u at iteration k and Δ_u denote the set of OD pairs in subnetwork u which correspond to artificial subnetwork links in other subnetworks. For subnetwork 1 in Figure 3.5, $d_{75}^k(2) = x_{98,2}^k$ and $\Delta_1 = \{7\text{-}5\}$, and for subnetwork 2 we have $d_{89}^k(1) = x_{57,1}^k$ and $\Delta_2 = \{8\text{-}9\}$. The demand for an OD pair $w \in \Delta_u$ is updated considering the flow on artificial links representing that OD pair in other subnetworks at the previous iteration ($d_w^{k+1} \leftarrow d_w + \sum_{\nu \in U \setminus u} d_w^k(\nu)$). For example, for iteration $k+1$, we first create the OD pair $7-5$ in subnetwork 1 corresponding to the artificial link $(9, 8)$ in subnetwork 2, if not already existing in previous iterations, and then set the demand to include the flow on the link in previous iteration ($d_{75} \leftarrow d_{75}^k(2)$).

The demand in a subnetwork for each iteration is thus influenced by the master problem and other subnetworks, through external trips which enter or leave the subnetwork at its boundary nodes. Based on whether an OD pair w belongs to Θ_u and/or Δ_u, the adjusted OD demand ($\breve{d}_w^{k+1}$) at iteration $k+1$ is

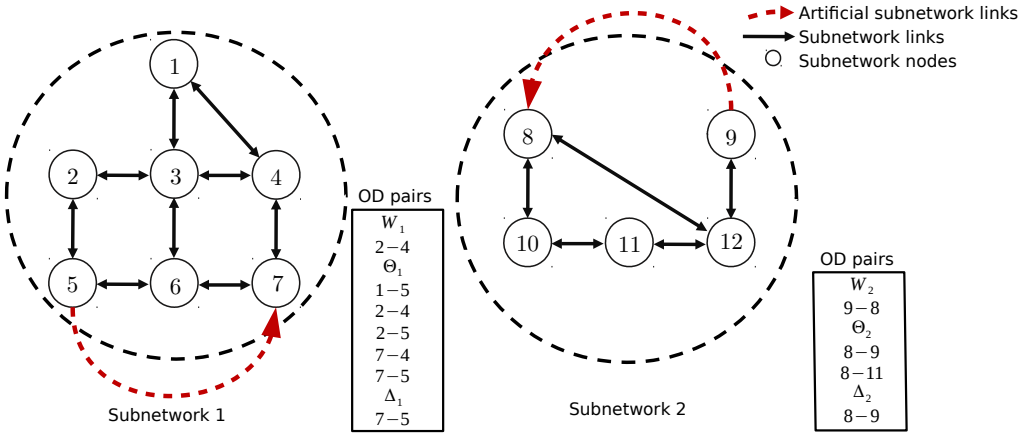


Figure 3.5: Subnetworks solved as the subproblems of DSTAP. Networks have all the subnetwork links, nodes and some artificial subnetwork links.

given by equation 3.13.

$$\breve{d}_w^{k+1} = \begin{cases} d_w + x_{w,r}^{k+1} & \text{if } w \in \Theta_u, w \notin \Delta_u \\ d_w + \sum_{\nu \in U \setminus u} d_w^k(\nu) & \text{if } w \notin \Theta_u, w \in \Delta_u \\ d_w + x_{w,r}^{k+1} + \sum_{\nu \in U \setminus u} d_w^k(\nu) & \text{if } w \in \Theta_u, w \in \Delta_u \\ d_w & \text{if } w \notin \Theta_u, w \notin \Delta_u \end{cases} \tag{3.13}$$

where $x_{w,r}^{k+1}$ denotes the regional demand assigned to OD pair w through the associated artificial regional link.

Let $\breve{W}_u = W_u \cup \Theta_u \cup \Delta_u$ denote the set of OD pairs in subnetwork u together with external demand pairs from the regional network, Θ_u, or from other subnetworks, Δ_u. The subproblem for subnetwork

u at iteration $k+1$ is given by:

$$\text{minimize} \quad \sum_{a \in A_u} \int_0^{x_{a,u}^{k+1}} t_a(\omega)\, d\omega + \sum_{\gamma \in \Gamma_u} \int_0^{x_{\gamma,u}^{k+1}} t_\gamma^{k+1}(\omega)\, d\omega \tag{3.14}$$

$$\text{subject to} \quad \sum_{\pi \in p_w} f_\pi^{k+1} = \breve{d}_w^{k+1}, \qquad \forall w \in \breve{W}_u \tag{3.15}$$

$$\sum_{w \in \breve{W}_u} \sum_{\pi \in p_w} f_\pi^{k+1} \delta_{a\pi} = x_{a,u}^{k+1}, \qquad \forall a \in A_u \tag{3.16}$$

$$\sum_{w \in \breve{W}_u} \sum_{\pi \in p_w} f_\pi^{k+1} \delta_{\gamma\pi} = x_{\gamma,u}^{k+1}, \qquad \forall \gamma \in \Gamma_u \tag{3.17}$$

$$f_\pi^{k+1} \geq 0, \qquad \forall \pi \in p_w, w \in \breve{W}_u \tag{3.18}$$

Now we can discuss the bias term $x_{a,s}^k$ in the travel time function of regional link a, described in equation (3.6). After all subproblems are solved, the bias term $x_{a,s}^k$ can be determined by summing subnetwork demands assigned to regional link a:

$$x_{a,s}^k = \sum_{u \in U} \sum_{\gamma \in \Gamma_u} x_{\gamma,u}^k \delta_{a\gamma}, \qquad \forall a \in A_r \tag{3.19}$$

where $\delta_{a\gamma}$ is 1 if regional link a is part of artificial subnetwork link γ and 0 otherwise. This bias term plays the role of the background flow when solving the master problem at iteration $k+1$.

3.5.3 Algorithm

Figure 3.6 is a flowchart of the DSTAP algorithm, in which the subproblems are solved in parallel. The algorithm alternates between shifting flow on the regional network (the master problem described in Section 3.5.1), and on the augmented subnetworks described in Section 3.5.2.

The algorithm starts by checking a gap measure on the regional network and continues by solving the regional network if the stopping condition is not met. In Section 3.6, we prove correctness and convergence of the algorithm. In the results presented in this paper, our termination criterion is the maximum excess cost, defined by the greatest difference between the longest used path and shortest path for each OD pair. The convergence gap value is given by ϵ_{od}^*.

Flows on both the regional network, problem (3.7)–(3.11), and each subnetwork $u \in U$, problem (3.14)–(3.18), are shifted according to the Goldstein-Levitin-Polyak gradient projection algorithm proposed by Bertsekas [1976] and investigated in Jayakrishnan et al. [1994] for the traffic assignment problem. The

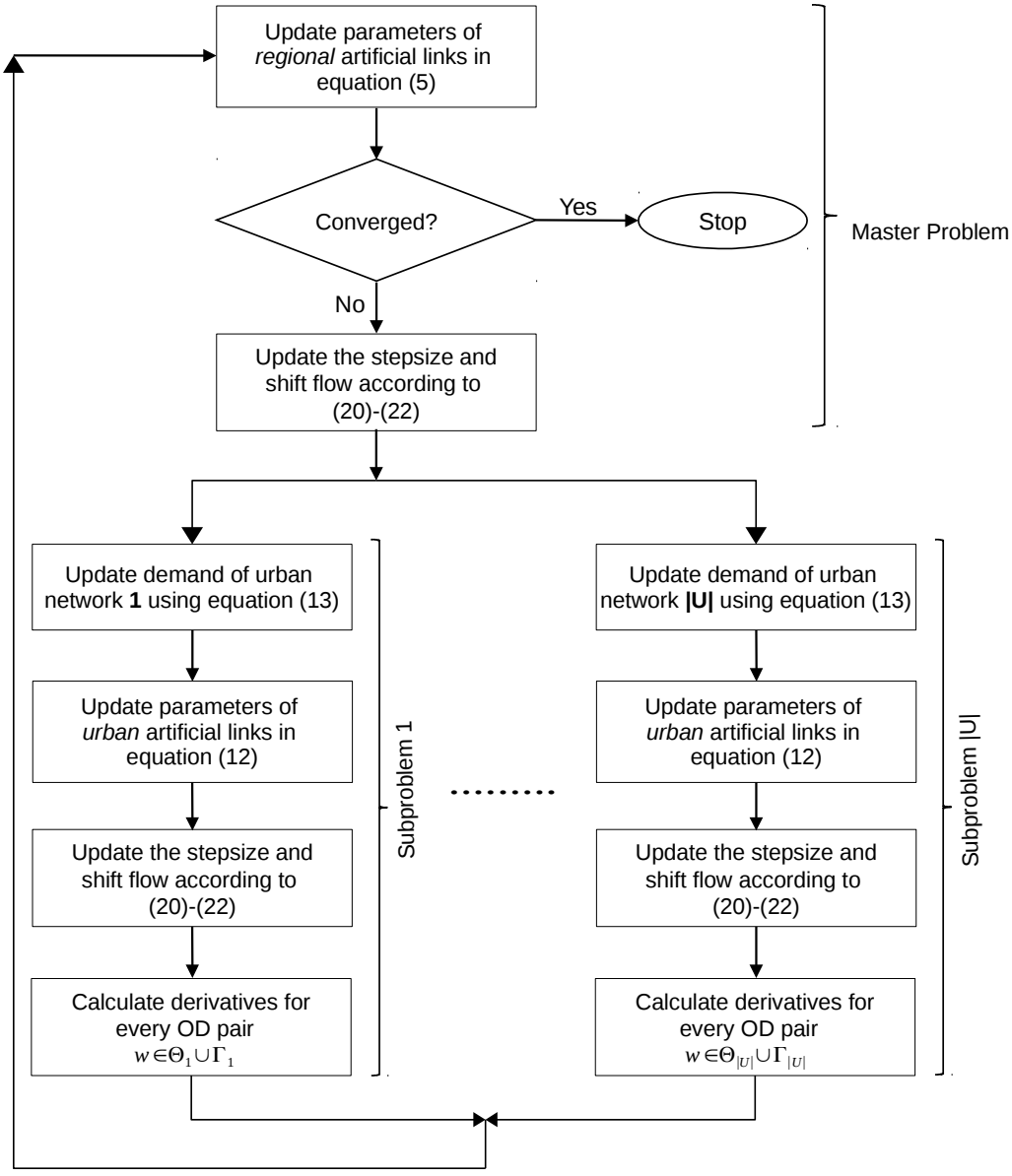


Figure 3.6: Flowchart of the DSTAP algorithm.

path flows for regional OD pair w are updated by moving along the direction opposed to the gradient:

$$f_\pi \leftarrow \left[f_\pi - \alpha(T_\pi - T_{b(\pi)})\right]^+, \qquad \forall \pi \in \widehat{p}_w \tag{3.20}$$

$$f_{b(\pi)} = d_w - \sum_{\pi \in \widehat{p}_w} f_\pi \tag{3.21}$$

where T_π is the travel time on path π, and $b(\pi)$ is a shortest path corresponding to the OD pair whose endpoints are the same as those of π (choosing the same shortest path for all such paths), $T_{b(\pi)}$ is travel time on $b(\pi)$, $\widehat{p}_w$ is the set of used paths excluding shortest path $b(\pi)$, and α is the stepsize. In case of ties, the choice of shortest paths $b(\pi)$ for OD pairs can be made arbitrarily. To gain a sharper rate of convergence, Bertsekas [1976] suggests a Newton-type step size, using second derivative information. To this end, one can scale the stepsize α with the second derivative s_π (i.e. $\alpha \leftarrow \alpha s_\pi$) in (3.20). The variable s_π may be formulated as:

$$s_\pi = \frac{1}{\sum_{a \in A_r} \tilde{t}'_a |\delta_{a\pi} - \delta_{ab(\pi)}| + \sum_{u \in U} \sum_{\theta \in \Theta_u} t'_\theta |\delta_{\theta\pi} - \delta_{\theta b(\pi)}|} \tag{3.22}$$

where $\tilde{t}'_a$ and t'_θ are, respectively, the derivative of biased regional link a travel time and artificial regional link θ travel time with respect to link flow evaluated at the current flow. The formulation for subnetwork flow shifts are identical. (This choice of algorithm is assumed in the convergence proof in Section 3.6, although it seems likely that other algorithm choices for the master problem will converge as well.)

When solving the regional network and subnetworks, the step size α, scaled with the second derivative s_π, in equation (3.20) plays an important role in the convergence of the algorithm and a large stepsize may result in a slow convergence or even divergence. This issue is similar to that faced in *trust region* algorithms for nonlinear programming, which improve the current solution by constructing an approximate model around the current solution. The approximate model is trusted to be a good approximation near to the current solution, but less so beyond the trust region. The general approach for using a trust region approach is to adjust the trust region from iteration to iteration. More precisely, if the new solution provides enough improvement for the original problem, then the trust region can be extended, otherwise we need to shrink the trust region. For more discussion on trust region approaches, refer to Yuan [2000].

In DSTAP, the artificial regional/subnetwork links represent an approximate model constructed based on the current subnetwork demand and flow assignment. In particular, the derivative information obtained from the sensitivity analysis procedures, discussed in Section 3.5.4, is only valid when the set of used paths do not change. A large change in the regional path flows, resulting from a large stepsize, may push

the subnetwork OD demands beyond the trusted region. Hence, we propose to adjust the stepsize at each iteration based on the gap value. The gap measure, before adjusting the path flows, can be used as a measure of performance of the trust region. Specifically, the algorithm starts with an initial stepize of α^0 and updates it as $\alpha^{k+1} = \zeta\alpha^k$ such that $\alpha^{k+1} \leq 1$. If the initial value of the gap measure at iteration $k+1$ is worse than iteration k, then we reduce the trust region by setting $\zeta < 1$, and if the new gap is lower, the trust region can be extended by setting $\zeta > 1$. This allows the algorithm to explore points further from the current solution. The best value of ζ is network and implementation dependent.

3.5.4 Calculating parameters for artificial links

The travel time function of artificial regional link θ created for subnetwork u in the regional network, defined in equation (4.23), involves two parameters: μ_θ^k and ψ_θ^k. The former represents the travel time between the endpoints of θ (an OD pair $w \in W_u$) in the subnetwork u, which can be directly computed from the assignment results of u at iteration k as the flow-weighted average of the subnetwork path costs:

$$\mu_\theta^k = \sum_{\pi \in p_\theta} \rho_\pi T_\pi \tag{3.23}$$

where the notation $\pi \in p_\theta$ defines all used subnetwork paths represented by the artificial link θ, T_π is travel time on path π, and

$$\rho_\pi = \frac{f_\pi}{\sum_{\pi' \in p_\theta} f_{\pi'}} \tag{3.24}$$

is the proportion of flow on path π.

The second parameter, ψ_θ^k, is the derivative of travel time between the endpoints of θ in the subnetwork u with respect to adjusted OD demand $\breve{d}_w^k$, accounting for the allocation of this demand across all used paths. As shown in Josefsson and Patriksson [2007] and Jafari and Boyles [2016], this partial derivative ψ_θ^k under the assumption of strict complementarity of route flows, is given by the solution to the following convex program:

$$\text{minimize} \sum_{a \in \tilde{\mathbf{A}}_w} \int_0^{\varrho_a^w} t_a'(\omega)\, d\omega \tag{3.25}$$

$$\text{subject to} \sum_{\pi \in p_w} \beta_\pi^w = 1, \tag{3.26}$$

$$\varrho_a^w = \sum_{\pi \in p_w} \beta_\pi^w \delta_{a\pi}, \quad \forall a \in \tilde{\mathbf{A}}_w \tag{3.27}$$

where $\tilde{\mathbf{A}}_w$ is the set of all links used by $\breve{d}_w^k$ at the current solution, $t'_a = \mathrm{d}t_a/\mathrm{d}x_{a,u}$ is the derivative of link a travel time with respect to link a flow evaluated at current assignment at iteration k, $\varrho_a^w = \partial x_{a,u}/\partial \breve{d}_w^k$ is the derivative of link a flow with respect to $\breve{d}_w^k$ evaluated at current demand at iteration k, $\beta_\pi^w = \partial f_\pi/\partial \breve{d}_w^k$ is the derivative of path π flow with respect to $\breve{d}_w^k$, and $\delta_{a\pi}$ is 1 if link a is used by path π. The OD travel time derivative can be written as:

$$\psi_\theta^k = \sum_{a \in \tilde{\mathbf{A}}_w} t'_a \varrho_a^w \delta_{a\pi} \tag{3.28}$$

where π can be any of the used paths for OD pair w.

Similarly, the artificial subnetwork link γ added to subnetwork u to model the alternative routes in subnetwork ν between the endpoints of OD pair w, and formulated in equation (3.12), requires two parameters from subnetwork ν: λ_γ^k, which is the travel time between the OD pair w in the subnetwork ν, formulated as the flow-weighted average of the subnetwork path costs (similar to (3.23) and (3.24)), and ϕ_γ^k, which is the derivative of OD travel time with respect to $\breve{d}_w^k$. The OD travel time derivative ϕ_γ^k can be computed using a convex program similar to (3.25)–(3.27) and (3.28) albeit for OD pair w within subnetwork ν.

The convex problem defined by (3.25)–(3.27) must be solved for each subnetwork $u \in U$ and each OD pair $w \in \Delta_u \cup \Theta_u$. These problems are essentially static equilibrium problems on the networks comprised of links in $\tilde{\mathbf{A}}_w$, with ϱ_a^w and β_π^w serving the role of link flows and path flows, respectively, linear cost functions of the form $t'_a \varrho_a^w$, unit demand between OD pair w, and no nonnegativity constraints on flow variables. This problem can be solved by making minor modifications to algorithms for the traditional TAP.

The derivation in Jafari and Boyles [2016] conducted this sensitivity analysis under the assumption that the initial flow were at equilibrium. The interpretation of the derivatives here are slightly different, because in DSTAP the flow at iteration k may not satisfy the equilibrium principle. The sensitivity analysis implies that any small change in OD demand will distribute among currently used paths such that they all experience the same change in their travel time, and this equal change is given by these derivatives. More details and computational experiments on this method can be found in Jafari and Boyles [2016] or Chapter 2 of this dissertation.

After updating the artificial link parameters at iteration $k+1$ of the DSTAP algorithm (Figure 3.6), the master problem starts with the flow assigned to the regional network at iteration k ($x_{\theta,r}^{k+1} = x_{\theta,r}^k, x_{a,r}^{k+1} = x_{a,r}^k$). This flow is adjusted to obtain the new flow assignment $x_{\theta,r}^{k+1}$ and $x_{a,r}^{k+1}$. The subproblems at iteration $k+1$, after updating the subnetwork demands and artificial subnetwork links, start with their

assignment solution from iteration k and adjust the flows on current set of used paths, proportionally inflating or deflating the path flows, to ensure feasibility of new regional flows.

3.6 Algorithm Correctness

Here we discuss properties of the DSTAP algorithm and then prove its correctness and convergence.

3.6.1 Correctness

This subsection discusses the termination criterion used for the entire DSTAP algorithm (the "*DSTAP converged?*" block). This leads directly to a discussion of algorithm correctness, that at termination the solution obtained from the DSTAP algorithm corresponds to the equilibrium solution obtained from a centralized algorithm applied to the full network. This is a separate issue from *convergence* to this correct solution, which is discussed in the following subsection.

More specifically, we show that near-equilibrium solutions from DSTAP map to near-equilibrium solutions on the full network, for any given convergence threshold. This is sufficient to show correctness, since all algorithms for TAP only converge to the equilibrium solution in the limit. The proposed convergence measure for the entire DSTAP algorithm is the maximum excess cost, calculated *before* solving the master problem. We believe it likely that similar properties can be established for other convergence measures as well, such as average excess cost or relative gap.

In this section, we use $\mathbf{f}^k$ to denote the path flow vectors at the end of iteration k of DSTAP, while $\mathbf{x}^k$ refers to the associated link flow vectors.

We start by defining a mapping from a DSTAP path flow solution to a full network flow solution; this full network flow solution is said to correspond to the DSTAP solution, and the link flows (and thus travel times) on each link common to the full network and the DSTAP regional network and subnetworks will agree. This construction makes extensive use of the correspondence between artificial links (both regional and subnetwork) and the set of path segments in the full network which they aggregate. For instance, the artificial link $\theta \in \Theta_u$ connects two boundary nodes of subnetwork u, say i_u and j_u, and conceptually represents all of the used paths between these nodes ($|\theta|$ paths). In this proof, the set p_θ, as introduced in (3.23), contains all such paths connecting the endpoints of the artificial link θ. Similarly, p_γ is the set of all used paths connecting the endpoints of a subnetwork artificial link γ. Iteration superscripts are dropped for brevity.

In the construction, paths using artificial links in DSTAP must be allocated to the full network paths corresponding to these artificial links. We will allocate the artificial link flow to the full network paths in

the same proportions as the path flows in the DSTAP subnetwork aggregated by the artificial link. These proportions for an artificial regional link $\theta \in \Theta_u$, and some subnetwork path $\pi \in p_\theta$ are given by (3.24).

Now the construction of the feasible full network flow is given. Consider any OD pair $w = (r, s)$ modeled in DSTAP algorithm which is also present in the full network (note that due to artificial links, subproblems in DSTAP may have additional OD pairs not present in the full network, i.e., Θ_u and Δ_u), and any path $\pi^* \in p_w$ connecting its origin to its destination in the regional network. We consider the following cases:

Case I: r and s form a regional OD pair (no subnetwork contains them both). Then path π^* can be expressed as the concatenation of path segments π_i^* using just regional links, and artificial regional links θ_i:

$$\pi^* = \pi_0^* \oplus \theta_1 \oplus \pi_1^* \oplus \theta_2 \oplus \cdots \oplus \pi_{n(\pi^*)}$$

where π_0^* or $\pi_{n(\pi^*)}$ may be empty if π^* begins or ends with an artificial regional link. Since each artificial regional link θ_i conceptually represents $|\theta_i|$ paths between its endpoints, the decomposition of path π^* results in $|\theta_1| * |\theta_2| * \cdots |\theta_{n(\pi^*)}|$ paths in the full network. Let π be one of these paths. Path π may be expressed as:

$$\pi = \pi_0^* \oplus \sigma_1 \oplus \pi_1^* \oplus \sigma_2 \oplus \cdots \oplus \pi_{n(\pi^*)}$$

where $\sigma_i \in p_{\theta_i}$ represents one of the used paths represented by the artificial regional link θ_i. We set

$$\underline{f}_\pi = d_w \rho_{\pi^*} \prod_{i=1}^{n(\pi^*)} \rho_{\sigma_i}$$

where an empty product (meaning only regional links are used) is unity.

Case II: r and s lie within the augmented subnetwork u, but π^* uses one or more artificial subnetwork links. Then path π^* can be expressed as the concatenation of path segments π_i^* in subnetwork u, and artificial subnetwork links γ_i:

$$\pi^* = \pi_0^* \oplus \gamma_1 \oplus \pi_1^* \oplus \gamma_2 \oplus \cdots \oplus \pi_{n(\pi^*)}$$

where π_0^* and $\pi_{n(\pi^*)}$ may be empty if π^* starts or ends with a segment outside of u. Similar to Case I, the decomposition of path π^* results in $|\gamma_1| * |\gamma_2| * \cdots |\gamma_{n(\pi^*)}|$ paths in the full network. Let

π be one of these paths. Path π may be expressed as:

$$\pi = \pi_0^* \oplus \sigma_1 \oplus \pi_1^* \oplus \sigma_2 \oplus \cdots \oplus \pi_{n(\pi^*)}$$

where $\sigma_i \in p_{\gamma_i}$ represents one of the used paths represented by the artificial subnetwork link γ_i. We set

$$\underline{f}_{\pi} = d_w \rho_{\pi^*} \prod_{i=1}^{n(\pi^*)} \rho_{\sigma_i}$$

Case III: r and s lie within the augmented subnetwork u, and π^* does not use any artificial subnetwrk link. Then path π^* exists in the subnetwork u of the full network as well, and we choose $\underline{f}_{\pi^*} = f_{\pi^*}$. Note that this formula can be seen as a special case of formula in Case II, where the empty product is equal to unity.

Lemma 2. *(Mapping from DSTAP flow to full network flow.) Given any regional and subnetwork path flow vector $\mathbf{f}^k$ which is feasible in the DSTAP algorithm, the corresponding full network path flow vector $\underline{\mathbf{f}}^k$ is feasible to the full network assignment problem*

Proof. Consider any regional OD pair w. Due to the way $\underline{\mathbf{f}}$ was constructed, each path $\pi \in p_w$ with positive flow in the full network is associated with a unique path π^* in the regional network which alternates between path segments using regional links, and artificial regional links θ_i for $i \in \{1, \ldots, n(\pi^*)\}$. We will use the notation $\pi \in \pi^*$ to indicate that path π in the full network is associated with path π^* in the regional network; this notation is meant to suggest that several paths π may map to the same π^*. We then have

$$\sum_{\pi \in p_w} \underline{f}_{\pi} = \sum_{\pi^* \in p_{w,R}} \sum_{\pi \in \pi^*} \underline{f}_{\pi} \tag{3.29}$$

$$= d_w \sum_{\pi^* \in p_{w,R}} \rho_{\pi^*} \sum_{\pi \in \pi^*} \prod_{i=1}^{n(\pi^*)} \rho_{\sigma_i} \,. \tag{3.30}$$

Now, for each regional path π^*, we have $\sum_{\pi \in \pi^*} \prod_{i=1}^{n(\pi^*)} \rho_{\sigma_i} = 1$: this is trivially true when $n(\pi^*) = 0$; and if it is true whenever $n(\pi^*) = m$, then for any path with $n(\pi^*) = m + 1$ we have

$$\sum_{\pi \in \pi^*} \prod_{i=1}^{m+1} \rho_{\sigma_i} = \sum_{\pi' \in p_{\sigma_{m+1}}} \rho_{\pi'} \sum_{\pi \in \pi''} \prod_{i=1}^{m} \rho_{\sigma_i}$$

where π'' is the subpath of π^* preceding θ_{m+1}. This identity follows from the distributive law, and the claim then follows from the induction hypothesis and the definition of $\rho_{\sigma_{m+1}}$.

Applying this result to (3.30), and since $\sum_{\pi^* \in p_{v,R}} \rho_{\pi^*} = 1$ by the definition of ρ_{π^*}, the constructed flow vector satisfies the demand constraints. Since it is clearly nonnegative, it thus is feasible to the full network equilibrium problem.

The proof for subnetwork flows is identical. □

To illustrate this mapping, assume that the internal path $\pi_a^* = \{2, 3, 4\}$ and the external path $\pi_b^* = \{2, 5, 7, 4\}$ are used paths for OD pair 2-4 in subproblem 1 of Figure 3.5. Based on the previous discussion, path π_a^* belongs to Case III, and π_b^* is of type of Case II with $\pi_0^* = (2, 5)$, $\gamma_1 = (5, 7)$, and π_1^*=(7,4). Let's assume that the artificial link $(5, 7)$ in subnetwork 1 represents internal paths $\sigma_{1,1} = \{8, 10, 11, 12, 9\}$ and $\sigma_{1,2} = \{8, 12, 9\}$ in subproblem 2. In the constructed solution to the full network (Figure 3.1), OD pair 2-4 will use 3 paths: internal path $\pi_1 = \{2, 3, 4\}$ copied directly from the subnetwork; and paths $\pi_2 = \{2, 5, 8, 10, 11, 12, 9, 7, 4\}$ and $\pi_3 = \{2, 5, 8, 12, 9, 7, 4\}$ obtained by splicing the paths represented by the artificial link $(5, 7)$ into the external path $\{2, 5, 7, 4\}$. Based on the previous notation we have $\pi_1 \in \pi_a^*$ and $\pi_2, \pi_3 \in \pi_b^*$. The flow values are given by:

$$\begin{aligned}
\underline{f}_{\pi_1} &= f_{\pi_a^*} \\
\underline{f}_{\pi_2} &= d_{24} \frac{f_{\pi_b^*}}{f_{\pi_a^*} + f_{\pi_b^*}} \frac{f_{\sigma_{1,1}}}{f_{\sigma_{1,1}} + f_{\sigma_{1,2}}} \\
&= f_{\pi_b^*} \frac{f_{\sigma_{1,1}}}{f_{\sigma_{1,1}} + f_{\sigma_{1,2}}}
\end{aligned}$$

where the third equality follows since $d_{24} = f_{\pi_a^*} + f_{\pi_b^*}$. Similarly

$$\underline{f}_{\pi_3} = f_{\pi_b^*} \frac{f_{\sigma_{1,2}}}{f_{\sigma_{1,1}} + f_{\sigma_{1,2}}}$$

Let $\tilde{B} = \sum_{u \in U} |B_u|$ denote the total number of boundary points across all subnetworks. We now show that near-equilibria for DSTAP solutions map to near-equilibria in the full network, where the excess gap in the full network is at most a constant multiple of the DSTAP gap.

Lemma 3. *(Near-equilibria in DSTAP are near-equilibria in the full network.) If the maximum excess cost for the regional network and all subnetworks are respectively less than ϵ_{od}^r and ϵ_{od}^s when the "DSTAP converged?" step is reached (i.e., just before solving the master problem), then the maximum excess cost in a corresponding full network path flow solution (see Lemma 2) is at most $2\tilde{B}(\epsilon_{od}^r + \epsilon_{od}^s)$.*

Proof. Let the full network solution $\underline{\mathbf{f}}^k$ be generated from the feasible DSTAP solution $\mathbf{f}^k$ using the procedure in Lemma 2. Choose any OD pair w and used path $\pi \in p_w$ in the full network, and let η be a shortest path for this OD pair in the full network. As in Lemma 2, if w is a regional OD pair, both π and η can be decomposed into an alternating set of path segments composed of regional links, and links from a particular subnetwork:

$$\pi = \pi_0 \oplus \sigma_1 \oplus \pi_1 \oplus \cdots \oplus \sigma_n \oplus \pi_{n(\pi)} \tag{3.31}$$

$$\eta = \eta_0 \oplus \varsigma_1 \oplus \eta_1 \oplus \cdots \oplus \varsigma_m \oplus \eta_{m(\eta)} \tag{3.32}$$

where each σ_i or ς_j correspond to some artificial regional link θ_i or ϑ_j. Thus, π and η in the full network correspond to the following paths π^* and η^* in the regional network:

$$\pi^* = \pi_0 \oplus \theta_1 \oplus \pi_1 \oplus \cdots \oplus \theta_n \oplus \pi_{n(\pi)} \tag{3.33}$$

$$\eta^* = \eta_0 \oplus \vartheta_1 \oplus \eta_1 \oplus \cdots \oplus \vartheta_m \oplus \eta_{m(\eta)} \tag{3.34}$$

Using T to denote the travel time on a path, since the maximum excess cost in the regional network is less than ϵ^r_{od} we have:

$$|T_{\pi^*} - T_{\eta^*}| \leq \epsilon^r_{od} \tag{3.35}$$

The cost functions on the artificial regional links θ_i and ϑ_j are given as the flow-weighted average of the subnetwork path costs (see Section 3.5.4). This implies that $|T_{\sigma_i} - T_{\theta_i}| \leq \epsilon^s_{od}$ and $|T_{\varsigma_j} - T_{\vartheta j}| \leq \epsilon^s_{od}$. Since the number of subnetwork path segments going through each subnetwork u is at most $\frac{1}{2}|B_u|$ (otherwise there will be cycle), the value of $n(\pi)$ is bounded by $\frac{1}{2}\tilde{B}$. Using this bound we may write:

$$|T_\pi - T_{\pi^*}| \leq \frac{1}{2}\tilde{B}\epsilon^s_{od}\,,$$
$$|T_\eta - T_{\eta^*}| \leq \frac{1}{2}\tilde{B}\epsilon^s_{od}\,,$$

From (3.35) we get the bound $\epsilon^r_{od} + \tilde{B}\epsilon^s_{od}$ on $|T_\pi - T_\eta|$.

If w is a subnetwork OD pair in subnetwork u, then we have

$$|T_{\pi^*} - T_{\eta^*}| \leq \epsilon^s_{od} \tag{3.36}$$

Furthermore, the number of external subpaths on any path π^* or η^* can be at most $\frac{1}{2}|B_u|$. These external subpaths in total can visit at most $\frac{1}{2}\tilde{B}$ subpaths in other subnetworks and include at most $\tilde{B}$ regional

subpaths (at most 2 regional subpaths for any subpath is a different subnetwork). This implies that:

$$|T_\pi - T_{\pi^*}| \leq \frac{1}{2}\tilde{B}\epsilon^s_{od} + \tilde{B}\epsilon^r_{od}\,,$$
$$|T_\eta - T_{\eta^*}| \leq \frac{1}{2}\tilde{B}\epsilon^s_{od} + \tilde{B}\epsilon^r_{od}\,,$$

From (3.36) we get the bound $2\tilde{B}\epsilon^r_{od} + \epsilon^s_{od}(1+\tilde{B}) \leq 2\tilde{B}(\epsilon^r_{od} + \epsilon^s_{od})$ on $|T_\pi - T_\eta|$ which dominates the previous bound $\epsilon^r_{od} + \tilde{B}\epsilon^s_{od}$. □

In practice this bound is very loose. In the numerical experiments which follow, the maximum excess cost of the corresponding full network solution is always less than $4(\epsilon^r_{od} + \epsilon^s_{od})$, and with virtually any reasonable subnetwork definition will have the upper bound of $2|U|(\epsilon^r_{od} + \epsilon^s_{od})$, since most paths will not re-use subnetworks. Nevertheless, this result shows that an arbitrary level of convergence in the full network can be obtained by solving DSTAP to a pre-defined level of convergence depending only on the network topology and desired full network convergence. Together with a proof of DSTAP convergence, in the following subsection, this result guarantees that DSTAP can produce full network solutions with arbitrarily small gap. We summarize the discussion in this subsection in the following result.

Theorem 4. *If the stopping condition of the DSTAP algorithm is reached at some iteration k (that is, the maximum excess cost in the regional network and all subnetworks is less than ϵ^*_{od}), one can construct a path flow solution in the full network with maximum excess cost no more than $4\tilde{B}\epsilon^*_{od}$.*

3.6.2 Convergence of the DSTAP Method

The previous subsection showed that *if* the DSTAP algorithm converges to a stable solution, this solution is an equilibrium solution on the full network to within any stated tolerance. This subsection addresses the question of whether in fact it converges. First we present a formal algorithmic description of DSTAP, and then prove the convergence of the algorithm using the global convergence theorem of Zangwill [1969].

Algorithmic Description: Define the state of the algorithm by a tuple containing the regional and subnetwork link flows, regional and subnetwork path flows, original network link travel times and derivatives, subnetwork demands, and the constants in the regional and subnetwork artificial link performance functions:

$$\chi = (\mathbf{f}_r, \mathbf{f}_s, \mathbf{d}, \boldsymbol{\mu}, \boldsymbol{\lambda}) \tag{3.37}$$

where $\mathbf{f}_r$ and $\mathbf{f}_s$ are subvectors of $\mathbf{f}$ defining the regional and subnetwork path flow vectors, i.e., $\mathbf{f} = [\mathbf{f}_r; \mathbf{f}_s]$.

Each state χ is an element of the compact, nonempty set X defined by the following constraints. Let $D = \sum_{w \in W} d_w$ be the total travel demand in the full network, $\bar{t} = \max_{a \in A} t_a(D)$ an upper bound on the travel time on each link, and $\overline{T} = |A|\bar{t}$ an upper bound on the travel time on a full network path.

1. For each $\theta \in \Theta$, $\mu_\theta \in [0, \overline{T}]$.
2. For each $u \in U$ and $\gamma \in \Gamma_u$, $\lambda_\gamma \in [0, \overline{T}]$.
3. For each $u \in U$ and $w \in W_u \cup \Theta_u \cup \Delta_u$, $d_w \in [0, D]$.
4. Each path flow component in $\mathbf{f}$ lies in the interval $[0, D]$.

Note that the state χ uniquely determines the flow, and thus the travel time, on each link and path in the *full network*, through the correspondence in Lemma 2. Furthermore, these mappings from χ to $\mathbf{x}$, $\mathbf{t}$ and $\mathbf{T}$ are differentiable. We use notation such as $\mathbf{x}(\chi)$ to reflect this dependence, but at times omit explicit dependence on χ for notational brevity.

The DSTAP algorithm, given by the point-to-set map $\Xi : X \rightrightarrows X$, is the composition of several sub-mappings, which are precisely specified below:

1. Updating artificial link parameters for the regional network, denoted by $\Xi_{r,\text{links}}$.
2. Shifting flow between paths in the regional network, denoted by $\Xi_{r,\text{shift}}$
3. Updating artificial link parameters for all subnetworks, denoted by $\Xi_{s,\text{links}}$.
4. Shifting flow between paths in all subnetworks, denoted by $\Xi_{s,\text{shift}}$.

The overall DSTAP mapping Ξ is the composition of these mappings for the regional network and each of the $|U|$ subnetworks:

$$\Xi = \Xi_{s,\text{shift}} \circ \Xi_{s,\text{links}} \circ \Xi_{r,\text{shift}} \circ \Xi_{r,\text{links}} \,. \tag{3.38}$$

The four mappings are specified below.

The map $\Xi_{r,\text{links}}$ updates the artificial link parameters in the regional network, and only changes the $\boldsymbol{\mu}$ components, that is,

$$\Xi_{r,\text{links}}(\mathbf{f}_r, \mathbf{f}_s, \mathbf{d}, \boldsymbol{\mu}, \boldsymbol{\lambda}) = (\mathbf{f}_r, \mathbf{f}_s, \mathbf{d}, \boldsymbol{\mu}', \boldsymbol{\lambda}) \tag{3.39}$$

where for each artificial regional link θ, μ'_θ is calculated as the weighted average cost of travel between the endpoints of θ (weighted by the path flows in the relevant subnetwork), as described in Section 3.5.4. That is, $\mu'_\theta = \left(\sum_{\pi \in p_\theta} f_\pi T_\pi\right) / \left(\sum_{\pi \in p_\theta} f_\pi\right)$.

The map $\Xi_{r,\text{shift}}$ shifts flow from every used path in the regional network to a shortest path for its OD pair. It is defined in terms of several simpler mappings. For OD pair $w \in W_r$, let b_w be a shortest path connecting this OD pair for the travel times $\mathbf{T}(\mathbf{f})$, B_w the set of all such shortest paths, $B = \times_{w \in W_r} B_w$ the set representing every possible choice of a single shortest path for each OD pair as a $|W_r|$-dimensional vector, and $\mathbf{b} \in B$ one choice of such a vector. The map for shifting flow to these paths with step size α, as formulated in (3.20) and (3.21), is given by

$$\Xi^{\mathbf{b},\alpha}_{r,\text{shift}}(\mathbf{f}_r, \mathbf{f}_s, \mathbf{d}, \boldsymbol{\mu}, \boldsymbol{\lambda}) = (\mathbf{f}'_r, \mathbf{f}_s, \mathbf{d}', \boldsymbol{\mu}, \boldsymbol{\lambda}) \tag{3.40}$$

where $\mathbf{f}'_r = \mathbf{f}_r + \Delta^{\mathbf{b},\alpha}\mathbf{f}_r$ is the regional path flow shift given by:

$$\Delta^{\mathbf{b},\alpha} f_\pi = -\min\{f_\pi, \alpha(T_\pi - T_{b(\pi)})\} \tag{3.41}$$

if π is not the shortest path for its OD pair w chosen in $\mathbf{b}$ (that is, $\pi \neq b(\pi)$), and

$$\Delta^{\mathbf{b},\alpha} f_{b_w} = -\sum_{\pi \in p_w \setminus \{b_w\}} \Delta^{\mathbf{b},\alpha} f_\pi \tag{3.42}$$

if it is; and $\mathbf{d}'$ is obtained from $\mathbf{f}'_r$ and $\mathbf{f}_s$ by (3.13).

Lemma 5. *Let $z^{\mathbf{b}}(\alpha)$ be the value of the objective function in the full network (1) based on the regional path flows $\mathbf{f}_r + \Delta^{\mathbf{b},\alpha}\mathbf{f}_r$, using the map in Lemma 2. As a function of α, the derivative $\frac{dz^{\mathbf{b}}}{d\alpha}$ exists and is nonnegative at $\alpha = 0$. Furthermore, $\frac{dz^{\mathbf{b}}}{d\alpha} < 0$ unless $\mathbf{f}$ corresponds to a full network user equilibrium solution.*

Proof. For purposes of calculating the derivative at zero, we may restrict attention to α small enough that $\alpha(T_\pi - T_{b(\pi)}) < f_\pi$ for all paths with positive flow, so that $\Delta^{\mathbf{b},\alpha} f_\pi = \alpha(T_\pi - T_{b(\pi)})$ if $f_\pi > 0$, and $\Delta^{\mathbf{b},\alpha} f_\pi = 0$ if $f_\pi = 0$ and $\pi \neq b(\pi)$. Within this neighborhood, $\Delta^{\mathbf{b},\alpha}\mathbf{f}_r$ is differentiable, and this differentiability carries through the map in Lemma 2 to the full network path flows, and thus the original network objective function.

Consider next the effect of a change in the flow f_π on a regional path on the full network objective function (3.1). Using the notation $\eta \in \pi$ as in Lemma 2, to refer to a full network path η corresponding

to regional path π, and using ρ_η to reflect the proportion of the flow on π which uses η, we have

$$\begin{aligned}\frac{\partial z^{\mathbf{b}}}{\partial f_\pi} &= \sum_{a\in A} t_a(x_a)\frac{\partial x_a}{\partial f_\pi} \\ &= \sum_{a\in A} t_a(x_a)\sum_{\eta\in\pi}\rho_\eta\delta_{a\eta} \\ &= \sum_{\eta\in\pi}\rho_\eta\sum_{a\in A} t_a(x_a)\delta_{a\eta} \\ &= \sum_{\eta\in\pi}\rho_\eta T_\eta\end{aligned}$$

But this is the same travel time as T_π in the regional network, because the mapping $\Xi_{r,\text{links}}$ sets the travel time on artificial regional links to the flow-weighted average of the constituent paths in the original network.

Hence,

$$\begin{aligned}\frac{dz^{\mathbf{b}}}{d\alpha} &= \sum_{w\in W_r}\sum_{\pi\in p_w}\frac{\partial z^{\mathbf{b}}}{\partial f_\pi}\frac{df_\pi}{d\alpha} \\ &= -\sum_{w\in W_r}\sum_{\pi\in p_w:\pi\notin b(\pi)}(T_\pi - T_{b(\pi)})^2 \\ &\leq 0\end{aligned}$$

Furthermore, the inequality is strict unless $T_\pi = T_{b(\pi)}$ for all paths π with positive flow, that is, unless the path flows $\mathbf{f}$ satisfy user equilibrium in the regional network (and thus in the full network). □

Now, let $z^{\mathbf{b}}(\alpha)$ denote the value of the objective (3.1) after the step $\Delta^{\mathbf{b},\alpha}$ is made. As shown in Lemma 5, the derivative $\frac{dz^{\mathbf{b}}}{d\alpha}(0)$ exists and is strictly negative unless χ is a full network equilibrium. Therefore, there is some $\overline{\alpha}(\mathbf{b},\chi)$ such that $z^{\mathbf{b}}(\alpha) < z^{\mathbf{b}}(0)$ whenever $0 < \alpha < \overline{\alpha}(\mathbf{b},\chi)$. Choose the constant $\epsilon_\alpha \in (0, \frac{1}{2})$, and define the map

$$\Xi^{\mathbf{b}}_{r,\text{shift}} = (\mathbf{f}_r, \mathbf{f}_s, \mathbf{d}, \boldsymbol{\mu}, \boldsymbol{\lambda}) = (\mathbf{f}'_r, \mathbf{f}_s, \mathbf{d}', \boldsymbol{\mu}, \boldsymbol{\lambda}) \tag{3.43}$$

where $\mathbf{f}'_r = \mathbf{f}_r + \Delta^{\mathbf{b},\alpha}\mathbf{f}_r$ for some $\alpha \in [\epsilon_\alpha\overline{\alpha}(\mathbf{b},\chi), (1-\epsilon_\alpha)\overline{\alpha}(\mathbf{b},\chi)]$. The key points of this map are that both the path shift and descent of the objective are bounded away from zero for nonequilibrium χ.

Finally, the choice of the shortest path vector b is arbitrary, and

$$\Xi_{r,\text{shift}} = \bigcup_{\mathbf{b} \in B} \Xi^{\mathbf{b}}_{r,\text{shift}} \,. \tag{3.44}$$

The maps $\Xi_{s,\text{links}}$ and $\Xi_{s,\text{shift}}$ for the subnetworks are defined in analogous ways, and their specification is omitted for brevity. Note that the subnetwork shifts are all based on the same path flow values from the last regional iteration, justifying a parallel solution. Lemma 5 can be applied individually to each subnetwork's shifts, and thus to the overall map $\Xi_{s,\text{shift}}$ by linearity of the derivative.

Proof of Convergence: We now show convergence of the DSTAP algorithm by appealing to the global convergence theorem of Zangwill [1969], stated below, which uses the following definition of a closed map. For more details, see Bazaraa et al. [2013].

Definition Let $\mathbf{A} : X \rightrightarrows X$ be a point-to-set map. The map Ξ is closed at χ^* if for any sequences χ^k and υ^k satisfying:

$$\chi^k \to \chi^* \tag{3.45}$$

$$\upsilon^k \to \upsilon^* \tag{3.46}$$

with $\upsilon^k \in \Xi(\chi^k)$ we have $\upsilon^* \in \Xi(\chi^*)$.

Theorem 6. *(Global Convergence Theorem, Zangwill, 1969*, p. 91) Let $\Xi : X \rightrightarrows X$ be a point-to-set map. Suppose $\chi^0 \in X$ and a sequence $\{\chi^k\}_{k=1}^{\infty}$ satisfying $\chi^{k+1} \in \Xi(\chi^k)$ are given. Let a solution set $X^* \subset X$ be given, and furthermore suppose that:*

1. *The set X is compact.*
2. *There is a continuous function z on X such that*
   - *(a) if $\chi \notin X^*$, then $z(\upsilon) < z(\chi)$ for all $\upsilon \in \Xi(\chi)$.*
   - *(b) if $\chi \in X^*$, then $z(\upsilon) \leq z(\chi)$ for all $\upsilon \in \Xi(\chi)$.*
3. *The map Ξ is closed at all points outside X^*.*

Then every limit point of $\{\chi^k\}$ belongs to the solution set X^*.

We define X^* to be the set of χ corresponding to user equilibrium solutions in the full network (that is, minimizers of (3.1)) under the mapping in Lemma 2, and check each of the conditions of Zangwill's Theorem in turn. The set X is clearly compact, and the full network objective function z is clearly continuous in Ξ (depending only on $\mathbf{f}_r$ and $\mathbf{f}_s$). We now show its descent property. It is enough to show this property for the $\Xi_{\cdot,\text{shift}}$ mappings, since the $\Xi_{\cdot,\text{links}}$ maps do not change the link flows and thus leave z unchanged. Lemma 7 shows this for the regional network shift $\Xi_{r,\text{shift}}$, the proof for the subnetwork shifts is identical but for a change in notation.

Lemma 7. *Let $\upsilon \in \Xi_{r,\text{shift}}(\chi)$. Then $z(\upsilon) \leq z(\chi)$, with equality only occurring if $\chi \in X^*$.*

Proof. If $\chi \in X^*$, then it corresponds to a user equilibrium solution in the full network, in which case $\Delta^{\mathbf{b},\lambda}\mathbf{f}_r = \mathbf{0}$, since paths are either unused and hence have zero shift based on the first term in the minimum (3.41), or have the same cost as the shortest path for their OD pair, and have zero shift based on the second term in the minimum. Hence $z(\upsilon) = z(\chi)$.

Otherwise, Lemma 5 applies, and the derivative of z for the shift $\Delta^{\mathbf{b},\lambda}\mathbf{f}$ is negative. The step size α is chosen such that $z(\upsilon) < z(\chi)$, proving the lemma. □

Lemma 8. *The maps $\Xi_{r,\text{shift}}$ and $\Xi_{s,\text{shift}}$ are closed.*

Proof. We prove the lemma for $\Xi_{r,\text{shift}}$, the proof is identical for $\Xi_{s,\text{shift}}$. Consider any convergent sequences $\chi^k \to \chi^*$ and $\upsilon^k \to \upsilon^*$ lying within X, and satisfying $\upsilon^k \in \Xi(\chi^k)$. Define the flow shift $\Delta\mathbf{f}_r^k = \mathbf{f}_r(\upsilon^k) - \mathbf{f}_r(\chi^k)$ to be the difference in the $\mathbf{f}_r$ components of υ^k and χ^k. The sequence of flow shifts converges to some vector $\Delta\mathbf{f}_r^*$. Further let $\mathbf{b}^k$ and α^k be values of $\mathbf{b}$ and α which correspond to $\Delta\mathbf{f}_r^k$. Now, by the definition of $\Xi_{r,\text{shift}}$ the vectors $\Delta\mathbf{f}_r^k$ have at most one strictly positive component for each regional OD pair v, corresponding to the choice of shortest path $b_w^{k^*}$. Therefore the limit flow shift vector $\Delta\mathbf{f}_r^*$ has at most one strictly positive component for each OD pair. For the OD pairs for which there is a strictly positive component in $\Delta\mathbf{f}_r^*$, the corresponding component in $\Delta\mathbf{f}_r^k$ must also be strictly positive for k sufficiently large. For each OD pair w with no strictly positive component in $\Delta\mathbf{f}_r^*$, there must be a shortest path b_w^* which appears infinitely often in the sequence b^k; we may therefore pass to a subsequence k' where all $\mathbf{b}^{k'}$ are identical. Denote this vector of shortest path choices as $\mathbf{b}^*$.

Within this subsequence, the values $\alpha^{k'}$ lie within the closed intervals $[\epsilon_\alpha\overline{\alpha}(\mathbf{b}^*, \chi^{k'}), (1-\epsilon_\alpha)\overline{\alpha}(\mathbf{b}^*, \chi^{k'})]$. Since $\overline{\alpha}(\mathbf{b}^*, \chi)$ is continuous in χ, we have $\overline{\alpha}(\mathbf{b}^*, \chi^{k'}) \to \overline{\alpha}(\mathbf{b}^*, \chi^*) \equiv \overline{\alpha}^*$, and a straightforward generalization of the Bolzano-Weierstrass theorem allows us to pass to a subsequence k'' where $\alpha^{k''} \to \alpha^*$ for some $\alpha^* \in [\epsilon_\alpha\overline{\alpha}^*, (1-\epsilon_\alpha)\overline{\alpha}^*]$.

Now, consider the path travel time vectors $\mathbf{T}^{k''}$, which converge to $\mathbf{T}^* \equiv \mathbf{T}(\chi^*)$ by continuity. The limit vector $\mathbf{b}^*$ must correspond to shortest paths given the travel times $\mathbf{T}^*$: if not, then $\mathbf{b}^{k''}$ must also contain a non-shortest path when k'' is large. But this contradicts the choice of subsequence ($\mathbf{b}^{k''} = \mathbf{b}^*$ uniformly) and the condition that $\mathbf{b}^{k''}$ corresponds to $\Xi_{r,\text{shift}}(\chi^{k''})$. Similarly, continuity gives $\overline{\alpha}(\mathbf{b}^*, \chi^*) = \overline{\alpha}^*$. Therefore $\mathbf{b}^*$ and α^* are valid choices for $\Xi_{r,\text{shift}}$ at the limit point χ^*, and $\upsilon^* \in \Xi_{r,\text{shift}}(\chi^*)$. □

Lemma 9. *The maps $\Xi_{r,\text{links}}$ and $\Xi_{s,\text{links}}$ are closed.*

Proof. Both of these maps are in fact single-valued functions which are continuous in χ, and hence closed. □

Theorem 10. *Given any sequence χ^k such that $\chi^0 \in X$ and $\chi^i \in \Xi(\chi^{i-1})$, every limit point of this sequence corresponds to a user equilibrium solution in the original network under the mapping in Lemma 2.*

Proof. We verify each condition in Zangwill's theorem. The feasible set X is clearly compact. Defining X^* to be the set of χ which map to user equilibrium solutions in the original network under Lemma 2, Lemma 7 establishes that $z(\chi^i) \leq z(\chi^{i-1})$, with strict inequality $\chi^{i-1} \in X^*$. Finally, by Lemmas 8 and 9, each of the mappings $\Xi_{r,\text{links}}$, $\Xi_{r,\text{shift}}$, $\Xi_{s,\text{links}}$, and $\Xi_{s,\text{shift}}$ is closed, and hence their composition is as well. □

3.7 Demonstrations

In this section, we study the properties of the DSTAP algorithm on the Austin, Texas network. We start by discussing issues related to the implementation of the DSTAP algorithm, and proceed to study the convergence properties and the accuracy of the solution compared to a centralized approach. The computational performance of DSTAP is discussed next. All tests are run on a 3.3 GHz Linux machine with 8 GB RAM.

The Austin network (full network) has 6349 nodes, 18696 links, 1117 zones, 231497 OD pairs and total demand of 687690 vehicles. We partition the network into 2 subnetworks: the *northern subnetwork* and *southern subnetwork*. These subnetworks are divided by the Colorado River, which flows through the city. Each subnetwork has 20 boundary nodes, and there are a total of 27 links connecting these 2 subnetworks through boundary nodes. There are no regional nodes in the proposed network decomposition. Table 3.1 illustrates the statistics for the Austin network (full network), and regional network and subnetworks introduced in the DSTAP algorithm. Note that the subnetworks are modeled without subnetwork artificial links. In our initial tests, we observed that the artificial subnetwork links were not used, indicating

Table 3.1: Statistics of the Austin network solved in centralized approach and DSTAP regional network and subnetworks.

Network	nodes	zones	OD pairs	demand	physical links	artificial links
Austin network	6349	1117	231497	687690	18696	0
Regional network	1854	907	67329	127695	27	24536
Southern subnetwork	2383	490	57557	185979	6863	176
Northern subnetwork	3966	627	106611	374016	11806	175

that internal subnetwork demands were restricted to links and routes inside the subnetworks and external subpaths were more expensive compared to internal ones. Thus, we removed all artificial subnetwork links to simplify the networks for final simulations.

3.7.1 Implementation

Although convergence of the DSTAP algorithm was shown in the previous section, efficient performance on practical networks requires further implementation choices to be made. This subsection discusses our findings based on the experiments we performed in the Austin network.

As discussed before, here we adjust the stepsize at each iteration based on the maximum excess cost value. Figure 3.8 shows the variation in stepsize over iterations when starting with an initial step size of $\alpha_0 = 1$ and adjusting it by applying the following rules: decrease the stepsize by a factor of 0.9 if the maximum excess cost increased from the previous iteration or increase it by a factor of 1.1 if the maximum excess cost improved in two consecutive iterations. In general, as seen in Figure 3.8, the stepszie has a decreasing trend and gets more stable at values less than 0.2. In our implementation of DSTAP, the best performance, in terms of convergence speed and accuracy, was obtained by choosing α_0 in the range $[0.1, 0.5]$, and we can boost the convergence rate by forcing a large stepsize, e.g. $\alpha = 1$, periodically (for example every $4 - 5$ iterations) for just one step and then reseting it back to the previous value.

The master problem starts with the flow assigned to the regional network at previous iteration, and re-equilibrates this flow based on the new artificial regional link parameters to obtain the new flow assignment. In addition, after solving the regional network at each iteration and updating the subnetwork OD demands, the subproblems need not be solved from scratch. Warm-starting the subproblems with the solution from the previous iteration, proportionally inflating or deflating the path flows for OD pairs whose demand changed, provided solutions with a good initial gap and a better convergence rate.

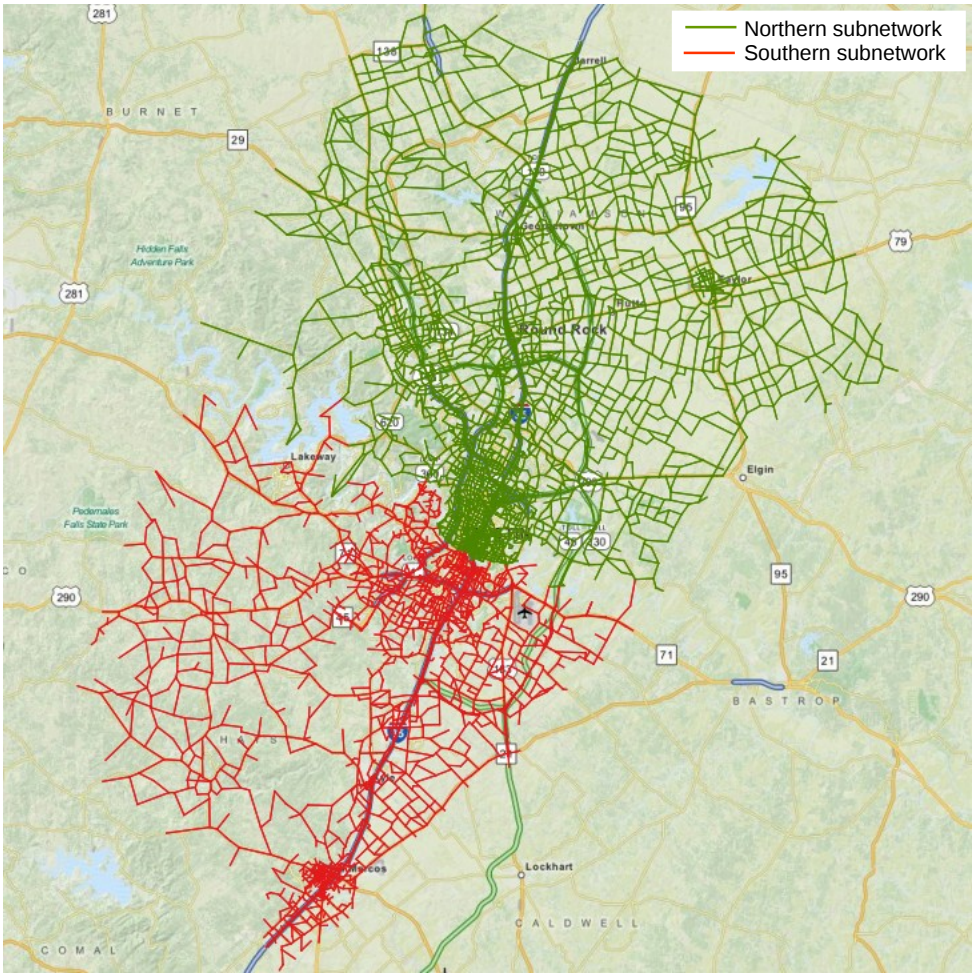


Figure 3.7: Austin network decomposed into two subnetworks: northern and southern subnetworks.

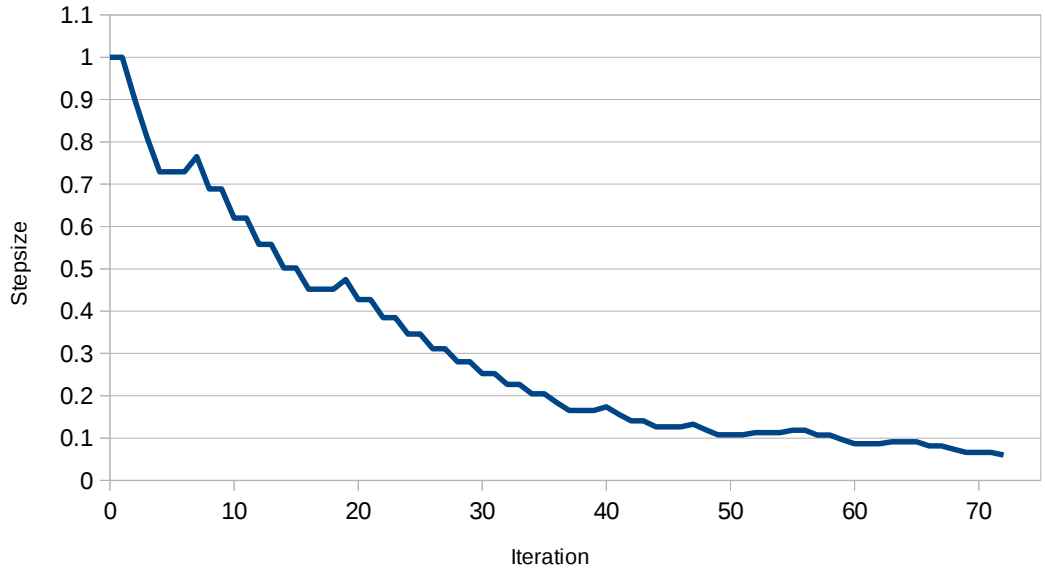


Figure 3.8: Iterative change of stepsize, α in equation (3.20), in the regional network, starting with $\alpha_0 = 1$.

3.7.2 Convergence Properties

Figure 3.9 plots the maximum excess cost values for the regional network, northern and southern subnetworks, and also the excess cost on the Austin network in a logarithmic scale for a termination criterion of 0.8 minute ("*DSTAP converged?*" block in Figure 3.6) and initial stepsieze of 0.2. The maximum excess cost for the Austin network is calculated by constructing a feasible path flow solution on the Austin network from the DSTAP path flow solution according to Lemma 2. The DSTAP algorithm converged in 90 iterations with a maximum excess cost of 0.76 minute on the regional network, and $4.28\text{E}{-4}$ minute and $1.74\text{E}{-4}$ minute on southern and northern subnetworks, respectively. The maximum excess cost on the full network is always within 10% of that of the regional network, and upon convergence, the full network has a maximum excess cost of 0.816 minute.

Figure 3.10 shows the average excess cost and relative gap values for the Austin network, regional network, and northern and southern subnetworks. For any general network u with set of OD pairs W_u, the average excess cost and relative gap measures may be defined as:

$$\text{Average excess cost} = \frac{\sum_{w \in W_u} \sum_{\pi \in p_w} f_\pi (T_\pi - T_{b_w})}{\sum_{w \in W_u} d_w} \tag{3.47}$$

$$\text{Relative gap} = \frac{\sum_{w \in W_u} \sum_{\pi \in p_w} f_\pi (T_\pi - T_{b_w})}{\sum_{w \in W_u} \sum_{\pi \in p_w} f_\pi T_\pi} \tag{3.48}$$

where T_{b_w} is cost of the shortest path for OD pair w. Upon convergence, the average excess cost and gap value of the DSTAP solution applied to Austin network were $1.16\text{E}{-6}$ and $8.74\text{E}{-11}$, respectively.

3.7.3 Correctness

To examine the accuracy of the DSTAP algorithm, we solved for equilibrium on the Austin network using the traditional gradient projection method, to a gap value of $1\text{E}{-10}$, and measured the percentage error in the equilibrium OD travel times as:

$$\mathcal{E}_t(w) = \frac{|t_D(w) - t_C(w)|}{t_C(w)}, \quad w \in W \tag{3.49}$$

where $\mathcal{E}_t(w)$ is the relative error in travel time of OD pair w, and $t_D(w)$ and $t_C(w)$ are respectively the equilibrium travel times from DSTAP and the centralized method, computed as the average travel time of all used paths at equilibrium. Figure 3.11 shows the average percentage OD travel time error $\left(\frac{\sum_{w \in W} \mathcal{E}_t(w)}{|W|} \right)$ against the iteration number of DSTAP algorithm in the final assignment. The average travel time error,

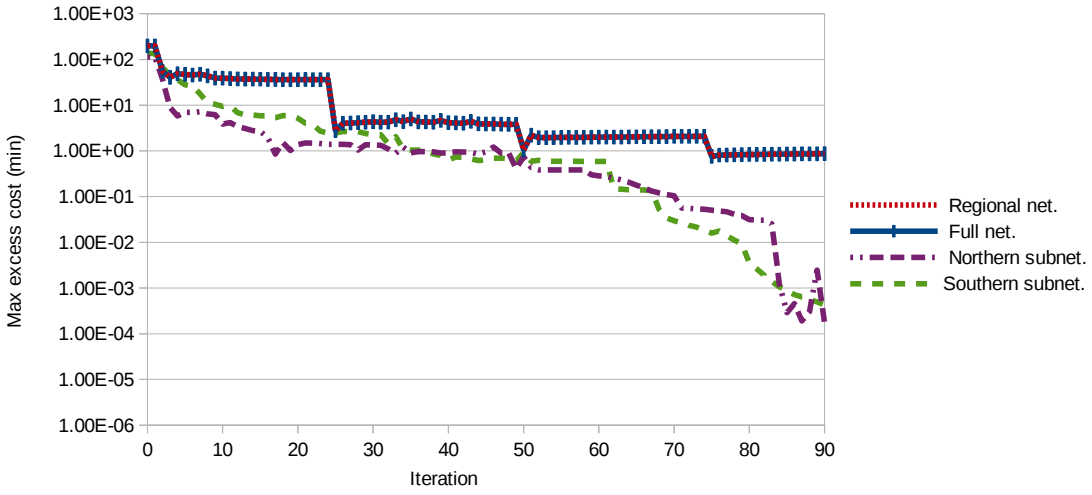


Figure 3.9: Maximum excess cost values of the regional network, northern and southern subnetworks, and the maximum excess cost of the full network (Austin network).

expressed in Figure 3.11, decreases and has a value of 0.006% upon termination. Out of 231,497 OD pairs, 231,017 have an error of less than 0.1%, and the travel time error of almost 99.8% of OD pairs is less than 1%.

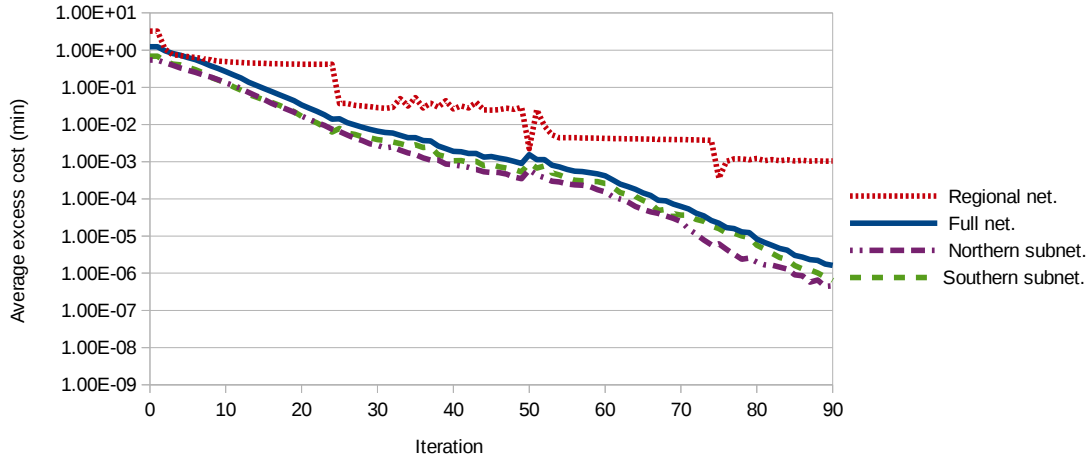
A similar measure was proposed to evaluate the accuracy of the link flows. Let $\mathcal{E}_f(a)$ denote the percentage error in flow of link a given by:

$$\mathcal{E}_f(a) = \frac{|x_D(a) - x_C(a)|}{x_C(a)} \tag{3.50}$$

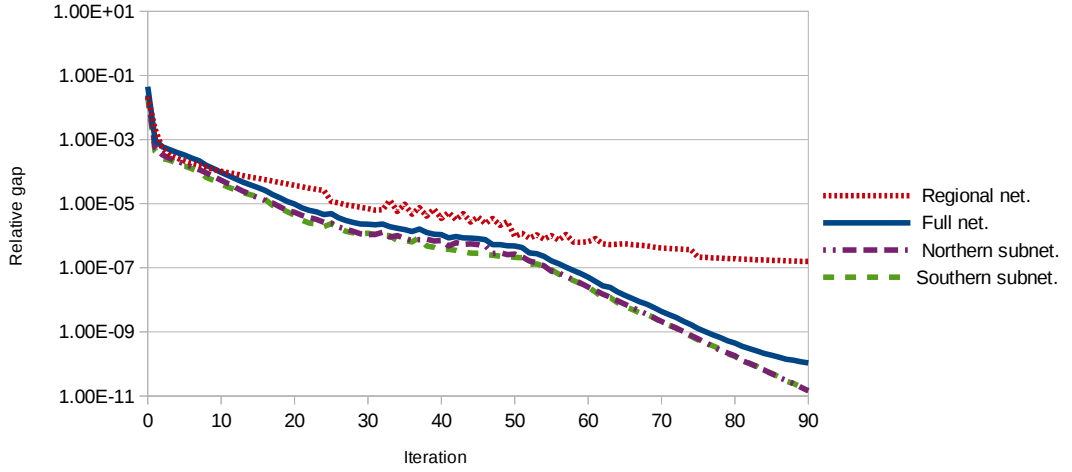
where $x_D(a)$ and $x_C(a)$ denote the flow assigned to link a in the DSTAP and centralized methods, respectively. The average link flow error is plotted in Figures 3.12. DSTAP algorithm terminates with an average link flow error of 0.067%, and more than 98.9% of links have an error less than 1%.

3.7.4 Computational Effort

This section investigates the computational requirements of the DSTAP algorithm, compared with the centralized approaches. Here we used relative gap as the measure of convergence, and both the DSTAP and centralized approaches were used to solve the network to a relative gap of 1E−5. The simulations were implemented on one machine and Thread class in Java was used to solve the subproblems simultaneously.



(a) Average excess cost



(b) Relative gap

Figure 3.10: The average excess cost (a) and relative gap (b) of the full network, regional network, and northern and southern subnetworks.

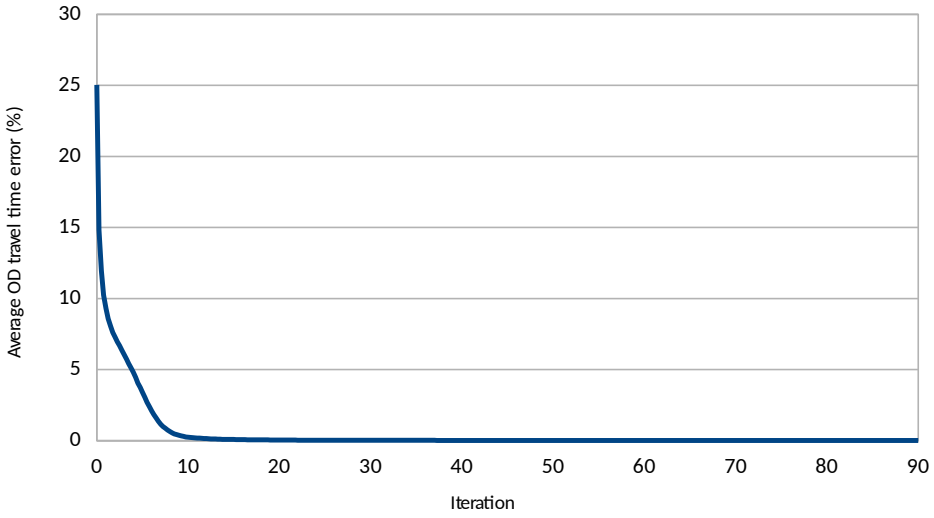


Figure 3.11: The average percentage error in OD travel times of DSTAP compared to centralized algorithm solution.

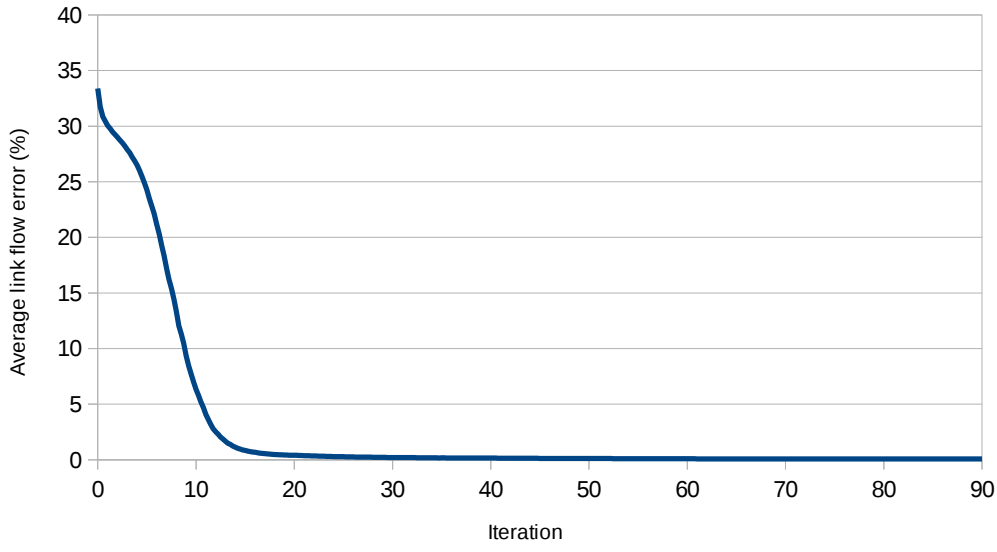


Figure 3.12: The average percentage error in flows assigned to links in DSTAP compared to centralized algorithm.

More specifically, one thread is created for each subproblem and if one of the threads ends earlier, it waits for the other thread to finish before calling the next task. The traditional, centralized approach resulted in a run time of 1780 seconds while the proposed DSTAP algorithm, with described parallel implementation, could solve the Austin network to the same level of relative gap in 1128 seconds: savings of almost 36%.

To get a broader understanding of the computational performance of DSTAP, we conducted a sensitivity analysis to the overall demand level in the network, scaling the OD matrix by factors ranging from 0.2 to 2. Figure 3.13 plots the runtime of DSTAP algorithm (regional network in red and parallelized subnetworks in green) compared to a centralized approach (black). Figure 3.13 shows that the computational savings of DSTAP are more significant in absolute terms for congested networks: more than 8500 seconds when demand is doubled. Almost independent of the demand level, roughly four times as much computational time is expended on the subproblems as on the master problem.

Figure 3.14 plots the percentage time saving for different demand levels. For low congestion cases, the saving varies between 35%–55%, increasing slightly as demand increases: the savings are almost 70% when OD demands are doubled. Also note that in our simulations, the time spent to perform the sensitivity analysis and estimate the artificial links for each subnetwork is between 15%–20% of total subnetwork computational time.

3.8 Conclusion and Discussion

We proposed a spatial decomposition approach for the traffic assignment problem. The DSTAP algorithm distributes the assignment task between the master problem, an equilibrium assignment over a simplified version of the full network, and subproblems, each solving for equilibrium on a smaller subnetwork.

Artificial regional and subnetwork links are created based on linear approximations obtained through sensitivity analysis. This is a critical component of the algorithm, because they allow for both the master problem and subproblems to anticipate the response from the other networks they interact with. Rather than having a strict separation between models, the use of "softer" boundaries was able to improve convergence of the algorithm.

The subproblems are modeled in the master problem using some artificial regional links, which are updated each iteration. The assigned regional demand to these artificial regional links are then used to update the OD demands in subproblems. This exchange of information between the master and subproblems is implemented in an abstract way to ensure an accurate and fast assignment process.

Experiments on the Austin, TX network showed the computational advantages of DSTAP, and its convergence to the correct equilibrium solution. A major question for future research is how best to divide

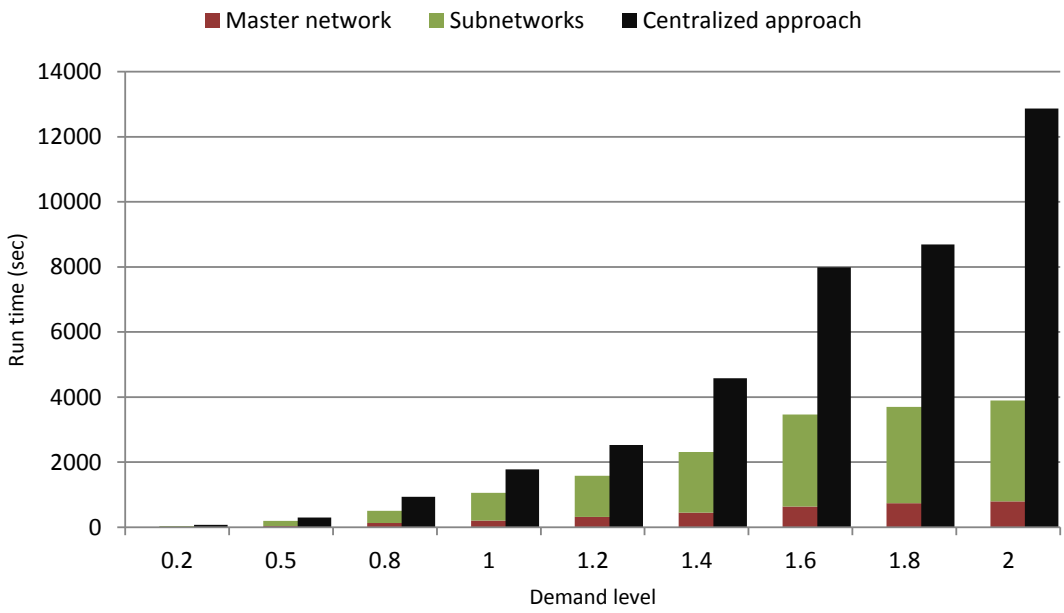


Figure 3.13: Computational time of the master (red) and subnetworks (green) in DSTAP compared to centralized run time (black) for different demand levels.

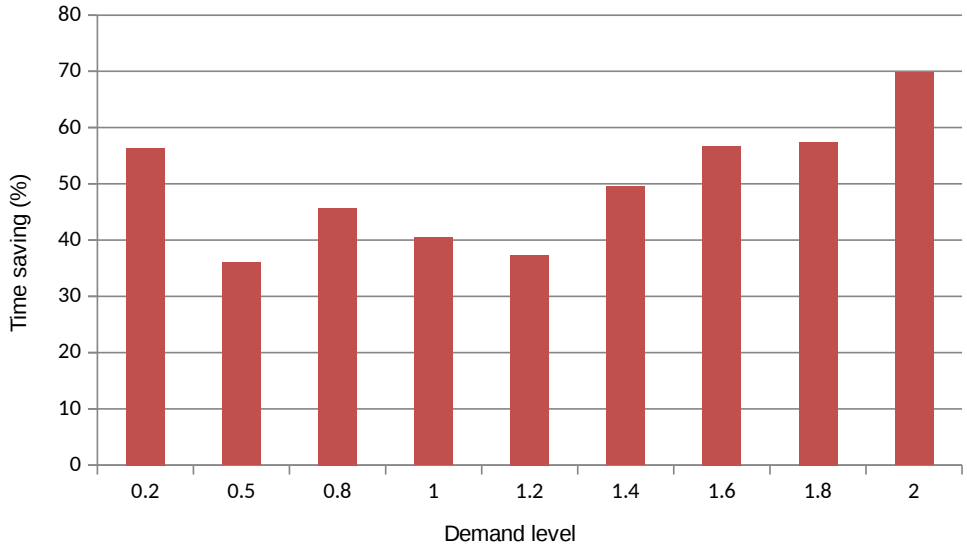


Figure 3.14: Computational savings of DSTAP algorithm.

a network into subnetworks. In the Austin network, the Colorado River provided an obvious partition into two regions of roughly equal size. It remains to determine general procedures for identifying subnetworks, and the extent to which network topology influences the computational requirements of the DSTAP algorithm.

In general, partitioning a network into more subnetworks will reduce computation time for the subproblems, but increases the complexity of the master problem by increasing the number of boundary nodes, regional OD pairs, and subnetwork interactions. Based on the detail and complexity of the proposed algorithm and our simulation experiments, we suggest the following criteria to be considered when partitioning a network.

- Number of boundary nodes: Artificial links are created between each pair of boundary nodes and also between regional origin/destination nodes and boundary nodes. As a result, partitioning a network such that the minimum number of boundary nodes are created can decrease the size of the regional network.

- Size of the regional OD matrix: Similar to the case of boundary nodes, the number of origin/destination nodes in the regional network controls the size of the master problem (number of links and OD pairs). This also increases the number of OD pairs in each subnetwork and the partition may result in subnetworks with number of OD pairs close to or even higher than that of the full network.

- Interaction between subnetworks: The subnetwork interactions are modeled with artificial links. Adding these subnetwork artificial links, in addition to increasing the size of the subproblems, requires performing additional sensitivity analysis to estimate the parameters of these artificial links. One possible way to evaluate the amount of interactions beforehand is to find shortest path tree from each subnetwork origin node in the full network. If the shortest path to any leaf node passes through more than one boundary node (this is an external path which crosses the subnetwork boundary and returns back later), then external subnetwork paths may be required. The number of OD pairs with an external shortest paths along with their associated trips may be used to weight the amount of interaction between subnetworks.

Experimental evaluation the impact of these parameters and defining specific threshold values for each of them are beyond the scope of this work, but is an important topic for future research.

Table 3.2: Glossary of terms

Full network $G = (N, A, W)$	$\triangleq$	contains all the nodes, links, and OD pairs under consideration
Subnetwork $u = (N_u, A_u, W_u)$	$\triangleq$	a subset of the full network ($N_u \subset N, A_u \subset A, W_u \subset W$)
Subnetwork nodes N_u	$\triangleq$	nodes within the boundary of subnetwork u
Boundary nodes $B_u \subseteq N_u$	$\triangleq$	subnetwork nodes which are the tail or head node of a regional link
Subnetwork links A_u	$\triangleq$	links whose tail and head nodes are in N_u
Subnetwork OD pairs W_u	$\triangleq$	OD pairs whose origin and destination are both nodes in N_u
Subnetwork demand	$\triangleq$	trips corresponding to subnetwork OD pairs
Subnetwork path	$\triangleq$	its endpoints correspond to a subnetwork OD pair
Internal subnetwork path	$\triangleq$	uses links and nodes from the same subnetwork
External subnetwork path	$\triangleq$	uses links and nodes from more than one subnetwork
Regional nodes N_r	$\triangleq$	nodes in the full network which are not part of any subnetwork
Regional links A_r	$\triangleq$	links in the full network which are not part of any subnetwork
Regional OD pairs W_r	$\triangleq$	OD pairs in the full network which are not part of any subnetwork
Regional demand	$\triangleq$	trips corresponding to regional OD pairs
Regional path	$\triangleq$	its endpoints correspond to a regional OD pair
Artificial regional links,	$\triangleq$	artificial links representing the subnetworks in regional network
Artificial subnetwork links,	$\triangleq$	artificial links added to subnetworks
Regional network $G_a = (N_a, A_a, W_a)$	$\triangleq$	abstracted version of full network augmented with artificial regional links
Centralized approach	$\triangleq$	solving the full network as one problem without regard to the subnetworks
Decomposed approach	$\triangleq$	solving the full network by partitioning the full network into subnetwork
Master problem	$\triangleq$	solves regional network
Subproblem u	$\triangleq$	solves subnetwork u augmented with artificial subnetwork links

Table 3.3: Table of notation

d_w	$\triangleq$	demand for OD pair w
p_w	$\triangleq$	set of paths connecting endpoints of for OD pair w
$\widehat{p}_w$	$\triangleq$	set of used paths excluding the shortest path for OD pair w.
T_π	$\triangleq$	travel time on path π
f_π	$\triangleq$	flow on path π
T_{b_w}	$\triangleq$	travel time on the shortest path for OD pair w
$b(\pi)$	$\triangleq$	shortest path corresponding to the OD pair whose endpoints are the same as those of π
$z(\cdot)$	$\triangleq$	value of the objective function in the full network
$x^k_{a,r}$	$\triangleq$	regional flow assigned to link a at iteration k of DSTAP
$x^k_{a,u}$	$\triangleq$	flow from subnetwork u assigned to link a at in iteration k of DSTAP
$x^k_{a,s}$	$\triangleq$	total subnetwork flow assigned to link a at iteration k of DSTAP
α	$\triangleq$	stepsize
s_π	$\triangleq$	second derivative of objective function $z(\cdot)$ with respect to path flow f_π
Θ_u	$\triangleq$	set of artificial regional links created for subnetwork u in regional network/ set of OD pairs in subnetwork u with regional demand
Γ_u	$\triangleq$	set of artificial subnetwork links added to subnetwork u
Δ_u	$\triangleq$	set of OD pairs in subnetwork u which correspond to artificial subnetwork links in other subnetwork
$\breve{d}^k_w$	$\triangleq$	adjusted subnetwork OD demand at iteration k
$t_a(\cdot)$	$\triangleq$	cost function of link a
$\tilde{t}^k_a(\cdot)$	$\triangleq$	biased cost function of link a in master problem at iteration k of DSTAP
$\breve{t}^k_a(\cdot)$	$\triangleq$	biased cost function of link a in subproblem at iteration k of DSTAP
$t^k_\theta(\cdot)/t^k_\gamma(\cdot)$	$\triangleq$	cost function of artificial regional/subnetwork link θ/γ at iteration k of DSTAP
$\mu_\theta/\lambda_\gamma$	$\triangleq$	free-flow time of cost function of artificial regional/subnetwork link θ/γ
ψ_θ/ϕ_γ	$\triangleq$	sensitivity term of cost function of artificial regional/subnetwork link θ/γ
ρ_π	$\triangleq$	proportion of flow assigned to path π
f^k_r	$\triangleq$	regional path flow vectors at iteration k of DSTAP
f^k_s	$\triangleq$	subnetwork path flow vectors at iteration k of DSTAP
$\mathrm{f}^k = [\mathrm{f}^k_r; \mathrm{f}^k_s]$	$\triangleq$	path flow vectors at iteration k of DSTAP
$\underline{\mathrm{f}}^k$	$\triangleq$	path flow vectors in the full network at the end of iteration k of DSTAP
x^k	$\triangleq$	link flow vectors at iteration k of DSTAP

Part II:
Network Design: A Distributed Problem Solving Approach

4

Network Design Problem: A Decentralized Approach

The network design problem is concerned with making investment decisions to maximize a system objective function subject to budget and feasibility constraints. Network design can be formulated as a bi-level problem where the system manager selects the design parameters, and users react by modifying their trip characteristics such as destination, mode, and route. These problems are hard to solve and a distributed problem solving approach can be used to develop an efficient framework for scaling these problems.

In this chapter, we develop a distributed algorithm for network design problem in which different planning agencies may have different objective functions and priorities while a regional agent (state or federal officials) decides about the finding allocation between different urban cities. Under the proposed *allocation-design* problem, the urban planning agencies do their own network planning and design independently while taking into account that their local plan and investments may have broader impact than their subnetwork jurisdiction. The regional agent has limited and indirect authorities over the subnetworks through budget allocation. We develop a solution algorithm based on a *sensitivity-analysis* heuristic to solve this problem and test our algorithm on two case studies: a hypothetical network composed of two copies of the Sioux Falls network, and a modified version of the Austin regional network. We evaluate the correctness of the decentralized algorithm and discuss a condition under which the decentralized algo-

rithm replicates the true solution. Simulation results also reveals the advantage of the proposed algorithm in this chapter in modeling the interactions between different regions. Furthermore, the implementation of distributed algorithm on Austin regional network demonstrates a computational saving of 22%.

4.1 Introduction

This chapter is concerned with formulating and solving network design problem (NDP) over a regional network composed of several urban networks together with intercity links and nodes. We propose a new implementation of the problem based on th idea of distributed problem solving. Distributed problem solving recommends partitioning the problem into smaller problems, called *subproblems (subtasks)*, and introducing multiple solvers, referred to as *local solvers (agents)*, to deal with the problem. In such a system, there is no central processor or central controller and tasks are divided between the local solvers. The local solvers, working on the subproblems, have limited access to local information, mainly from the assigned task, and none of them is equipped with global information or knowledge. The subproblems must be *cooperative* in the sense that, due to lack of sufficient information, a mechanism should be implemented to share information between them. In addition, the local solvers should be *loosely coupled*: the local solvers spend most of their time on solving the assigned task rather than communicating with other solvers [Davis and Smith, 1983].

In the proposed distributed implementation, regional agent is responsible for managing the intercity roadways and has no authority over urban cities. The urban cities are managed independently and the exact design plan at each urban city is not known to the regional agent. The regional agent has a fixed budget which can be distributed between urban cities and invested on intercity roadways. The proposed *allocation-design problem* can also be viewed as a multiresolution network design problem in which the regional agent deals with a less detailed network in a higher level while the urban cities are managed in lower levels but higher details.

There have been many studies on NDPs, and numerous heuristic and exact algorithms have been developed to solve variations of the problem. The proposed network design algorithm in this chapter, in addition to suggesting a new algorithm for traditional bi-level NDPs, also proposes a solution algorithm for multi-level design problems. In practice, urban regions often implement their planning projects without considering the impact of their local plans on a larger scale. This happens mainly because internal concerns usually have higher priority compared to system-level effects. The proposed decentralized scheme for NDP in this chapter can alleviate this problem by efficiently modeling the linkage between different players (local and regional planing agencies).

The network design problem is a special class of bi-level optimization which is composed of two interconnected problems: the upper-level and the lower-level optimization task. The upper-level problem is usually concerned with design variables (toll values, capacity improvements, facility location, signal timing plans, link and lane additions, etc.), and the lower-level problem captures the reaction of users in response to design decisions. The general formulation of a bi-level program is:

$$\min_{y} f_u(y, x(y)) \tag{4.1}$$

$$s.t. \qquad (y, x(y)) \in S_u \tag{4.2}$$

where $S_u = \{(y, x) : g_u(y, x) \leq 0, h_u(y, x) = 0\}$ defines the feasible region for the upper-level problem, and $x(y)$ is well-defined in our model since the lower-level problem will have a unique solution. The value of $x(y)$ is defined implicitly to be the solution to the following lower-level problem:

$$\min_{x} f_l(x, y) \tag{4.3}$$

$$s.t. \qquad x \in S_l(y) \tag{4.4}$$

where $S_l(y) = \{x : g_l(x, y) \leq 0, h_l(x, y) = 0\}$ is the feasible set for the lower-level problem.

Bilevel programming problems (BLPP) have been discussed and used in the economic field, especially in applications related to the Stackelberg game [Von Stackelberg, 1952]. In a two-person Stackelberg game, the players wish to minimize their own cost (maximize their utility). The first player, referred to as the leader, has perfect information about the objective function of the second player, the follower, and chooses the strategy anticipating the reaction of the follower. The follower may or may not know the objective of the leader but is aware of the strategy taken by the leader and updates their strategy using this information.

Falk and Liu [1995] divided algorithms for dealing with a general BLPP into three classes. In the first class of algorithms, both the upper-level and lower-level problem are approximated by unconstrained minimization problems of penalized augmented objective functions. These solution algorithms are also called double-penalty methods [Ishizuka and Aiyoshi, 1992, Dempe et al., 2015] and suffer from slow convergence rate. The second class of solution algorithms replaces the lower-level problem with the equivalent Karush-Kuhn-Tucker (KKT) conditions [Edmunds and Bard, 1991, Bard, 1983]. This transforms the bi-level problem into a single-level mathematical program with complementary slackness conditions. The resulting problem can be very complicated and consequently hard to solve. The third class of algorithms uses the gradient information of the lower-level problem to compute a descent direction for the upper-level problem [Kolstad and Lasdon, 1990]. The proposed algorithm described in this chapter falls into the

third category.

It is easy to see the connection between the transportation network design problem and such a Stackelberg game. In transportation network design applications, the leader is a network manager (planner) making design decisions such as which new links should be added or which set of links should be improved, and the followers are users traveling through the network. The leader knows that the followers select their route in order to minimize their travel cost (user equilibrium principle) and the followers select their route taking into account the decisions made by the leader. Predicting the reaction of the users to design decisions suggested by the system manager is critical, and without correct modeling of the users' response, the design problem may result in unexpected situations in which no one may benefit from the implemented improvement plans. A well-known example of such issue is the Braess paradox in which adding a link increases the travel time of all users [Braess, 1969].

The transportation network design problem can also be categorized as a mathematical program with equilibrium constraints (MPEC), which is a class of problems closely related to bi-level optimization. In an MPEC, some of the constraints are defied as equilibrium constraints such as complimentary constraints or variational inequalities. For the case of transportation network design problem, one can replace the lower-level problem, which is a user equilibrium (UE) problem, as complimentary constraints or the equivalent variation inequalities. Luo et al. [1996] provide more detail on this type of problem.

The transportation NDP can be divided into three classes: *discrete network design problems* (DNDP), *continuous network design problems* (CNDP), and *mixed network design problems* (MNDP). The first class refers to problems in which the design decisions are discrete variables (e.g., constructing new roads, adding new lanes, location of bus stops, turning movements at signalized intersection). The second class deals with design problems with continuous decision variables. These variables usually include how much to expand a link, how much toll should be charged on candidate links, or timing the signals. The last class of problems is a mixture of the previous classes and include both discrete and continuous design variables. Note that the main theme of the network design problem discussed in this chapter is CNDP.

The continuity of decision variables in case of CNDPs allows a variety of algorithms and methodologies to be used in order to develop efficient solution algorithms. This is the main benefit of CNDPs compared to DNDPs and MNDPs and has resulted in a larger body of research to deal with this type of problems. In the following paragraphs we provide a summary of the current studies on the CNDPs. The interested reader is referred to Gao et al. [2005], Luathep et al. [2011], Farahani et al. [2013], and Wang et al. [2013] for more detail on discrete and mixed network design problems.

To the best of our knowledge, Abdulaal and LeBlanc [1979] were the first to formulate and solve the CNDP in the field of transportation science. The authors formulated the problem as a nonlinear

unconstrained optimization and solved it using the methods developed in Powell [1964] and Hooke and Jeeves [1961]. Since then there has been a substantial amount of research on the CNDP [Yang and Bell, 2001]. Upper-level objectives have included system travel time [Mathew and Sharma, 2009, Meng et al., 2001], consumers' surplus [Yang, 1997], construction cost [Friesz et al., 1993], and reserve capacity [Wong and Yang, 1997, Ziyou and Yifan, 2002]. The decision variables have included capacity improvements [Chiou, 2005, Meng and Yang, 2002], optimal toll values [Yang and Bell, 1997], and scheduling traffic lights [Yang and Yagar, 1995, Chiou, 2008]. The distributed network design problem proposed in this chapter aims to minimize system travel time in all levels. While the decision variables for design problems solved in the lower level are capacity improvements, the higher level design problem includes a mixture of budget allocation and capacity improvements decisions.

Dantzig et al. [1979] solved the CNDP using a decomposition algorithm in which a separate subproblem is solved for each link. A master problem, where the solutions to subproblems form its objective, is also introduced to find the flow for each link. Suwansirikul et al. [1987] developed a solution algorithm, called equilibrium decomposed optimization. Similar to Dantzig et al. [1979], the problem is decomposed into a set of interacting subproblems where each subproblem deals with one link with specific bounds on decision variables. After finding the optimal design variables for all links, a UE problem is solved and then the bounds are updated using information from the gradient values. This process is repeated until the upper and lower bounds on decision variables are close enough. Meng et al. [2001] introduced the lower-level problem (UE problem) as a gap function in the upper-level problem and employed an augmented Lagrangian algorithm to relax this nonlinear equality constraint. This method was later used by Yang et al. [2004] to select the optimal tolls for private highways where tolls are charged based on the entry-exit points.

Chiou [2007] proposed a generalized bundle subgradient projection algorithm to solve the CNDP where the user equilibrium subproblem is presented as a variational inequality. This work was later extended in Chiou [2009] by proposing a new conjugate subgradient projection method. In a similar study, Ban et al. [2006] formulated the CNDP with the UE subproblem expressed as a nonlinear complementary problem. Wang and Lo [2010] formulated the NDP as a single-level problem with equilibrium constraints and then approximated the equilibrium constraints as a set of mixed-integer constraints. Luathep et al. [2011] generalized the mixed-integer programming formulation developed by Wang and Lo [2010] for mixed NDP and proposed a link-based formulation to avoid path enumeration. More recently, Li et al. [2012] used the gap function of the lower-level UE problem and decomposition techniques to convert the CNDP into a sequence of single-level concave optimization problems.

The difficulty of a NDP is due to its nonconvexity, nondifferentiability, and the need to solve one

equilibrium problem after any change in upper level decision variables. As discussed by Yang et al. [1994], the implicit function relating the link flows to improvement decision variables is a nonlinear equality constraint making the problem nonconvex. More precisely, let $(\mathbf{y}_1, \mathbf{x}_1(\mathbf{y}_1))$ and $(\mathbf{y}_2, \mathbf{x}_2(\mathbf{y}_2))$ be two feasible points where $\mathbf{y}_z$, $z = 1, 2$, denotes the vector of decision improvements selected by the planner and $\mathbf{x}_z(\mathbf{y}_z)$ is the UE link flow solution for improvements $\mathbf{y}_z$. Now consider a point $(\mathbf{y}_3, \mathbf{x}_3)$ with $\mathbf{y}_3 = \lambda\mathbf{y}_1 + (1-\lambda)\mathbf{y}_2$ and $\mathbf{x}_3 = \lambda\mathbf{x}_1(\mathbf{y}_1) + (1-\lambda)\mathbf{x}_2(\mathbf{y}_2)$ as the convex combination of these two feasible solutions. This, however, may not be a feasible solution since we cannot guarantee that $x(y_3) = x_3$. Also, as proved by Hall [1978] and Lu and Nie [2010b], the UE link flow and origin-destination (OD) travel times are continuous functions of demand, but they may not be convex, concave, or differentiable.

Due to these difficulties, searching for a global optimal solution requires an extensive amount of computational resources and time, and the algorithms discussed above are mostly capable of solving small problems instances. In addition, due to to non-convexity of the problem, checking the global optimality of a solution generated by algorithms is not practical. As a result, there has been a significant amount of research developing heuristic algorithms for the problem aiming to find a local optima. One of the earliest heuristics developed for solving the NDP is the iterative optimization-assignment algorithm [Allsop, 1974, Steenbrink, 1974], which iterates between solving the upper level problem (optimization) for fixed flows and solving the lower level problem (assignment) for fixed link improvement decisions. This algorithm has reasonable computational time but may end up at a solution dramatically different from the optimal one [Harker and Friesz, 1984]. Friesz and Harker [1985] showed that the iterative optimization-assignment algorithm is exact for problems in which the planner is myopic to the users's behavior (Cournot-Nash game), but is not appropriate for solving the user equilibrium network design problem, a Stackelberg game, in which the reaction of users play the core role in the success of the improvement decisions. Another class of heuristics are link usage proportion-based algorithms, where a given path-flow solution to the lower level problem is used predict how link flows may change [Yang and Bell, 1997, Yang et al., 2004, 1994]. This class of algorithms are mainly suitable for design problems in which OD demand is an upper-level decision variable (e.g., ramp metering, OD matrix estimation).

Sensitivity analysis based (SAB) algorithms are another class of heuristics which have received special attention. The SAB approach performs the user equilibrium sensitivity analysis on the lower-level problem with respect to upper-level decision variables to capture the reaction of users to upper-level design variables. This sensitivity analysis information is then used to compute a descent direction for the upper-level problem and to update the upper-level decision variables. The SAB algorithms are the same as gradient-based solution algorithms for a general BLPP, and have been successfully applied in many network design applications [Yang et al., 1992, 1994, Yang and Yagar, 1995, Miyagi and Suzuki, 1996, Chiou, 1999, Ziyou

and Yifan, 2002, Josefsson and Patriksson, 2007, Chiou, 2009, Dempe and Zemkoho, 2012]. The proposed formulation and solution algorithm in this chapter heavily relies on the theory of user equilibrium sensitivity analysis.

The rest of this chapter is organized as follows. Section 4.2 discusses the main advantages of the proposed allocation-design problem. Section 4.3 formally defines the problem of interest by introducing the regional-level and urban-level design problems, describing the interactions between them. The mathematical formulations of these regional- and urban-level problems are then presented in Section 4.4. Section 4.5 develops a solution algorithm based on a SAB heuristic for both design problems and discusses how to relax the non-linear constraints and address demand elasticity. The implementation of the algorithm is discussed in Section 4.6, and Section 4.7 concludes this work by discussing the implications and limitations of this study.

4.2 Motivation

This section discusses the main advantages of the decentralized network design algorithm proposed in this chapter. The first two points are directly related to traditional bi-level NDPs, while the last three ones are mainly for multi-level application where network is composed of different regions with possibly different priorities and concerns.

1. Simplifying he problem: As with the DSTAP algorithm discussed in Chapter 3, the decentralized scheme for NDP developed here also partitions the network into smaller subnetworks, and introduces a solver for each subnetwork. The local solvers are responsible for solving the NDP over their subnetworks while taking into account the system-level impact of their planning decisions. As a result, the subnetwork design problems are of smaller scale and easier to formulate and deal with compared to the original problem for the complete network.

2. A faster solution algorithm: The decentralized algorithm for NDP has a better run time because of two properties: (1) the subnetwork design problems are defined over a region of smaller scale with fewer decision variables, and (2) the subnetwork design problems can be parallelized.

3. Modeling interactions: Developing transportation planning projects by considering just the local concerns while neglecting the system-level impacts can result in sub-optimal solutions. Neighborhood traffic calming is one example. Adding stop signs, lowering the speed limit, narrowing traffic lanes, curb extension, roundabounts, and blocking some turning movements can be used to encourage safer driving. These strategies may also lower the congestion. Without proper modeling of

the impact of such local changes on a larger scale, however, the traffic may shift from these neighborhoods to other corridors causing additional congestion in other parts of the network. The same problem can happen in a network of different urban cities. The proposed scheme in this chapter for representing the interactions between different regions can alleviate this problem.

4. Modeling conflicting objectives: Different set of goals can be considered when developing transportation improvement projects. In addition to minimizing the total system travel time, regions may prefer to improve safety, decrease pollution, encourage bicycling and ridesharing, make the city more attractive for visitors and residents, boost the economy and job market by attracting businesses and industries, etc. Dealing with these different and sometimes even conflicting objectives (minimizing travel time may not be well aligned with boosting the economy or attracting more visitors or investors) individually in a decentralized implementation is another main advantage of the solution algorithm developed in this chapter.

5. Equity: A measure of equity, defined as the regret function and introduced in Section 4.3, ensures that small subnetworks with marginal benefits compared to larger regions will receive an equitable amount of funding.

4.3 Problem Statement

This section reviews the formulation of the CNDP used in this study and introduces definitions used in the rest of this chapter. Figure 4.1 provides an illustration of the definitions given below, for a full network with 2 urban cities, and the following origin-destination (OD) pairs: r-s, r-10, 10-s, 1-11, 2-4 and 9-8.

We partition the network into a regional network, which is managed by a regional agent (such as a state Department of Transportation), and cities governed by urban agents (maybe metropolitan planning organization). We denote the regional agent by E_r, and urban agent for city u by E_u. Let $G_u = (N_u, A_u, W_u)$ denote the network for city $u \in U$, where N_u, A_u, and W_u are the sets of nodes, links, and OD pairs in city u. We call nodes, links, and OD pairs of an urban city as urban nodes, links, and OD pairs, respectively, and trips corresponding to urban OD pairs are referred to as urban demand. We assume that urban cities do not overlap and are managed by independent urban agents.

Links and nodes which are not part of any urban city are called regional links and regional nodes, respectively. The urban cities are connected via regional links and regional nodes, and the urban nodes which are the tail or head node of a regional link are denoted as boundary nodes. Example of boundary nodes in Figure 4.1 are nodes $2, 4, 6, 9$ and 12. In addition, the OD pairs which start and end at dif-

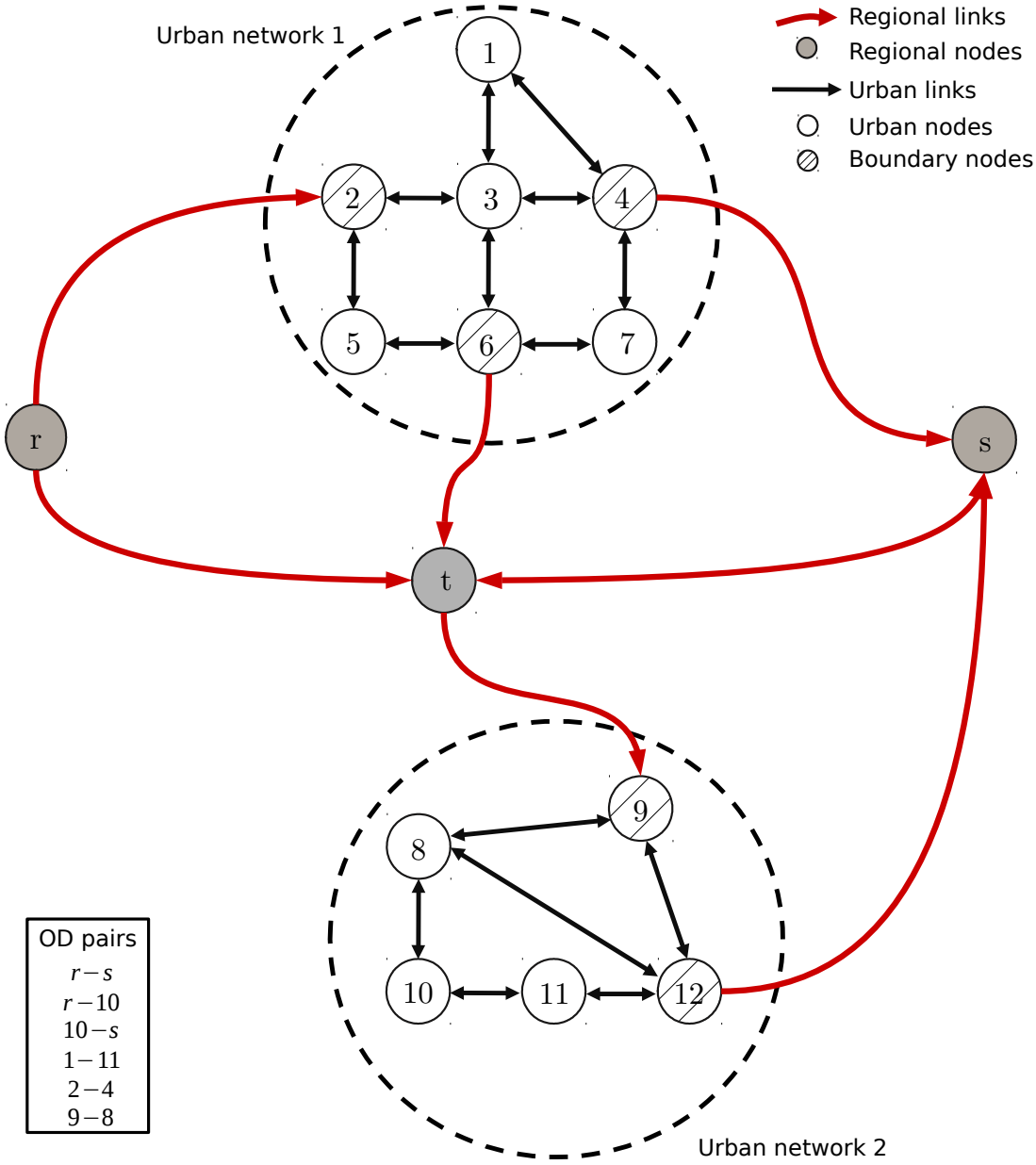


Figure 4.1: Full network with two urban cities.

ferent urban cities or at regional nodes are referred to as regional OD pairs, and trips between them are called regional demand. Examples of regional OD pairs are r-s, r-10, 10-s and 1-11. These regional trips, in addition to regional links and nodes, may use some urban links and nodes by entering or exiting at boundary nodes. Let $\langle a, b\rangle$ denote the set of all alternatives routes between nodes a and b. As an example, there exist 5 alternatives for trips between OD pair r-s: $\{r, \langle 2, 4\rangle, s\}$, $\{r, \langle 2, 6\rangle, t, s\}$, $\{r, t, s\}$, $\{r, \langle 2, 6\rangle, t, \langle 9, 12\rangle, s\}$, and $\{r, t, \langle 9, 12\rangle, s\}$. As an example, the set $\langle 9, 12\rangle$ includes the following paths: $\{9, 12\}$, $\{9, 8, 12\}$, and $\{9, 8, 10, 11, 12\}$. For the purpose of modeling the regional network, and similar to the approach proposed in Chapter 2 and employed in Chapter 3, we model these internal subpaths with artificial links between the urban nodes which may attract some regional demand (boundary nodes and urban nodes which are origin or destination to some regional demand).

Let $G_r = (N_r, A_r, W_r)$ denote the regional network, where N_r, A_r, and W_r are the sets of nodes, links, and OD pairs in the regional network. Based on the above discussion, N_r includes all regional nodes and urban nodes which are either an endpoint of a regional link (urban boundary nodes), or are the origin or destination of some regional demands. Set A_r consists of all regional links and artificial links representing the urban networks, and W_r is the set of OD pairs with start and end at different urban cities or at a regional node. Figure 4.2 depicts the regional network associated with the network plotted in Figure 4.1.

As mentioned before, the urban cities are managed by independent agents which have full authority over their networks. The regional agent can only manage the regional links and nodes, and does not have the authority to plan or dictate any change or modification for urban areas. The regional agent, however, has a total budget B to be split among the regional projects (capacity improvements on regional links) and urban networks. Let H_r denote set of candidate regional projects considered by the regional agent. The urban agent solves a CNDP on the regional network with the objective to improve the quality of trip for regional demands while minimizing the *dissatisfaction* of urban agents when splitting the budget B. Let $B_{r,u}$ be the external budget assigned to urban city u. The proposed dissatisfaction factor is proportional to the difference between the objective value of urban agent E_u for current budget $B_{r,u}$ and the case when total budget B is allocated to urban city u. More detail on the formulation of the dissatisfaction factor are presented later.

Urban agents, on the other hand, use their internal budget in conjunction with the budget assigned to them from the regional agent to implement their planning projects. Let B_u denote the internal budget of urban city u. Furthermore, let H_u denote the set of candidate links subject to capacity improvement at urban city u. The urban agents aim to optimize their objective by splitting their total budget, $B_u + B_{r,u}$, between (a subset of) candidate links. Here we assume that urban cities are spatially distributed such that the internal trips from any urban city u are constrained to the nodes and link within u.

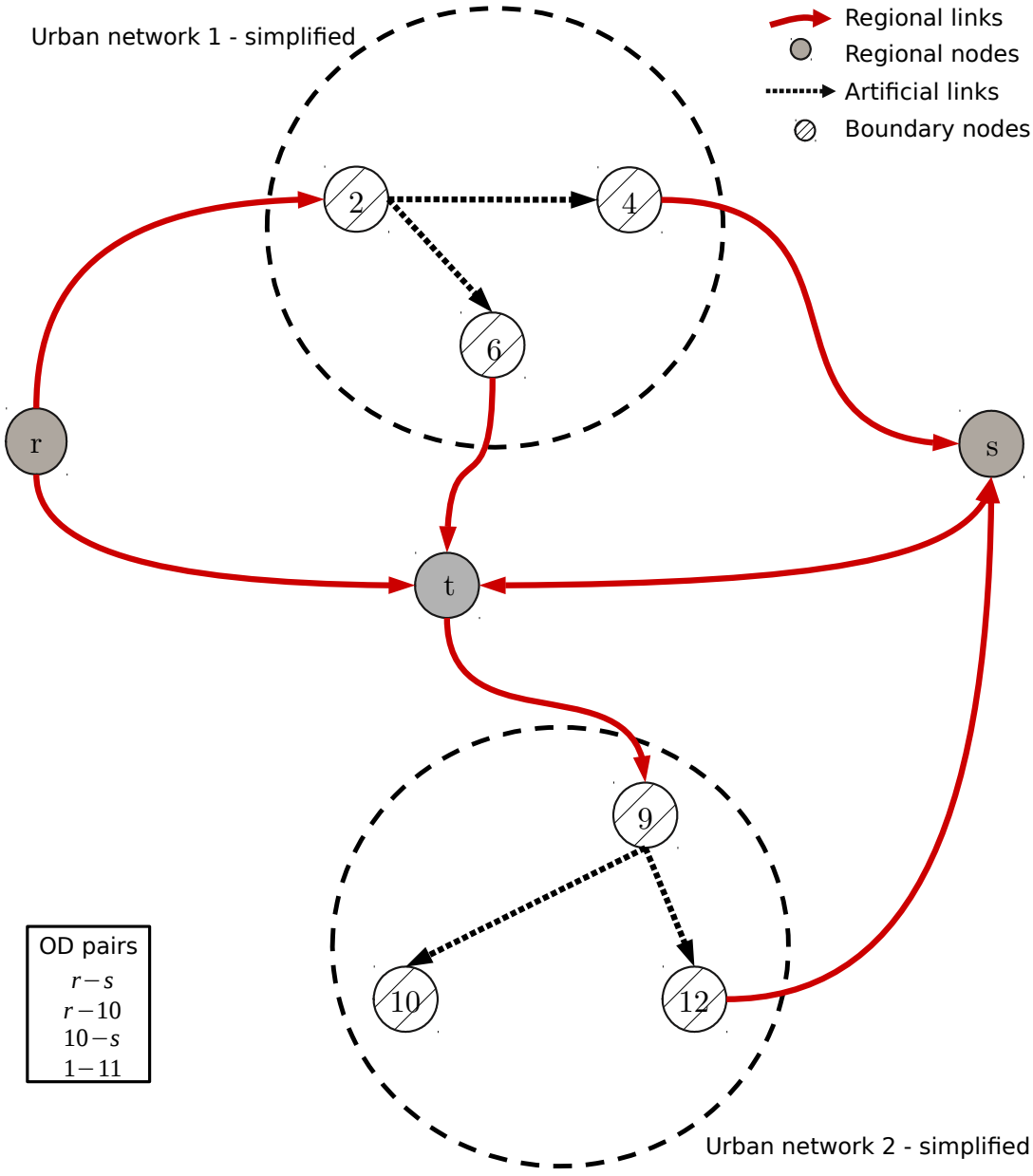


Figure 4.2: Regional network in which urban networks are represented as artificial links between nodes which can attract regional demand.

The urban agents, however, need to account for route choice behavior of the regional demand when designing the improvements. More precisely, increasing the capacity of the links used by regional demand may attract more regional demand which in turn can increase the congestion level on those corridors. To this end, the urban agents solve a CNDP with elastic demand for OD pairs which may attract regional demand. Specifically, OD pairs 2-4 and 2-6 in urban city 1, and OD pairs 9-10 and 9-12 in urban city 2 are OD pairs with elastic demand. These OD pairs are represented by artificial links in the regional network. Due to demand elasticity, minimizing the total system travel time may result in plans discouraging users from traveling. This is not acceptable. Thus, for urban agents, the objective of the upper-level problem is to maximize social surplus for OD pairs with elastic demand and to minimize the total system travel time for those with fixed demand, and the lower-level problem is a UE problem with elastic demand. Section 4.4 formulates the continuous network design problems for the regional and urban networks.

The solution algorithm for solving the proposed network design problems is a sensitivity analysis based heuristic. In this heuristic, and at each iteration, the upper-level decision variables (regional link improvements and budget allocation in case of the regional network and urban link improvements in case of urban networks) are updated by solving an equivalent linear program and finding the steepest descent direction. In Section 4.5.1 we discuss a general framework for performing the equilibrium sensitivity analysis which will be used extensively later to develop solution algorithms for the proposed network design problems in Sections 4.5.4 and 4.5.5, respectively.

Figure 4.3 provides a schematic of the overall network design problem studied here. This problem is a CNDP with 4 inter-connected levels which are solved iteratively in search of a local optimum:

L 1- $P_r(0)$: The first level deals with the decision of the regional agent regarding how to allocate the budget between regional projects and different urban cities.

L 2- $P_r(1)$: The second level models the route choice behavior of regional demand in response to regional and urban improvements suggested by the regional and urban agents, respectively.

L 3- $P_u(0)$: The third level captures the investment decisions of the urban agents based on their internal budget and the external budget received from the regional agent. At this level, one problem is defined for each urban agent and these urban design problems are solved simultaneously.

L 4- $P_u(1)$: The last level models the route choice behavior of urban trips in response to improvement plans implemented by the urban agents. These UE subproblems are solved simultaneously.

In the next section, we provide more detail on different stages of the problem and mathematically formulate the problem dealt at each level.

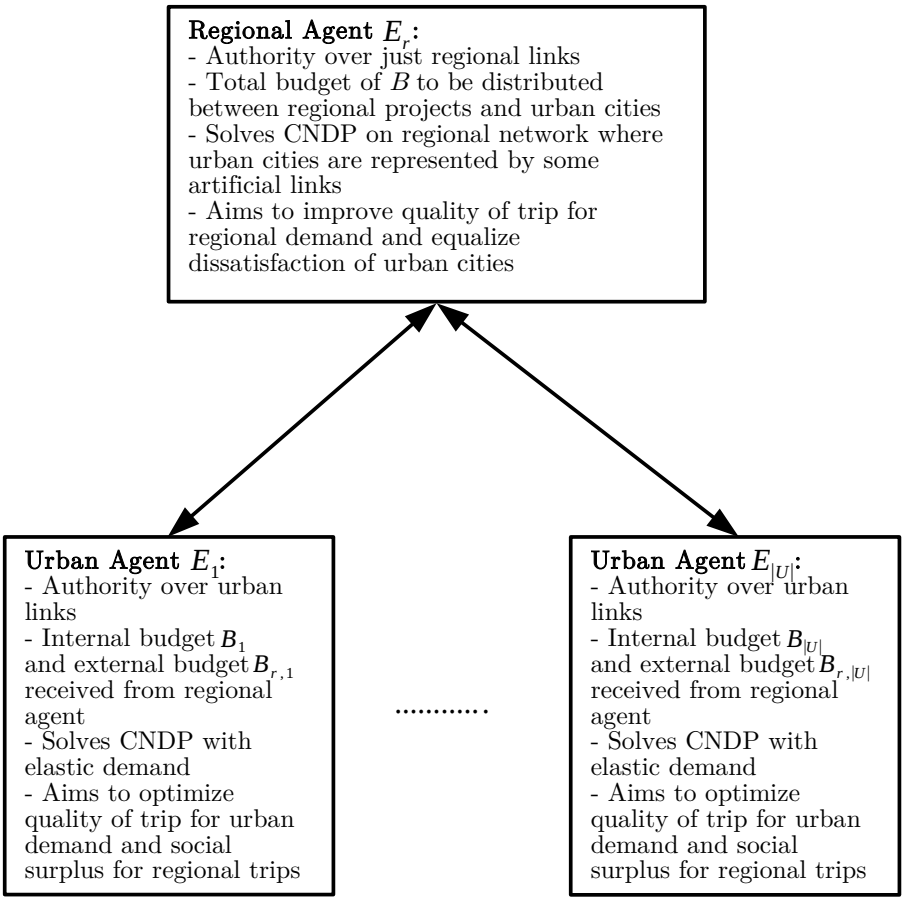


Figure 4.3: The overall design of the proposed network design problem and tasks of each agent.

4.4 Problem Formulation

In this section, we first present the network design problem of the regional agent and then formulate the problem of the urban agents.

4.4.1 Regional-level network design problem

The regional agent E_r must allocate the budget B among regional links in H_r and urban cities. In assigning the budget, the regional agent aims to improve the quality of trip for regional demands, formulated as total travel time, and also to equalize the dissatisfaction of urban cities. The dissatisfaction is formulated as regret function [Cassidy et al., 1971]. For any urban city u and under assigned budget $B_{r,u}$, the regret function measures the difference between the objective value of urban city u under external budget $B_{r,u}$ and the ideal case in which the total budget is assigned to u, i.e., $B_{r,u} = B$. This function, denoted by $R_u(B_{r,u})$, is formulated as:

$$R_u(B_{r,u}) = \frac{F_u^0(B_{r,u}) - F_u^0(B)}{F_u^0(B)} \tag{4.5}$$

where $F_u^0(\cdot)$ is the optimal value of the upper-level objective function of city $u \in U$ given a specific budget value. Note that $R_u(B_{r,u})$ is a nonnegative variable, and $R_u(B_{r,u}) = 0$ indicates the optimal budget assignment from the perspective of urban city u. The goal of the regret function is to provide equity among different urban cities. If we simply sum the objectives of urban cities in the objective of the regional agent, then larger cities may receive a high portion of the budget because improvements, in terms of absolute change in system travel time, can be significant. The normalization factor introduced as the denominator of the regret function ensures that equal improvements, in terms of the percentage values, will have the same weights.

The urban agent solves the network design problem on a network consisting of regional links and nodes and artificial links representing the urban cities (Figure 4.2). Let Θ_u^r denote the set of artificial links created for urban city u in the regional network, and let Θ be the set of all artificial links in the regional network, i.e., $\Theta = \bigcup_{u \in U} \Theta_u^r$. The flow on each artificial link corresponds to external demand for urban city u, and we use Θ_u to denote the associated set of OD pairs in urban city u. In Figure 4.2, for urban city 1 we have $\Theta_1^r = \{(2,4),(2,6)\}$ and $\Theta_1 = \{2\text{-}4, 2\text{-}6\}$.

Let y_b be a decision variable denoting the capacity improvement for link $b \in H_r$ with the associated cost of $G_b(y_b)$. We assume that $G_b(y_b)$ is a differentiable and non-decreasing function of y_b. We group the capacity improvement decision variables for regional links into an $|H_r|$-dimensional vector $\mathbf{y}_r$ and budget

assigned to different cities into a $|U|$ dimensional vector $\mathbf{B}$. The network design problem over the regional network, denoted R-CNDP, may be written as:

$$P_r(0): \quad \min_{\mathrm{y}_r, \mathrm{B}} F_r^0(\mathrm{y}_r, \mathrm{B}) = \sum_{w \in W_r} d_w T_w(\mathbf{x}_r) + \sum_{u \in U} \eta_u R_u(B_{r,u}) \tag{4.6}$$

$$\text{subject to} \quad \sum_{u \in U} B_{r,u} + \sum_{b \in H_r} G_b(y_b) \leq B, \tag{4.7}$$

$$0 \leq y_b \leq \tilde{y}_b \qquad \forall b \in H_r, \tag{4.8}$$

$$0 \leq B_u \leq \tilde{B}_{r,u} \qquad \forall u \in U. \tag{4.9}$$

where $\mathbf{x}_r = [x_a]$ is the vector of link flows and $T_w(\cdot)$ denotes the travel time between the endpoints of OD pair w. The objective function (4.6) is the weighted sum of the total travel time on the regional network and the urban regret functions, constraint (4.7) ensures that the sum of the budget assigned to urban cities and invested on candidate regional links is less than available budget, and equations (4.8) and (4.9) display the lower and upper bounds on link improvements and urban budget allocations, respectively.

In problem $P_r(0)$, and for a given set of decision variables $\mathbf{y}_r$ and $\mathbf{B}$, the UE flow vector $\mathbf{x}_r$ is defined implicitly by the following convex optimization problem:

$$P_r(1): \quad \min_{\mathrm{x}_r} F_r^1(\mathbf{x}_r) = \sum_{a \in A_r \backslash \Theta} \int_0^{x_a} t_a(\omega, y_a)\, d\omega + \sum_{\theta \in \Theta} \int_0^{x_\theta} t_\theta(\omega, B_{r,u})\, d\omega \tag{4.10}$$

$$\text{subject to} \sum_{\pi \in p_w} f_\pi = d_w \qquad \forall w \in W_r, \tag{4.11}$$

$$\sum_{w \in W_r} \sum_{\pi \in p_w} f_\pi \delta_{a\pi} = x_a \qquad \forall a \in A_r \backslash \Theta, \tag{4.12}$$

$$\sum_{w \in W_r} \sum_{\pi \in p_w} f_\pi \delta_{\theta\pi} = x_\theta \qquad \forall \theta \in \Theta, \tag{4.13}$$

$$f_\pi \geq 0 \qquad \forall \pi \in p_w, w \in W_r. \tag{4.14}$$

where d_w and p_w denote the travel demand and set of paths between regional OD pair w, f_π is flow on path π, and the indicator variable $\delta_{a\pi}(\delta_{\theta\pi})$ is 1 if path π uses link $a(\theta)$, and 0 otherwise. Note that $y_a = 0$ for any link a not included in set H_r.

The first summation in the objective function of problem $P_r(1)$, equation (4.10), is over all physical regional links, and the second term sums over all artificial links in the regional network. The travel time on any physical artificial link $a \in A_r \backslash \Theta$ is a function of regional link improvement variable y_a, while the

travel time on any artificial link $\theta \in \Theta_u^r$ depends on the external budget $B_{r,u}$ assigned to urban city u.

Let $\hat{\mathbf{y}}_r$, $\hat{\mathbf{B}}$, and $\hat{\mathbf{x}}_r$ denote the current solutions to problems $P_r(0)$ and $P_r(1)$, respectively. (Details of the solution algorithm will be discussed later.) Before solving the CNDP for urban cities based on the new external budget $\hat{\mathbf{B}}$, we need to update the demand functions of urban OD pairs with regional demand. Recall that these OD pairs belong to artificial links modeled in the regional network, and the demand functions are introduced to predict the number of regional trips attracted to each urban OD pair as a function of OD equilibrium travel time.

Let $w \in \Theta_u \subseteq W_u$ be an urban OD pair represented in the regional network by artificial link $\theta \in \Theta_u^r \subseteq A_r$, $d_{r,w}(\cdot)$ be the regional demand attracted to travel between the endpoints of OD pair w inside urban city u, and $\hat{d}_{r,w} = \hat{x}_\theta$ denote the current value of $d_{r,w}$. We approximate the demand function, $D_{r,w}(\cdot)$, as a linear function of OD travel time:

$$\begin{aligned} d_{r,w}(T_w) &= D_{r,w}(T_w) \\ &= \hat{d}_{r,w} + \frac{\partial d_{r,w}}{\partial T_w}(T_w - \hat{T}_w) \end{aligned} \tag{4.15}$$

where $\hat{T}_w$ is the current travel time between the endpoints of w (demand $\hat{d}_{r,w}$ is attracted based on this travel time), and $\partial d_{r,w}/\partial T_w$ is the derivative of $d_{r,w}(\cdot)$ with respect to T_w evaluated at $\hat{T}_w$. This derivative is the only unknown variable to be estimated, and will be discussed in the Section 4.5.4.

Next, we formulate the continuous network design problem for each urban city u.

4.4.2 Urban-level network design problem

In the previous section we formulated the CNDP of the regional agent. The solutions to problems $P_r(0)$ and $P_r(1)$ determine the external budget assigned to each urban city $u \in U$, i.e., $B_{r,u}$, and the demand functions, formulated in (4.15), specifying the regional demand attracted to urban OD pairs modeled in the regional network (OD pairs is Θ_u). Here we describe the CNDP of urban cities, denoted as U-CNDP, and then discuss how the artificial links in the regional network are updated.

Each urban agent E_u has a total budget of $B_u + B_{r,u}$ to be invested on urban projects defined by set H_u. Likewise the regional agent who aims to minimize the total system travel time of the regional travelers subject to a constraint on the satisfaction rate of urban agents, equations (4.6)–(4.9), the urban agents need to consider the consumers' surplus. Minimizing the total system travel time when demand is elastic may result in implementing plans discouraging the regional travelers from traveling through the urban cities. This is not practical and plausible. An appropriate objective function for the upper-level

problem at each urban city can be to maximize the social surplus for the regional demand and to minimize the total travel cost for urban demands. The social surplus for regional demand can be defined as the difference between the social benefit, $\sum_{w\in\Theta_u}\int_0^{d_{r,w}} D_{r,w}^{-1}(\omega)\,d\omega$, and social cost incurred by regional travelers, $\sum_{w\in\Theta_u} d_{r,w}T_w$. The upper-level problem for urban network u, denoted by $P_u(0)$, can be formulates as:

$$P_u(0):\ \min_{\mathrm{y}_u} F_u^0(\mathbf{y}_u) = \sum_{w\in W_u} d_w T_w(\mathbf{x}_u) - \sum_{w\in\Theta_u}\Big(\int_0^{d_{r,w}} D_{r,w}^{-1}(\omega)\,d\omega - d_{r,w}T_w(\mathbf{x}_u)\Big) \tag{4.16}$$

$$\text{subject to}\quad \sum_{b\in H_u} G_b(y_b) \le B_u + B_{r,u}, \tag{4.17}$$

$$0 \le y_b \le \tilde{y}_b \qquad \forall b\in H_u \tag{4.18}$$

where $\mathbf{y}_u$ denotes the vector of improvement decision variables for urban links, and $\mathbf{x}_u$ solves the following UE problem with elastic demand:

$$P_u(1):\qquad \min_{\mathbf{x}_u,\mathbf{d}_u} F_u^1(\mathbf{x}_u,\mathbf{d}_u) = \sum_{a\in A_u}\int_0^{x_a} t_a(\omega,y_a)\,d\omega - \sum_{w\in\Theta_u}\int_0^{d_{r,w}} D_{r,w}^{-1}(\omega)\,d\omega \tag{4.19}$$

$$\text{subject to}\ \sum_{\pi\in p_w} f_\pi = d_w \qquad \forall w\in W_u, \tag{4.20}$$

$$\sum_{w\in W_u}\sum_{\pi\in p_w} f_\pi\delta_{a\pi} = x_a \qquad \forall a\in A_u, \tag{4.21}$$

$$f_\pi \ge 0 \qquad \forall \pi\in p_w, w\in W_u. \tag{4.22}$$

where $\mathbf{x}_u$ and $\mathbf{d}_u$ denote the vector of link flows and regional flows attracted to u, respectively.

Let $\hat{\mathbf{y}}_u$, $\hat{\mathbf{x}}_u$ and $\hat{\mathbf{d}}_u$ denote the solutions to problems $P_u(0)$ and $P_u(1)$. (Details of the solution algorithm will be discussed in Section 4.5.) Before solving the R-CNDP, problems $P_r(0)$ and $P_r(1)$ for new flow assignment at urban cities, we need to update the parameters of artificial links in the regional network. Recall that these artificial links belong to urban OD pairs with regional demand (those with elastic demand) which are denoted by Θ_u.

Each artificial link $\theta\in\Theta_u^r$ represents all used paths in the urban city u connecting its tail to its head, and will be equipped with cost functions which represent the equilibrium travel time between these nodes inside u, as a function of the regional demand between these points and the external budget assigned to urban city u. Let w denote the OD pair represented by the artificial link θ in the regional network. At

each iteration of the R-CNDP, these artificial regional links have the following cost function:

$$t_\theta(x_\theta) = \mu_\theta + \psi_{\theta,x}(x_\theta - \hat{x}_\theta) + \psi_{\theta,B}(B_{r,u} - \hat{B}_{r,u}), \qquad \forall \theta \in \Theta^r_u, u \in U \tag{4.23}$$

where $t_\theta(\cdot)$ denotes the travel time variable, x_θ is the amount of regional demand on the artificial link θ, $\hat{x}_\theta = \hat{d}_{r,w}$ is the current regional demand attracted to urban city u, $B_{r,u}$ is the external budget assigned to u, and $\hat{B}_{r,u}$ is the current value of $B_{r,u}$. Variable μ_θ is the weighted average travel time of the paths represented by θ (weighted by flow), and $\psi_{\theta,x}$ and $\psi_{\theta,B}$ respectively denote the derivative of this average travel time with respect to flow x_θ and external budget $B_{r,u}$ evaluated at $\hat{x}_\theta$ and $\hat{B}_{r,u}$. In equation (4.23), x_θ and $B_{r,u}$ are decision variables adjusted at the regional level, $\hat{B}_{r,u}$ is known, and parameters μ_θ and $\hat{x}_\theta$ can be directly computed from the solutions to problems $P_u(0)$ and $P_u(1)$. Thus the only unknown variables are $\psi_{\theta,x}$ and $\psi_{\theta,B}$, which will be discussed in Section 4.5.5.

Next we discuss the solution algorithms for regional problems $P_r(0)$ and $P_r(1)$ and urban problems $P_u(0)$ and $P_u(1)$, for every urban city $u \in U$.

4.5 Solution Algorithms

As discussed in Section 4.1, the CNDP is a non-convex optimization problem, and searching for global solution may require a tremendous amount of effort and computing resources. In addition, each iteration of the CNDP requires solving one UE problem to evaluate the impact of improvement decision variables, which can be time-consuming on large-scale networks. As a result, the literature has mainly focused on developing heuristic algorithms for this problem. SAB algorithms are among the best heuristics successfully applied to network design problems. Each iteration of an SAB algorithm starts with solving the upper-level problem, where the impact of upper-level decision variables is modeled through lower-level sensitivity analysis. Essentially, a SAB algorithm predicts the behavior of the users, modeled as the lower-level problem, in response to link improvement decision variables adjusted at the upper-level. After updating the upper-level decision variables, the lower-level UE problem is solved for the new design parameters. Then a sensitivity analysis is performed to compute the derivatives of link flows with respect to link improvement decision variables. This process is repeated until a measure of convergence is reached.

Before discussing the detail of the SAB algorithm for regional and urban network design problems, we describe a procedure to perform equilibrium sensitivity analysis (required for updating artificial links in regional network and demand functions for urban cities), discuss how to relax the nonlinear inequality constraints (4.7) and (4.17), and reformulate the elastic-demand user equilibrium problem $P'_u(1)$ ((4.19)–

(4.22)) as a fixed-demand problem.

4.5.1 User Equilibrium Sensitivity Analysis

This section formulates the user equilibrium sensitivity analysis problem. Let $\boldsymbol{\epsilon} = [\epsilon]$ be the vector of disturbances where ϵ denotes the disturbance parameter affecting a subset of link travel costs and/or OD demand. Let $\varphi(\epsilon)$ and $\varrho(\epsilon)$ respectively denote the set of links and OD pairs affected by disturbance parameter ϵ. In the following discussion, the variable $\delta_a(\epsilon)$ is 1 if travel time on link a depends on ϵ, and 0 otherwise. Similarly, $\delta_w(\epsilon)$ is 1 if demand between the OD pair w is a function of disturbance parameter ϵ, and 0 otherwise. Here we discuss how to compute the derivative of links flows, path flows, and OD travel times with respect to ϵ.

Let $\hat{\mathbf{x}}$ denote the UE flow for some $\boldsymbol{\epsilon}$ on a network $G = (N, A, Z)$ (regional or urban), and $\hat{p}_w$ and $\hat{A}_w$, respectively, be the set of paths and links used by d_w at the equilibrium flow $\hat{\mathbf{x}}$. Furthermore, let $\beta_\pi^\epsilon = \partial h_\pi / \partial \epsilon$ be the derivative of path π's flow with respect to ϵ, $\alpha_a^\epsilon = \partial x_a / \partial \epsilon$ be the derivative of link a's flow with respect to ϵ, and $t_a' = \mathrm{d}t_a / \mathrm{d}x_a$ be the derivative of link a's travel time with respect to the link flow evaluated at $\hat{\mathbf{x}}$.

The disturbance parameter ϵ affects all OD pairs in $\varrho(\epsilon)$, and also the OD pairs using links in $\varphi(\epsilon)$ (OD pairs with at least one path using links in $\varphi(\epsilon)$ under flow assignment $\hat{\mathbf{x}}$.) Let $G(\epsilon)$ denote the set of affected OD pairs, and $A(\epsilon) = \bigcup_{w \in G(\epsilon)} \hat{A}_w$ be the union of the links used by the OD pairs in $G(\epsilon)$. Based on the assumption that the set of used paths remains unchanged, the user equilibrium sensitivity analysis problem with respect to ϵ can be formulated as the following convex optimization problem:

$$\text{minimize} \sum_{a \in A(\epsilon)} \Big(\int_0^{\alpha_a^\epsilon} t_a' \omega \, d\omega + \frac{\mathrm{d}t_a}{\mathrm{d}\epsilon} \alpha_a^\epsilon \delta_a(\epsilon) \Big) \tag{4.24}$$

$$\text{subject to} \sum_{\pi \in \hat{p}_w} \beta_\pi^\epsilon = \delta_w(\epsilon), \qquad \forall w \in G(\epsilon) \tag{4.25}$$

$$\alpha_a^\epsilon = \sum_{w \in G(\epsilon)} \sum_{\pi \in \hat{p}_w} \beta_\pi^\epsilon \delta_{a\pi}, \qquad \forall a \in A(\epsilon) \tag{4.26}$$

This is essentially a static user equilibrium problem on the network comprised of paths used by the OD pairs in $G(\epsilon)$ (subnetwork comprised of links in $A(\epsilon)$), with α_a^ϵ and β_π^ϵ serving the role of link flows and path flows, respectively. Solving this problem yields the derivatives of link flows and path flows (α_a^ϵ and

β_π^ϵ, respectively), and derivative of OD travel time for any OD pair $w \in G(\epsilon)$ can be formulated as:

$$\frac{\partial T_w}{\partial \epsilon} = \sum_{a \in A(\epsilon)} \left(t'_a \alpha_a^\epsilon + \frac{\mathrm{d}t_a}{\mathrm{d}\epsilon} \delta_a(\epsilon) \right) \delta_{a\pi} \tag{4.27}$$

where path π is any of the used paths between endpoints of OD pair w. These derivatives only exist if the solution is strictly complementary in the sense that all minimum-cost routes have positive flow. Please refer to Jafari and Boyles [2016] for more detail on user equilibrium sensitivity analysis problem (4.24)–(4.26).

The proposed sensitivity analysis problem will be used in the following sections to estimate the parameters of urban demand functions, equation (4.15), and artificial links, equation (4.23). In addition, the implementation of SAB algorithm extensively relies on this equilibrium sensitivity problem.

4.5.2 Relaxing Non-linear Constraints

The ALM is a well-known algorithm for solving optimization problems with non-linear constraints. In the augmented Lagrangian method, the non-linear side constraints are excluded by incorporating them into the objective function and solving a sequence of less constrained optimization problems. The augmented Lagrangian method starts with a point, not necessarily feasible, and reduces the violation of the constraints by updating the Lagrange multipliers and penalty term iteratively. Chapter 4 of Bertsekas [1999] discusses details of the implementation and correctness of the ALM algorithm.

The augmented Lagrangian formulation with quadratic penalty function at iteration k of R-CNDP is a problem of the following form

$$P'_r(0): \quad \min_{\mathbf{y}_r^k, \mathbf{B}^k} \quad L_r^0(\mathbf{y}_r^k, \mathbf{B}^k, \lambda^k, c^k) \tag{4.28}$$

$$\text{subject to} \quad 0 \le y_b^k \le \tilde{y}_b, \qquad \forall b \in H_r \tag{4.29}$$

$$0 \le B_{r,u}^k \le \tilde{B}_{r,u}, \qquad \forall u \in U \tag{4.30}$$

where $L_r^0(\mathbf{y}_u^k, \mathbf{B}^k, \lambda^k, c^k)$ is the augmented Lagrangian function, c^k is the external penalty coefficient, and λ^k is Lagrange multiplier for the non-linear inequality constraint defined in equation (4.7) at iteration k. The augmented Lagrangian function $L_r^0(\mathbf{y}_u^k, \mathbf{B}^k, \lambda^k, c^k)$ is given by:

$$L_r^0(\mathrm{y}_r^k, \mathbf{B}^k, \lambda_r^k, c_k) = F_r^0(\mathbf{y}_r^k, \mathbf{B}^k) + \frac{1}{2c^k} \left(\left(\max\{0, \lambda^k + c^k g_r(\mathbf{y}_r^k, \mathbf{B}^k)\} \right)^2 - (\lambda^k)^2 \right) \tag{4.31}$$

where $g_r(\mathbf{y}_r^k, \mathbf{B}^k) = \sum_{u \in U} B_{r,u}^k + \sum_{b \in H_r} G_b(y_b^k) - B$. Bertsekas [1999] suggests the following rule

to update these parameters:

$$\begin{aligned} \lambda^k &:= \lambda^{k-1} + c^{k-1} g_r(\mathbf{y}_r^{k-1}, \mathbf{B}^{k-1}) \\ c^k &:= \beta^{k-1} c^{k-1} \end{aligned} \tag{4.32}$$

and:

$$\beta^{k-1} = \begin{cases} \rho & \text{if } \big(g_r^+(\mathbf{y}_r^{k-1}, \mathbf{B}^{k-1})\big)^2 \geq \varphi \big(g_r^+(\mathbf{y}_r^{k-2}, \mathbf{B}^{k-2})\big)^2 \\ 1 & \text{otherwise} \end{cases} \tag{4.33}$$

where $g_r(\mathbf{y}_r^{k-1}, \mathbf{B}^{k-1})$ and $g_r(\mathbf{y}_r^{k-2}, \mathbf{B}^{k-2})$ denote the nonlinear constraint at iteration $k-1$ and $k-2$, respectively, $\rho > 1$, $\varphi = 0.25$, and $g_r^+(x) = \max\{0, g_r(x)\}$.

For the case of U-CNDPs, and similar to regional CNDP, the augmented Lagrangian formulation with quadratic penalty function at iteration k of U-CNDP consists of solving a problems of the form

$$P_u'(0): \qquad \min_{\mathbf{y}_u^k} \quad L_u^0(\mathbf{y}_u^k, \lambda^k, c^k) \tag{4.34}$$

$$\text{subject to} \quad 0 \leq y_b^k \leq \tilde{y}_b, \qquad \forall b \in H_u \tag{4.35}$$

where $L_u^0(\mathbf{y}_u^k, \lambda^k, c^k)$ is the objective function (4.16) augmented with quadratic penalty function of the non-linear inequality constraint (4.17):

$$L_u^0(\mathrm{y}_u^k, \lambda^k, c^k) = F_u^0(\mathbf{y}_u^k) + \frac{1}{2c^k}\Big(\big(\max\{0, \lambda^k + c^k g_u(\mathbf{y}_u^k)\}\big)^2 - (\lambda^k)^2\Big) \tag{4.36}$$

where $g_u(\mathbf{y}_u^k) = \sum_{b \in H_u} G_b(y_b^k) - B_{r,u}^k - B_u$ and update rules similar to (4.32) and (4.33).

4.5.3 Converting Elastic Demand into Fixed Demand

Gartner [1980] proposed a transformation to represent an elastic-demand user equilibrium problem as a fixed-demand problem. To this end, artificial links are added between the endpoints of elastic OD pairs with a cost function given by the inverse demand function. Let l_w denote the artificial link added between the endpoints of OD pair $w \in \Theta_u$ with a demand function given by (4.15), and furthermore, let $\theta \in \Theta_u^r$ be the associated artificial link in the regional network. The maximum regional demand which can go

through urban OD pair $w \in \Theta_u$ would be:

$$\bar{d}^k_{r,w} = \sum_{\nu \in W_r} d_\nu \delta_{\theta\nu}$$

where $\delta_{\theta\nu} = 1$ if regional OD pair ν uses artificial link θ at iteration k, and is 0 otherwise.

The cost of travel on link l_w would be:

$$\begin{aligned} t_{l_w}(x_{l_w}) &= D^{-1}(d^k_{r,w}(\mathbf{x}_u)) \\ &= T^k_w + \frac{1}{\partial d^k_{r,w}/\partial T_w}(\bar{d}^k_{r,w} - x_{l_w} - d^k_{r,w}) \end{aligned} \tag{4.37}$$

where x_{l_w} is the flow on link l_w, and parameters T^k_w, $\partial d^k_{r,w}/\partial T_w$, and $d^k_{r,w}$ are input parameters from the regional network at iteration k. The value of x_{l_w} denotes the excess demand, those regional trips which are not accommodated by OD pair w. Please refer to Gartner [1980] for more detail on this transformation.

Using this transformation, the lower-level problem $P_u(1)$ at iteration k can be written as:

$$P'_u(1): \qquad \min_{\mathbf{x}^k_u} F^1_u(\mathbf{x}^k_u) = \sum_{a \in A_u} \int_0^{x^k_a} t_a(\omega, y^k_a)\, d\omega + \sum_{w \in \Theta_u} \int_0^{x^k_{l_w}} t_{l_w}(\omega)\, d\omega \tag{4.38}$$

$$\text{subject to} \sum_{\pi \in p_w} f_\pi = d_w \qquad \forall w \in W_u \backslash \Theta_u, \tag{4.39}$$

$$\sum_{\pi \in p_w} f_\pi + x^k_{l_w} = d_w + \bar{d}^k_{r,w} \qquad \forall w \in \Theta_u, \tag{4.40}$$

$$\sum_{w \in W_u} \sum_{\pi \in p_w} f_\pi \delta_{a\pi} = x^k_a \qquad \forall a \in A_u, \tag{4.41}$$

$$f_\pi \geq 0 \qquad \forall \pi \in p_w, w \in W_u, \tag{4.42}$$

$$x^k_{l_w} \geq 0 \qquad \forall w \in \Theta_u. \tag{4.43}$$

where constraints (4.39) and (4.40) ensure consistency between path flows and OD demand for OD pairs with fixed demand and elastic demand, respectively. Problem (4.38)–(4.43) is a UE problem with fixed demand for which there are efficient solution algorithms.

In what follows, we discuss the procedure for performing iteration k of the proposed allocation-design problem, and then provide the reader with details on the full implementation of the algorithm. We assume that iteration $k-1$ is finished successfully and regional decisions variables $\mathbf{y}^{k-1}_r$, $\mathbf{B}^{k-1}$, $\mathbf{x}^{k-1}_r$, and urban decision variables $\mathbf{y}^{k-1}_u$ and $\mathbf{x}^{k-1}_u$ are available for every urban city $u \in U$.

4.5.4 Iteration k of R-CNDP

At iteration k, the regional agent E_r decides about the capacity improvements for the regional links, $\mathbf{y}_r^k$, and the external budget assigned to urban cities, $\mathbf{B}^k$. Here we discuss the detail of SAB solution algorithm, and then describe the procedure to estimate the demand functions for urban cities.

SAB Algorithm for R-CNDP

Assuming the regional agent E_r moves from $\mathbf{y}_r^{k-1}$ and $\mathbf{B}^{k-1}$ along a direction $\mathbf{q}_r$, the rate of change of the value of cost function (4.28) may be written as:

$$DL_r^0(\mathbf{y}_r^{k-1}, \mathbf{B}^{k-1}; \mathbf{q}_r) = \left(\nabla_{\mathbf{y}_r} L_r^0(\mathbf{y}_r^{k-1}, \mathbf{B}^{k-1}); \nabla_{\mathbf{B}} L_r^0(\mathbf{y}_r^{k-1}, \mathbf{B}^{k-1})\right)^T \mathbf{q}_r \tag{4.44}$$

where $\nabla_{\mathbf{y}_r} L_r^0(\mathbf{y}_r^{k-1}, \mathbf{B}^{k-1}) = [\frac{\partial L_r^0(\mathbf{y}_r^{k-1}, \mathbf{B}^{k-1})}{\partial y_b}]$ and $\nabla_{\mathbf{B}} L_r^0(\mathbf{y}_r^{k-1}, \mathbf{B}^{k-1}) = [\frac{\partial L_r^0(\mathbf{y}_r^{k-1}, \mathbf{B}^{k-1})}{\partial B_{r,u}}]$ are gradient vectors with respect to upper-level decision variables $\mathbf{y}_r$ and $\mathbf{B}$ evaluated at $\mathbf{y}_r^{k-1}$ and $\mathbf{B}^{k-1}$ (for simplicity we dropped λ_r^k and c^k from the argument of the augmented Lagrangian function). A reasonable strategy is to move along the steepest descent direction and try to minimize the linearized problem, i.e.,

$$\min_{\|\mathbf{q}_r\| \leq 1} \quad DL_r^0(\mathbf{y}_r^{k-1}, \mathbf{B}^{k-1}; \mathbf{q}_r) = \left(\nabla_{\mathbf{y}_r} L_r^0(\mathbf{y}_r^{k-1}, \mathbf{B}^{k-1}); \nabla_{\mathbf{B}} L_r^0(\mathbf{y}_r^{k-1}, \mathbf{B}^{k-1})\right)^T \mathbf{q}_r \tag{4.45}$$

where $\|\cdot\|$ is some norm. Note that choice of Euclidean norm results in moving along the negative of gradient direction.

Differentiating (4.28) with respect to y_b, we get:

$$\frac{\partial L_r^0(\mathbf{y}_r^{k-1}, \mathbf{B}^{k-1})}{\partial y_b} = \frac{\partial F_r^0(\mathbf{y}_r^{k-1}, \mathbf{B}^{k-1})}{\partial y_b} + \max\{0, \lambda^{k-1} + c^{k-1} g_r(\mathbf{y}_r^{k-1}, \mathbf{B}^{k-1})\} \frac{\mathrm{d}G_b(y_b^{k-1})}{\mathrm{d}y_b} \tag{4.46}$$

where $\partial F_r^0(\mathbf{y}_r^{k-1}, \mathbf{B}^{k-1})/\partial y_b$ can be calculated be taking a derivative from (4.6) with respect to y_b:

$$\frac{\partial F_r^0(\mathbf{y}_r^{k-1}, \mathbf{B}^{k-1})}{\partial y_b} = \sum_{w \in W_r} d_w \frac{\partial T_w(\mathbf{x}_r^{k-1})}{\partial y_b} + \sum_{u \in U} \eta_u \frac{\partial R_u(B_{r,u}^{k-1})}{\partial y_b} \tag{4.47}$$

In equation (4.47), the derivatives $\partial T_w(\mathbf{x}_r^{k-1})/\partial y_b$ are implicit functions and can be calculated by performing the equilibrium sensitivity analysis on lower-level problem $P_r(1)$ (convex optimization problem formulated in (4.24)–(4.26).)

The derivative $\partial R_u(B_{r,u}^{k-1})/\partial y_b$ denotes the change in the objective value of CNDP of the urban city u due to a small perturbation in y_b, where b is a regional link. Using the chain rule we can write these derivatives as follows:

$$\frac{\partial R_u(B_{r,u}^{k-1})}{\partial y_b} = \sum_{\theta \in \Theta_u^r} \frac{\partial R_u(B_{r,u}^{k-1})}{\partial x_\theta} \frac{\partial x_\theta}{\partial y_b} \tag{4.48}$$

where $\partial x_\theta/\partial y_b$ can be computed by solving the problem (4.24)–(4.26) for y_b, i.e., $\partial x_\theta/\partial y_b = \alpha_\theta^b$. For derivative $\partial R_u(B_{r,u}^{k-1})/\partial x_\theta$ we can write (refer to (4.5)):

$$\frac{\partial R_u(B_{r,u}^{k-1})}{\partial x_\theta} = \frac{1}{F_u^0(B)} \frac{\mathrm{d}F_u^0(B_{r,u}^{k-1})}{\mathrm{d}x_\theta} \tag{4.49}$$

where $\mathrm{d}F_u^0(B_{r,u}^{k-1})/\mathrm{d}x_\theta$ indicates the change in the objective function value of the upper-level problem for urban city u for one unit change in regional flow, and can be computed by doing sensitivity analysis on U-CNDP of urban city u. Let $w \in \Theta_u$ be the urban OD pair represented by artificial link $\theta \in \Theta_u^r$. The change in x_θ can be translated as the change in $d_{r,w}$. Thus we get:

$$\begin{aligned} \frac{\mathrm{d}F_u^0(B_{r,u}^{k-1})}{\mathrm{d}x_\theta} &= \frac{\mathrm{d}F_u^0(B_{r,u}^{k-1})}{\mathrm{d}d_{r,w}} \\ &= d_w \frac{\partial T_w(\mathbf{x}_u^{k-1})}{\partial d_{r,w}} - D_{r,w}^{-1}(d_{r,w}^{k-1}) + T_w(\mathbf{x}_u^{k-1}) \end{aligned} \tag{4.50}$$

where the second line follows by taking a derivative from objective function (4.16) with respect to $d_{r,w}$, $d_{r,w}^{k-1}$ denotes the regional demand traveling between the endpoints of urban OD pair $w \in \Theta_u$ at iteration $k-1$, which is known, and he value of $\partial T_w(\mathbf{x}_u^{k-1})/\partial d_{r,w}$ can be estimated by solving the sensitivity analysis problem (4.24)–(4.26). Here we are assuming that change in $d_{r,w}$ will only impact those traveling between OD oar w.

Going back to problem (4.45), for the remaining components of the gradient vector we can write:

$$\frac{\partial L_r^0(\mathbf{y}_r^{k-1}, \mathbf{B}^{k-1})}{\partial B_{r,u}} = \frac{\partial F_r^0(\mathbf{y}_r^{k-1}, \mathbf{B}^{k-1})}{\partial B_{r,u}} + \max\{0, \lambda^{k-1} + c^{k-1} g_r(\mathbf{y}_u^{k-1})\} \tag{4.51}$$

where $\partial F_r^0(\mathbf{y}_r^{k-1}, \mathbf{B}^{k-1})/\partial B_{r,u}$ can be calculated by taking a derivative from (4.6) with respect to $B_{r,u}$:

$$\frac{\partial F_r^0(\mathbf{y}_r^{k-1}, \mathbf{B}^{k-1})}{\partial B_{r,u}} = \sum_{w \in W_r} d_w \frac{\partial T_w(\mathbf{x}_r^{k-1})}{\partial B_{r,u}} + \sum_{u \in U} \eta_u \frac{\mathrm{d}R_u(B_{r,u}^{k-1})}{\mathrm{d}B_{r,u}} \tag{4.52}$$

Recall that in the regional network, and as modeled in (4.23), the variable $B_{r,u}$ is a parameter of artificial links representing urban city u, i.e., Θ_u^r. Thus, the derivatives $\partial T_w(\mathbf{x}_r^k)/\partial B_{r,u}$ are implicit functions and can be calculated by performing the equilibrium sensitivity analysis on lower-level problem $P_r(1)$ with respect to $B_{r,u}$. Note that $B_{r,u}$ can be present in more than one link cost function, all artificial links θ representing urban city u, i.e., Θ_u^r, and any change in $B_{r,u}$ will be reflected on all these links.

The second unknown parameter in (4.52) is $\mathrm{d}R_\nu(B_{r,\nu}^{k-1})/\mathrm{d}B_{r,u}$ which can be formulated as (refer to (4.5)):

$$\frac{\mathrm{d}R_u(B_{r,u}^{k-1})}{\mathrm{d}B_{r,u}} = \frac{1}{F_u^0(B)} \frac{\mathrm{d}F_u^0(B_{r,u}^{k-1})}{\mathrm{d}B_{r,u}} \tag{4.53}$$

where $\mathrm{d}F_u^0(B_{r,u}^{k-1})/\mathrm{d}B_{r,u}$ indicates the change in the optimal value of the upper-level problem for urban city u for one unit change in $B_{r,u}$, and can be computed by doing sensitivity analysis on U-CNDP of urban city u. From (4.16), and for urban city u, we have:

$$\frac{\partial F_u^0(B_{r,u}^{k-1})}{\partial B_{r,u}} = \sum_{w \in W_u} d_w \frac{\partial T_w(\mathbf{x}_u^{k-1})}{\partial B_{r,u}} - \sum_{w \in \Theta_u} \left(D_{r,w}^{-1}(d_{r,w}^{k-1}) \frac{\partial d_{r,w}^{k-1}}{\partial B_{r,u}} - \frac{\partial d_{r,w}^{k-1}}{\partial B_{r,u}} T_w(\mathbf{x}_u^{k-1}) - d_{r,w}^{k-1} \frac{\partial T_w(\mathbf{x}_u^{k-1})}{\partial B_{r,u}} \right) \tag{4.54}$$

where furthermore we can write:

$$\begin{aligned} \frac{\partial d_{r,w}^{k-1}}{\partial B_{r,u}} &= \sum_{b \in H_u} \frac{\partial d_{r,w}^{k-1}}{\partial y_b} \frac{\partial y_b}{\partial B_{r,u}} \\ \frac{\partial T_w(\mathbf{x}_u^k)}{\partial B_{r,u}} &= \sum_{b \in H_u} \frac{\partial T_w(\mathbf{x}_u^{k-1})}{\partial y_b} \frac{\partial y_b}{\partial B_{r,u}} \end{aligned} \tag{4.55}$$

In (4.55), the values of $\partial d_{r,w}^{k-1}/\partial y_b$ and $\partial T_w(\mathbf{x}_u^{k-1})/\partial y_b$ can be computed by performing sensitivity analysis on the lower-level problem of urban city u, solving problem (4.24)–(4.26). For any OD pair $w \in$

$G(y_b) \cap \Theta_u$, we have:

$$\frac{\partial d_{r,w}^k(\mathbf{x}_u^{k-1})}{\partial y_b} = -\alpha_{l_w}^b \tag{4.56}$$

where $\alpha_{l_w}^b = \partial l_w / \partial y_b$ is the derivative of flow on artificial link l_w, introduced through Gartner's transformation, with respect to y_b.

To compute the value of $\partial y_b / \partial B_{r,u}$, we increment $B_{r,u}^{k-1}$ by one unit, i.e., $B_{r,u}^{k-1} := B_{r,u}^{k-1} + 1$, and then find the steepest descent direction for the upper-level problem at urban city u (the details of computing the steepest descent direction on urban city u is discussed in Section 4.5.5). Let $\mathbf{q}_u^*$ be the steepest descent direction, then we can write:

$$\frac{\partial y_b}{\partial B_{r,u}} = q_b^* \tag{4.57}$$

where q_b^* is the entry of $\mathbf{q}_u^*$ associated with y_b.

Let $\mathbf{q}_r^*$ denote the optimal solution to the linear program (4.45). The values of the decision variables at iteration k can be updated by taking a step along the descent direction $\mathbf{q}_r^*$ and then projecting the new point into feasible set defined by the bound constraints (4.29) and (4.30), i.e.,

$$[\mathbf{y}_r^k; \mathbf{B}^k] = \text{Proj}_\Omega([\mathbf{y}_r^{k-1}; \mathbf{B}^{k-1}] + \vartheta^k \mathbf{q}_r^*) \tag{4.58}$$

where ϑ^k is the step length along the steepest descent direction $\mathbf{q}_r^*$, $\Omega = \{\mathbf{y}_r, \mathbf{B} | 0 \leq y_b \leq \tilde{y}_b, \forall b \in H_r; 0 \leq B_{r,u} \leq \tilde{B}_{r,u}, \forall u \in U\}$ is the feasible set, and $\text{Proj}_\Omega(\mathbf{z}) = \arg\min_{\mathbf{y} \in \Omega} ||\mathbf{y} - \mathbf{z}||$ is the projection of $\mathbf{z}$ into set Ω. For feasible set Ω, the projection is easy and can be computed as follows:

$$\text{Proj}_\Omega(\mathbf{z}) = \max\{\mathbf{0}, \min\{\mathbf{z}, \tilde{\mathbf{y}}\}\} \tag{4.59}$$

where $\mathbf{0}$ is a vector of all zeros, and $\max$ and $\min$ operators are applied component-wise.

After updating the upper-level decision variables, equation (4.58), we solve lower-level problem $P_r(1)$ and compute the flow pattern over the regional network, i.e., $\mathbf{x}_r^k$. As the next step, and before solving the urban network design problems, we need to update the urban demand functions.

Updating Urban Demand Functions

The demand function for any OD pair $w \in \Theta_u$ represented by artificial link $\theta \in \Theta_u^r$ is formulated as a linear function of OD travel time (equation (4.15)):

$$d_{r,w}(\mathbf{x}_u) = d_{r,w}^k + \frac{\partial d_{r,w}^k}{\partial T_w}(T_w(\mathbf{x}_u) - T_w^k) \tag{4.60}$$

where $d_{r,w}^k = x_\theta^k$, $T_w^k = t_\theta(x_\theta^k)$, and $T_w(\mathbf{x}_u)$ is the travel time between of OD pair w, which plays the role of independent variable in this equation but depends on urban flow $\mathbf{x}_u$. To compute $\partial d_{r,w}^k/\partial T_w$, we perturb the travel cost on artificial link θ and compute the change in the regional demand going through artificial link θ. Let z_θ be the disturbance parameter. The modified cost function on artificial link θ would be $t_\theta(x_\theta) + z_\theta$. This way we can write:

$$\frac{\partial d_{r,w}^k}{\partial T_w} = \frac{\partial x_\theta^k}{\partial z_\theta} \tag{4.61}$$

which can be calculated by performing the equilibrium sensitivity analysis on lower-level problem $P_r(1)$, i.e., solving convex optimization problem (4.24)–(4.26) for $\epsilon = z_\theta$ and $\mathrm{d}t_\theta/\mathrm{d}\epsilon_\theta = 1$. Finally we may write:

$$\frac{\partial d_{r,w}^k}{\partial T_w} = \alpha_\theta^\theta \tag{4.62}$$

where $\alpha_\theta^\theta = \partial x_\theta^k/\partial z_\theta$ is the derivative of flow on artificial link θ with respect to z_θ.

After updating the urban demand functions, we need to solve U-CNDPs for each urban city u based on the new budget assignment $B_{r,u}^k$.

4.5.5 Iteration k of U-CNDPs

At iteration k, each urban agent E_u solves a CNDP with elastic demand on the urban city u based on the external budget $B_{r,u}^k$ assigned from the regional agent E_r. For regional decision variables $\mathbf{y}_r^k$, $\mathbf{B}^k$, and $\mathbf{x}_r^k$ at iteration k, solving the U-CNDP for each urban city u requires multiple iterations. Let $\bar{k}$ denote the index of internal iteration of U-CNDP at iteration k of the R-CNDP.

In the following sections, first we formulate the SAB solution algorithm for the U-CNDP of urban city u, and then describe the process for updating the artificial links in the regional network.

SAB Algorithm for U-CNDP

Suppose we have reached lower-level solution $(\mathbf{x}_u^{\bar{k}-1}, \mathbf{d}_u^{\bar{k}-1})$ for upper-level improvement choice $\mathbf{y}_u^{\bar{k}-1}$ at internal iteration $\bar{k}-1$ performed under iteration k of the R-CNDP. The steepest descent direction at internal iteration $\bar{k}$ can be formulated as the solution to the following linearized problem

$$\min_{\|\mathbf{q}_u\| \leq 1} \quad DL_u^0(\mathbf{y}_u^{\bar{k}-1}; \mathbf{q}_u) = \left(\nabla_{\mathbf{y}_u} L_u^0(\mathbf{y}_u^{\bar{k}-1})\right)^T \mathbf{q}_u \tag{4.63}$$

where $\|\cdot\|$ is some norm.

Differentiating (4.36) with respect to y_b, we get:

$$\frac{\partial L_u^0(\mathbf{y}_u^{\bar{k}-1})}{\partial y_b} = \frac{\partial F_u^0(\mathbf{y}_u^{\bar{k}-1})}{\partial y_b} + \max\{0, \lambda^{\bar{k}-1} + c^{\bar{k}-1} g_u(\mathbf{y}_u^{\bar{k}-1})\} \frac{\mathrm{d}G_b(y_b^{\bar{k}-1})}{\mathrm{d}y_b} \tag{4.64}$$

where $\partial F_u^0(\mathbf{y}_u^{\bar{k}-1})/\partial y_b$ can be calculated by taking a derivative from (4.16) with respect to y_b:

$$\begin{aligned}\frac{\partial F_u^0(\mathbf{y}_u^{\bar{k}-1})}{\partial y_b} = \sum_{w \in W_u} d_w \frac{\partial T_w(\mathbf{x}_u^{\bar{k}-1})}{\partial y_b} - \sum_{w \in \Theta_u} \Big(D_{r,w}^{-1}(d_{r,w}^{\bar{k}-1}) \frac{\partial d_{r,w}^{\bar{k}-1}(\mathbf{x}_u^{\bar{k}-1})}{\partial y_b} - \frac{\partial d_{r,w}^{\bar{k}-1}(\mathbf{x}_u^{\bar{k}-1})}{\partial y_b} T_w(\mathbf{x}_u^{\bar{k}-1}) \\ - d_{r,w}^{\bar{k}-1}(\mathbf{x}_u^{\bar{k}-1}) \frac{\partial T_w(\mathbf{x}_u^{\bar{k}-1})}{\partial y_b} \Big)\end{aligned} \tag{4.65}$$

where $d_{r,w}^{\bar{k}-1} = \bar{d}_{r,w}^{\bar{k}-1} - x_{l_w}^{\bar{k}-1}$ denotes the regional demand attracted to urban OD pair w. In equation (4.65), the derivatives $\partial T_w(\mathbf{x}_u^{\bar{k}-1})/\partial y_b$, for all OD pairs, and $\partial d_{r,w}^{\bar{k}-1}/\partial y_b$, for OD pairs with elastic demand, can be calculated by doing the equilibrium sensitivity analysis on lower-level problem $P_u(1)$.

Let $\mathbf{q}_u^*$ denote the optimal solution to the linear program (4.63). The value of decision variables at iteration $\bar{k}$ can be updated by taking a step along the descent direction $\mathbf{q}_u^*$ and then projecting the new point into feasible set defined by the bound constraints (4.35), i.e.,

$$\mathbf{y}_u^{\bar{k}} = \mathrm{Proj}_{\Omega}(\mathbf{y}_u^{\bar{k}-1} + \vartheta^{\bar{k}} \mathbf{q}_u^*) \tag{4.66}$$

where $\Omega = \{\mathbf{y}_u | 0 \leq y_b \leq \tilde{y}_b, \forall b \in H_u\}$ is the feasible set.

The next step after computing the vector of decision variables $\mathbf{y}_u^{\bar{k}}$ is to solve the lower-level problem $P_u'(0)$, which is a traditional UE problem, for $\mathbf{x}_u^{\bar{k}}$. Then we move to the next internal iteration $\bar{k}+1$ and solve the linearized problem (4.63) to compute $\mathbf{y}_u^{\bar{k}+1}$ from (4.66). This process is repeated until a measure

of convergence (based on the Euclidean norm of the gradient vector or changes in decision variables) is met (Please refer to Section 4.6 for more discussion on stopping criteria.)

Next we discuss how to update the artificial links in the regional network after solving the problems $P'_u(0)$ and $P'_u(1)$.

Updating Artificial Regional Links

Let $\mathbf{y}_u^k$ and $\mathbf{x}_u^k$ be the solutions to problems $P'_u(0)$ and $P'_u(1)$ after the stopping criterion of U-CNDP is met at some iteration $\bar{k}$ run under regional iteration k. At this point, we need to update the artificial links representing each urban city $u \in U$ in the regional network before solving R-CNDP for the next iteration, i.e., iteration $k+1$. As discussed in Section 4.4.2, each artificial link $\theta \in \Theta_u^r$ represents all used paths in the urban city u connecting its tail to its head with the following cost function at iteration $k+1$:

$$t_\theta^{k+1}(x_\theta) = \mu_\theta^k + \psi_{\theta,x}^k(x_\theta - x_\theta^k) + \psi_{\theta,B}^k(B_{r,u} - B_{r,u}^k), \qquad \forall \theta \in \Theta_u^r, u \in U \tag{4.67}$$

where x_θ and $B_{r,u}$ are decision variables adjusted at the regional level, $B_{r,u}^k$ is known from iteration k, and parameters μ_θ^k and x_θ^k can be directly computed from the solutions to problems $P'_u(0)$ and $P'_u(1)$ ($\mathbf{y}_u^k$ and $\mathbf{x}_u^k$):

$$\begin{aligned} \mu_\theta^k &= \sum_{\pi \in \hat{p}_w} \frac{f_\pi}{d_w + x_\theta} C_\pi \\ x_\theta^k &= \bar{d}_{r,w}^k - x_{l_w}^k \end{aligned} \tag{4.68}$$

where $w \in \Theta_u$ is the urban OD pair represented by artificial link θ. Here μ_θ^k is set as the average travel time of the paths represented by θ (weighted by flow), and x_θ^k is equal to the regional demand accommodated by OD pair w in urban city u.

For variable $\psi_{\theta,x}^k$ we have:

$$\begin{aligned} \psi_{\theta,x}^k &= \frac{\partial t_\theta^k}{\partial x_\theta} \\ &= \frac{\partial T_w(\mathbf{x}_u^k)}{\partial d_w} \end{aligned} \tag{4.69}$$

where $\partial T_w / \partial d_w$ can be estimated by sensitivity analysis problem described in Section 4.5.1.

The parameter $\psi_{\theta,B}^k$ determines how the travel time on link θ, or between OD pair w, is influenced

by external budget $B_{r,u}$, and can be formulated as:

$$\begin{aligned}\psi^k_{\theta,B} &= \frac{\partial t^k_\theta}{\partial B_{r,u}} \\ &= \frac{\partial T_w(\mathbf{x}^k_u)}{\partial B_{r,u}} \\ &= \sum_{b \in H_u} \frac{\partial T_w(\mathbf{x}^k_u)}{\partial y_b} \frac{\partial y_b}{\partial B_{r,u}}\end{aligned} \tag{4.70}$$

where the last equality is written using the chain rule, and the derivatives $\partial T_w / \partial y_b$ can be estimated from Section 4.5.1.

To compute the value of $\partial y_b / \partial B_{r,u}$, and similar to our discussion in Section 4.5.4, we increment $B^k_{r,u}$ by one unit, i.e., $B^k_{r,u} := B^k_{r,u} + 1$, and then solve the linear programming problem (4.63). Let $\mathbf{q}^*$ be the steepest descent direction, then we can write:

$$\frac{\partial y_b}{\partial B_{r,u}} = q^*_b \tag{4.71}$$

where q^*_b is the entry of $\mathbf{q}^*$ associated with y_b.

In summary, the procedure for solving the U-CNDP for urban city $u \in U$ and for an external budget $B^k_{r,u}$ at regional iteration k may be written as:

Algorithm 1: U-CNDP Pseudo-code

Step 0: Select initial values for decision variables $\mathbf{y}^0_u$, and set $\bar{k} = 0$.

Step 1: Solve the lower-level problem $P'_u(1)$ and get $\mathbf{x}^{\bar{k}}_u$ and $\mathbf{d}^{\bar{k}}_u$.

Step 2: Solve the linear programming problem (4.63), as a local linear approximation of the upper-level augmented objective function (4.34), to obtain $\mathbf{q}^*_u$.

Step 3: Move along the steepest descent direction $\mathbf{q}^*_u$: $\mathbf{y}^{\bar{k}+1}_u = \text{Proj}_\Omega(\mathbf{y}^{\bar{k}}_u - \vartheta^{\bar{k}} \mathbf{q}^*_u)$.

Step 4: Go to Step 5 if the convergence criterion is met, otherwise set $\bar{k} := \bar{k} + 1$ and go to Step 1.

Step 5: Let $\mathbf{y}^k = \mathbf{y}^{\bar{k}}$ and $\mathbf{x}^k = \mathbf{x}^{\bar{k}}$. Update parameters of artificial links in regional network, and call the R-CNDP solver.

Next we overview the allocation-design problem.

4.5.6 Algorithm Overview

Before implementing the discussed allocation-design problem we need to compute the value of $F_u^0(B)$ for every urban city $u \in U$, as required in (4.5). To this end, we solve the CNDP for every city $u \in U$ with the full budget, i.e. $B_{r,u} = B$. Let $\mathbf{y}_u^*$ denote the optimal improvements at city $u \in U$ for $B_{r,u} = B$, and $\mathbf{x}_u^* = \mathbf{x}(\mathbf{y}_\mathbf{u}^*)$ be the associated UE flow. We can write:

$$F_u^0(B) = \sum_{w \in W_u} d_w T_w(\mathbf{x}_u^*) \tag{4.72}$$

Note that $F_u^0(B)$ is the best that each city can obtain under maximal possible budget.

As the next step, we need to set up the problem by some initial budget allocation. There are multiple ways to initialize the urban budgets, but the measure selected here is to assign the budget proportional to the value of $F_u^0(B)$:

$$B_{r,y}^0 = B \frac{F_u^0(B)}{\sum_{v \in U} F_v^0(B)} \tag{4.73}$$

For this initial budget assignment, we solve the U-CNDP for each urban city $u \in U$ and then estimate the the artificial links in the regional network, as discussed in *Algorithm 1*.

After these initialization steps, the main loop starts by iteratively solving the R-CNDP on the regional network and updating the budget assignments, and then simultaneously solving the U-CNDPs on urban cities and updating the regional network for the next step. This process is repeated until a measure of convergence on both urban cities and regional network is reached.

The procedure for solving the proposed regional-urban network design problems is as follows:

Algorithm 2: Pseudo-code of the proposed allocation-design problem

Step 0: Initialization

- Step 0-1: Run Algorithm 1 for each urban city $u \in U$ for $B_{r,u} = B$, and compute $F_u^0(B)$ according to (4.72).
- Step 0-2: Allocate budget to urban cities according to (4.73), and run Algorithm 1 to set up the regional network.
- Step 0-3: Set $k = 0$.

Step 1: Solve the lower-level problem $P_r(1)$ and urban network design problems $P'_u(0)$ and $P'_u(1)$ for $\mathbf{x}_r^k$, $\mathbf{x}_u^k$, and $\mathbf{y}_u^k$ for every $u \in U$ until converged:

- Step 1-1: Solve the UE problem $P_r(1)$. Let the output be $\mathbf{x}_r^k$.
- Step 1-2: Update urban demand functions (Section 4.5.4).
- Step 1-3: Solve U-CNDP for each urban city u using SAB algorithm (Algorithm 1). Let the output be $\mathbf{y}_u^k$ and $\mathbf{x}_u^k$.
- Step 1-4: Update the artificial links (Section 4.5.5), and go to Step 1-1

Step 2: Update upper-level problem $P_r(0)$ (Section 4.5.4). Call them $\mathbf{y}_r^{k+1}$ and $\mathbf{B}^{k+1}$.

Step 3: Stop if the convergence criterion is met, otherwise set $k := k + 1$ and go to Step 1

4.6 Demonstration

This section deals with investigating the properties of the proposed network design problem. First we evaluate the quality of the solution generated by the proposed decentralized network design algorithm compared to the solution obtained by solving the network design problem on the full network (centralized implementation). Then we discuss the importance of modeling the interactions, evaluate the computational benefits of our algorithm, and demonstrate its advantages in addressing the design problems with conflicting objective functions.

The first case study includes two urban cities where each urban city is a modified version of the Sioux Falls network. The Sioux Falls network has 24 zones, 24 nodes, and 76 links [Bar-Gera, 2013]. Before

discussing the details of the decentralized implementation, we implement the network design problem with SAB algorithm on a single-network case study. Note that many instances of the single-network case, as discussed in Algorithm 2, needs to be solved at each iteration of the allocation-design problem after updating the regional decision variables. Figure 4.4 depicts a schematic of the Sioux Falls network and Table 4.1 describes the set of candidate links and their associated improvement cost function parameters. This is the same network used by Suwansirikul et al. [1987]. Each UE subproblem is solved to a relative gap of 1E−6, where relative gap is defined as:

$$\text{relative gap} = \frac{\sum_{w\in W}\sum_{\pi\in\hat{p}_w} h_\pi C_\pi - \sum_{w\in W}\sum_{\pi\in\hat{p}_w} h_\pi \kappa^w}{\sum_{w\in W}\sum_{\pi\in\hat{p}_w} h_\pi C_\pi} \tag{4.74}$$

where κ^w represents the time spent on the fastest path between OD pair w. As verified by Boyce et al. [2004], this relative gap is enough to ensure that traffic assignment is converged to a stable link flow solution.

The algorithm stops when the Euclidean norm of the gradient vector is less than 0.5. Table 4.2 displays the value of model parameters selected for this network.

Figure 4.5 shows the norm of the gradient vector and also the variation in the decision variables at each iteration, and Figure 4.6 plots the value of the objective function at each iteration relative to the optimal value (value upon convergence) and the allocated budget at each iteration of the algorithm (positive values indicate that more than available budget is allocated and negative values mean that the allocated budget is less than available budget). Except the first few iterations where the algorithm is still adjusting the stepsize, these values show a decreasing trend. In Figure 4.6-(a), we have values of the objective function lower than the optimal value. These solutions, as can be seen from Figure 4.6-(b), are infeasible points due to budget constraint violation. This is a property of the ALM algorithm: it allows infeasible solutions but penalizes them by increasing the penalty value until it converges to an optimal and feasible solution. Taking into account the results plotted in Figure 4.6, one can see that the solution has converged enough when the norm of the gradient vector is less than 10 or when the variation in decision variables is less than 1%. These values are used as stopping condition of the UNDPs when dealing with the decentralized implementation.

Next we implement the decentralized allocation-design problem on the test-bed composed of two urban cities where each urban city is a copy of the Sioux Falls network. Figure 4.7 shows the considered case study. The set of urban projects for each urban city includes all urban links with an improvement cost function of $G_a(y_a) = \zeta_a y_a^2$ where ζ_a is the cost coefficient. The cost coefficients are generated randomly from a uniform distribution with support of $[20, 40]$. In addition, the set of regional candidate links includes all 10 regional links with a improvement cost function similar to those of urban links.

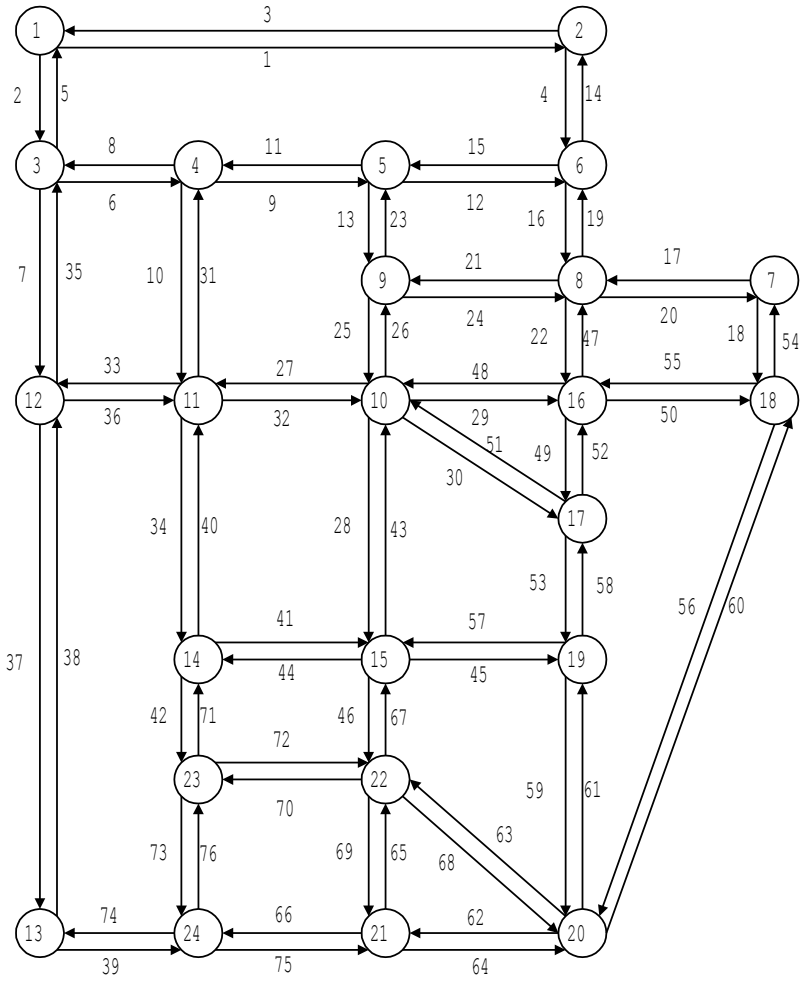
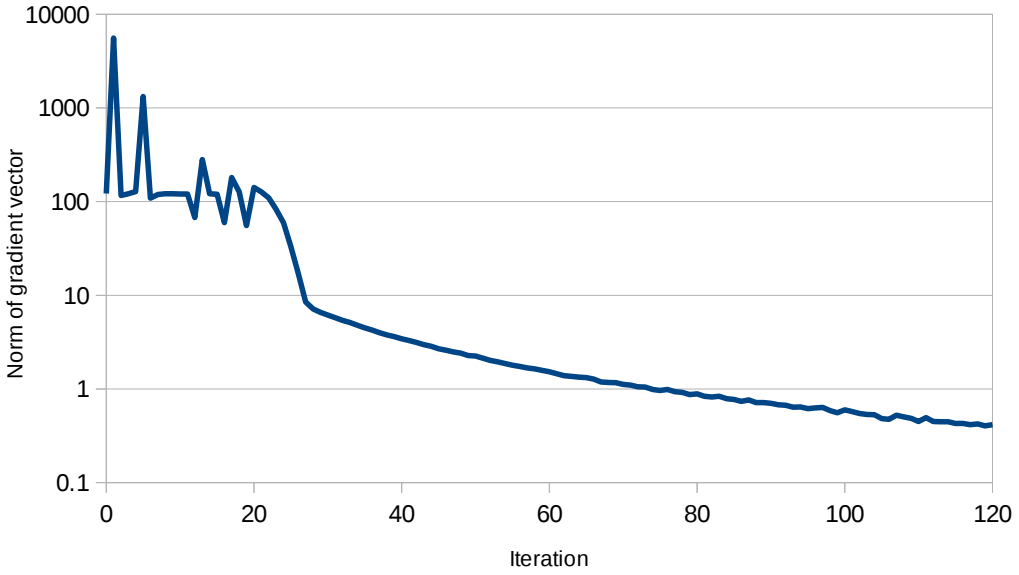
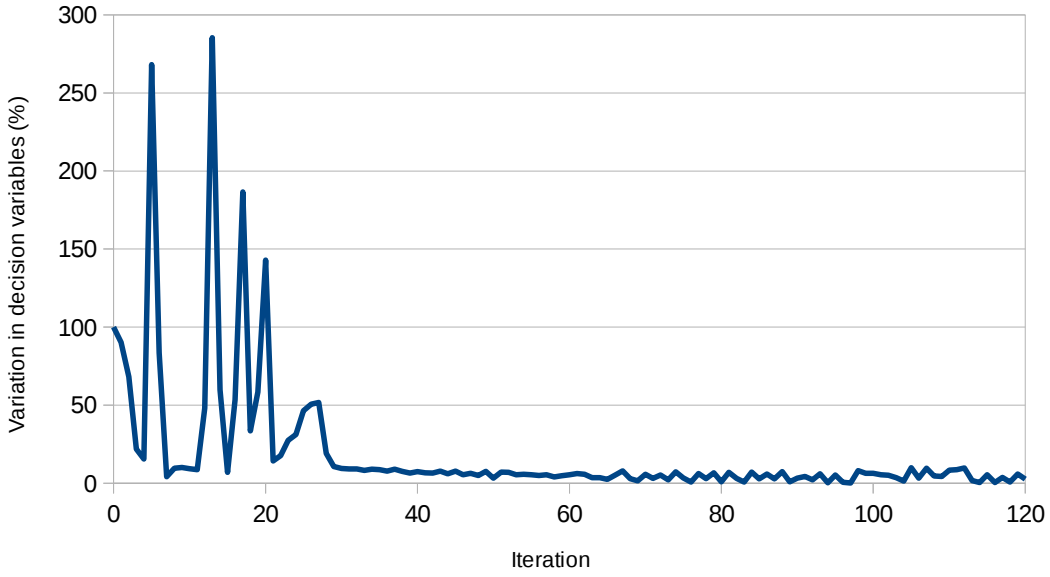


Figure 4.4: The Sioux Falls network used in hypothetical network shown in Figure 4.7

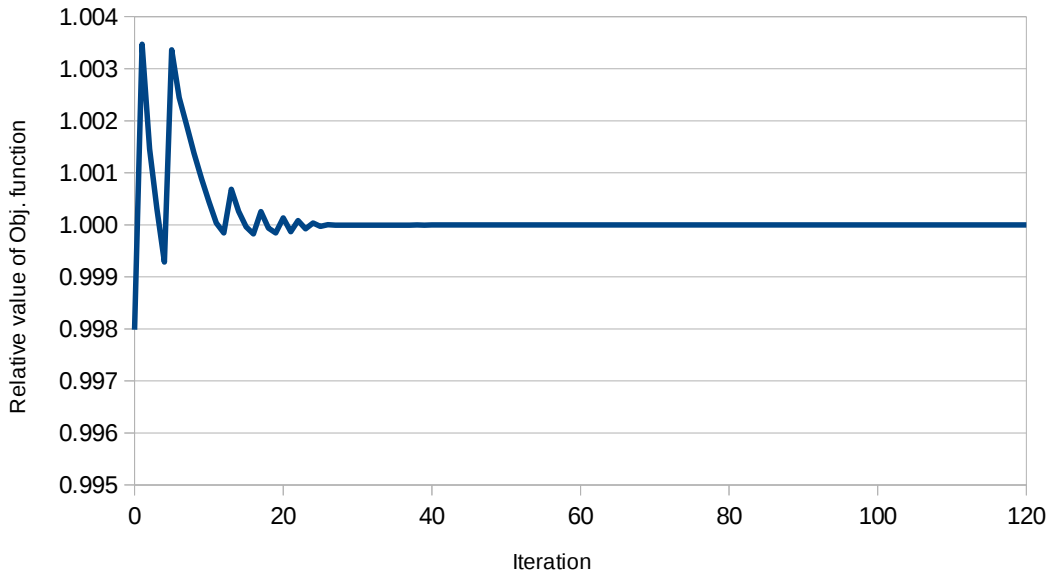


(a)

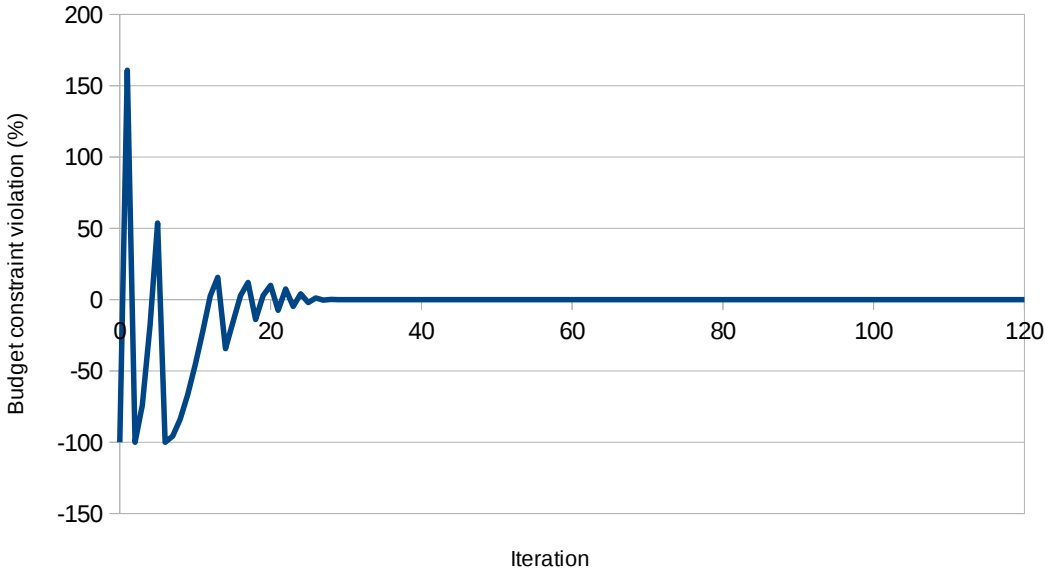


(b)

Figure 4.5: (a) The Euclidean norm of the gradient vector and (b) percentage variation of decision variables (compared to the previous iteration) (b).



(a)



(b)

Figure 4.6: (a) The value of the objective function relative to the optimal value and (b) budget violation (negative values indicate unallocated budget and positive values indicate budget constraint violation).

Table 4.1: The set of candidate links for Sioux Falls network and their improvement cost function parameters.

$G_a(y_a) = \zeta_a y_a^2$					
Links	16 and 19	17 and 20	25 and 26	29 and 48	30 and 75
ζ_a	26	40	25	48	34

Table 4.2: The simulation parameters.

c^0	λ^0	ρ	β	ϕ
1	1	1.0001	2	0.25

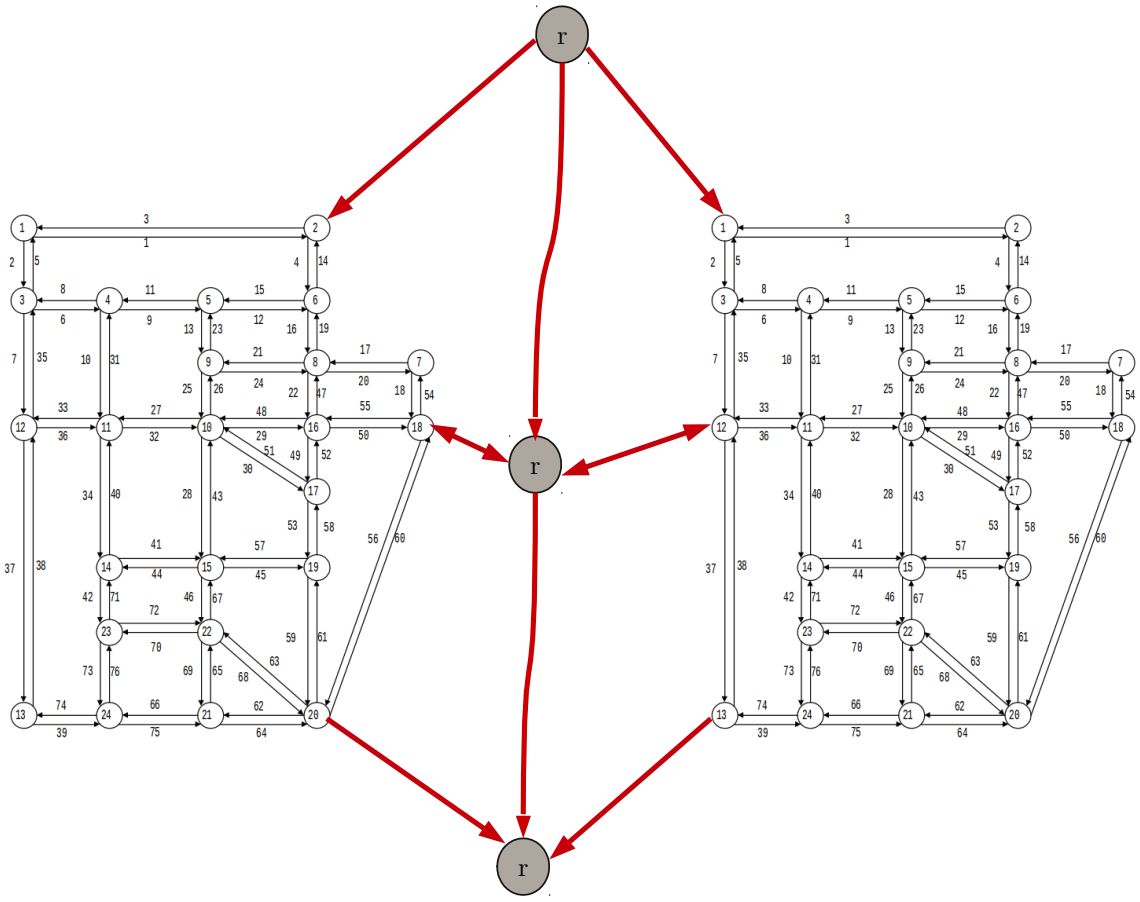


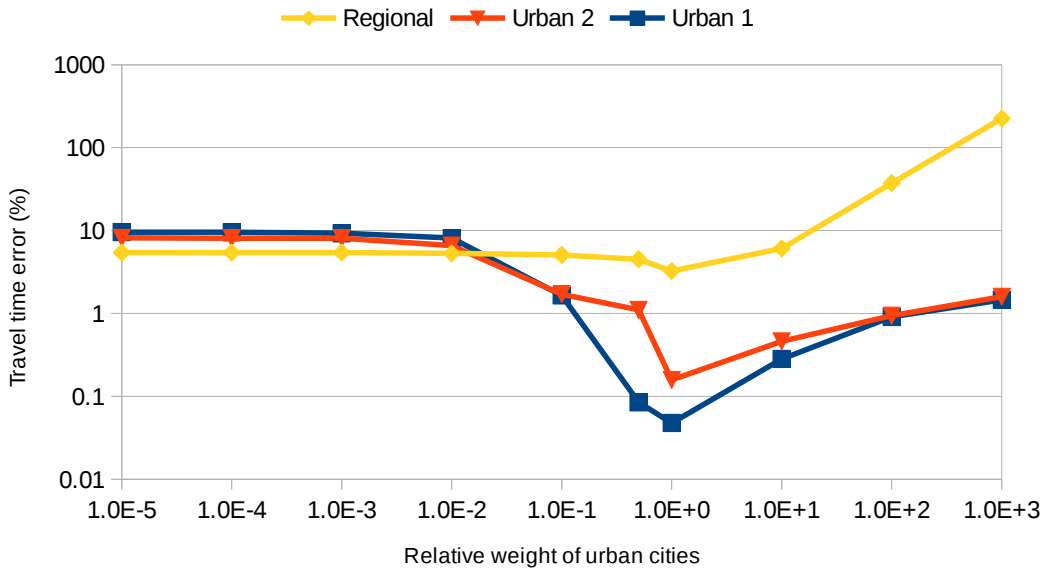
Figure 4.7: The hypothetical network composed of two copies of the Sioux Falls network.

First we solved the network design problem on the full network in a centralized implementation and then solved the same problem according to the proposed decentralized algorithm. In the decentralized algorithm, the weights assigned to urban cities in the objective function of the regional agent, η_u in (4.6), play an important role in the budget allocation. Figure 4.8 plots the results of decentralized algorithm compared with the centralized implementation. The horizontal axis in both figures is the value of η_u normalized by $F_u^0(B)$ (please refer to (4.5)) such that value of 1 means that the weight assigned to each urban city in (4.6) in equal to their best objective value under budget B, i.e., $\eta_u = F_u^0(B)$. Both the horizontal and vertical axes are shown in a logarithmic scale. The results plotted in Figure 4.8 show that the error in travel times and budget allocation is high for both low and high values of the relative urban weights, while for relative weight of 1, $\eta_u = F_u^0(B)$, the errors in travel times and allocated budget are minimum and close to those obtained under centralized implementation. In light of the urban regret functions, equation (4.5), the objective function of the regional agent, equation (4.6), may be written as:

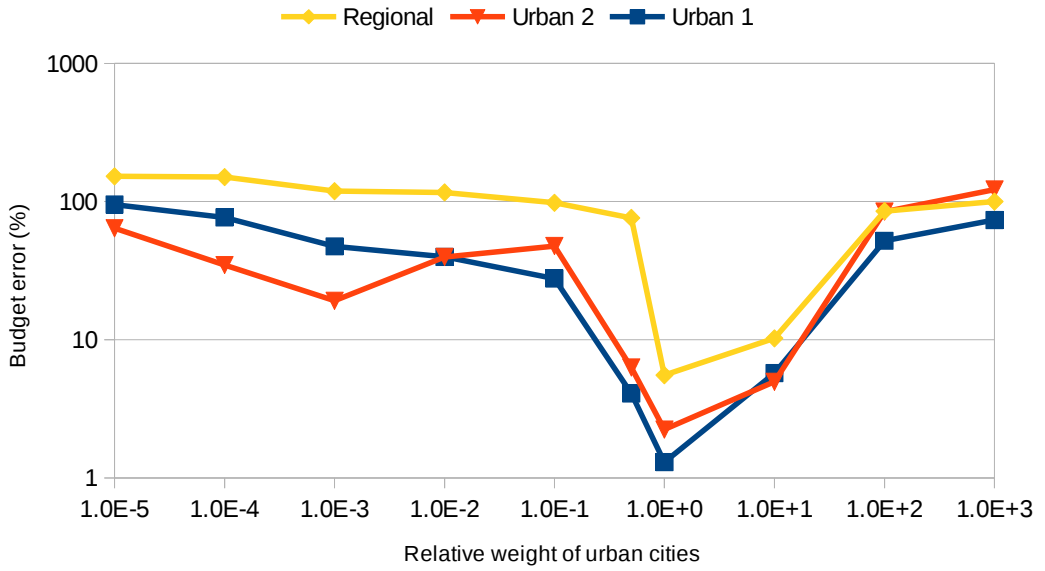
$$\begin{aligned}\min_{\mathrm{y}_r,\mathrm{B}} F_r^0(\mathrm{y}_r,\mathrm{B}) &= \sum_{w\in W_r} d_w T_w(\mathbf{x}_r) + \sum_{u\in U} \eta_u R_u(B_{r,u}) \\ &= \sum_{w\in W_r} d_w T_w(\mathbf{x}_r) + \sum_{u\in U} \eta_u \frac{F_u^0(B_{r,u}) - F_u^0(B)}{F_u^0(B)} \\ &= \sum_{w\in W_r} d_w T_w(\mathbf{x}_r) + \sum_{u\in U} \frac{\eta_u}{F_u^0(B)} F_u^0(B_{r,u}) - \sum_{\in U} \eta_u\end{aligned} \tag{4.75}$$

where η_u is fixed and $F_u^0(B_{r,u})$ is the objective of the urban agent E_u. One can see that selecting $\eta_u = F_u^0(B)$ assigns equal weight to the objective of regional and urban agents and converts the problem to that of the centralized formulation. On the other hand, for the choice of $\eta_u < F_u^0(B)$ the objective of the regional agent has a higher priority compared to those of urban agents and as a result more budget will be allocated to regional projects, and for the choice of $\eta_u > F_u^0(B)$ the urban objectives get the higher priority and their share from the total budget will increase. This is consistent with the findings in Figure 4.8.

To evaluate the importance of capturing the system-level impacts of the local planning decisions, and modeling the interactions between different urban regions, we consider the following scenarios. First, we solve the UE problem on the complete network, shown in Figure 4.7, to find the route choice of regional demand. This defines the external demand to urban cities. Then, we fix the urban external demands and solve CNDP for each urban city independently without modeling the interactions and global effects. For this case, the budget is equally split between two urban regions. We refer to this model as Scenario 1. As Scenario 2, we solve the CNDP by implementing the distributed scheme developed in this chapter. Under



(a)



(b)

Figure 4.8: (a) Error in travel time and (b) budget allocation as a function of weight of urban regret functions.

Scenario 2, the budget assigned to urban 1 (network on left in Figure 4.7) and urban 2 (network on right in Figure 4.7) are 52.5% and 47.5%, respectively. These values are very close to the split ratio applied under Scenario 1. The design decisions, however, are significantly different. Under Scenario 2, where interactions between urban cities and the route choice behavior of regional travelers are modeled, the total system travel time improved by 13%: 2.24% for urban 1, 2.23% for urban 2, and 52% for regional traffic. The improvement for regional traffic is more significant because their reaction was completely ignored under Scenario 1, while in Scenario 2 the local plans are developed by considering the behavior of regional traffic.

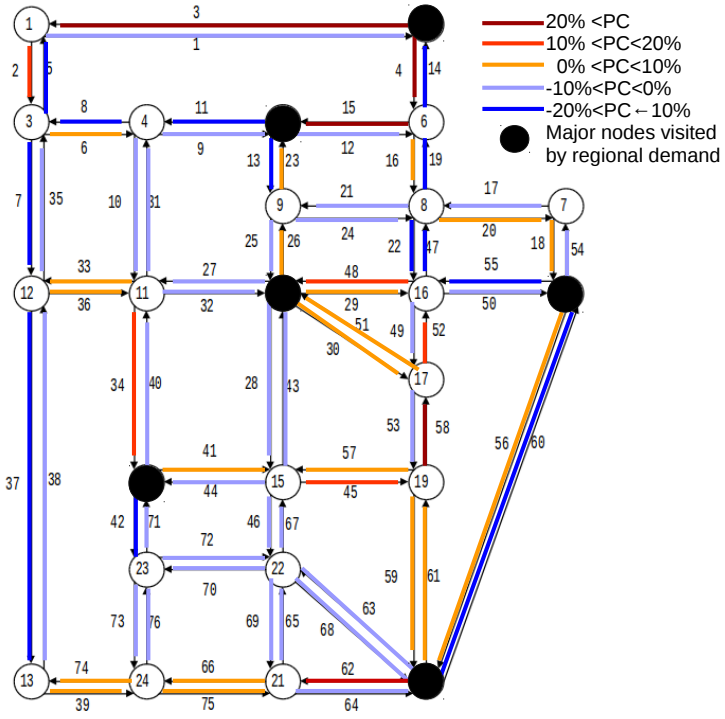
Let y_a^1 and y_a^2 denote the additional capacity assigned to link a under Scenarios 1 and 2, respectively. Figure 4.9 shows the percentage change (PC) in link improvement decision variables, defined as follows for link a:

$$PC(a) = 100\frac{y_a^2 - y_a^1}{y_a^1} \tag{4.76}$$

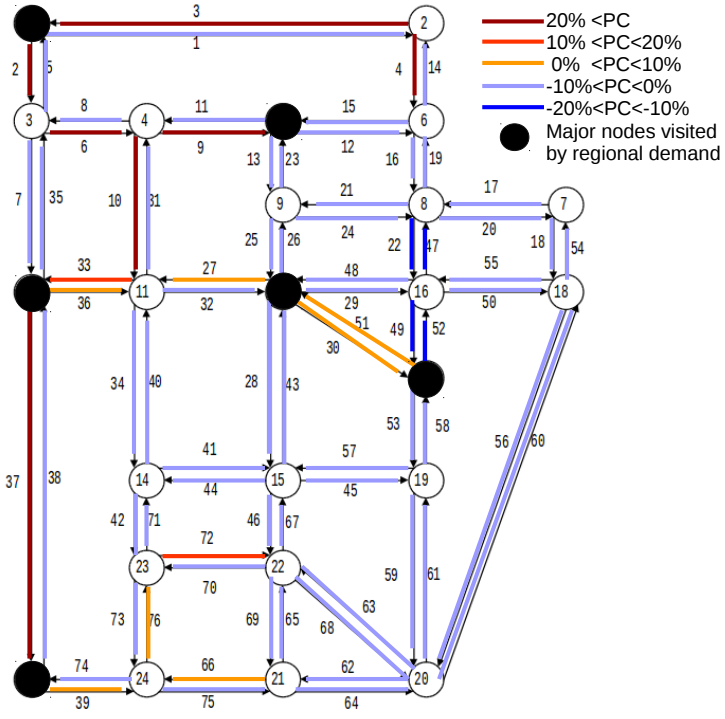
Figure 4.9 shows that mainly the links close to nodes with regional demand experience an increase in their capacity under Scenario 2. This is due to the fact that in Scenario 2 all interactions are modeled and decision variables will be adjusted to provide a global solution rather than a local solution.

The computational savings for this case study were not significant primarily because the networks are small and the master and subproblems are easy to solve. To evaluate the computational advantages of the proposed decentralized allocation-design problem, we tested our code on Austin regional network also used in Chapter 3. The northern subnetwork (north of the Colorado River) is treated as one urban city and the southern subnetwork (south of the river) form another urban city. The links connecting these two subnetworks are considered as regional links and their free flow travel time is scaled up such that the subnetwork stand independent from each other for the purpose of our problem. For each urban network, 10% of links are randomly selected as the urban projects while all the links connecting these two urban regions are subject to improvement. All urban and regional projects have an improvement cost function of $G_a(y_a) = \zeta_a y_a^2$ where ζ_a is selected from the interval $[20, 40]$. We solved the network design problem using both the centralized and decentralized implementations. The error in budget allocation and travel times were less than 4% and 0.06%, respectively, and the decentralized algorithm demonstrated a computational saving of 22%.

As discussed before, one of the main advantages of the proposed allocation-design problem is its capability to handle cases where different regions have different priorities. This essentially results in NDPs with different objectives. Examples of such cases can be NDPs with the objectives to minimize the travel time, pollution, network vulnerability, to maximize network safety, etc. Here, as a simple example, we just



(a)



(b)

Figure 4.9: Percentage change in link improvement decision variables for (a) urban 1, and (b) urban 2.

assume that all urban cities opt to minimize the system travel time, but different corridors have different weights. In this setting, each OD pair w is a assigned a weight γ_w indicating the importance of the travel time of users traveling between the endpoints of w. The objective function of urban cities, $P_u(0)$, may be written as:

$$P_u(0): \min_{\mathbf{y}_u} F_u^0(\mathbf{y}_u) = \sum_{w \in W_u} \gamma_w d_w T_w(\mathbf{x}_u) - \sum_{w \in \Theta_u} \gamma_w \Big(\int_0^{d_{r,w}} D_{r,w}^{-1}(\omega)\, d\omega - d_{r,w} T_w(\mathbf{x}_u) \Big) \tag{4.77}$$

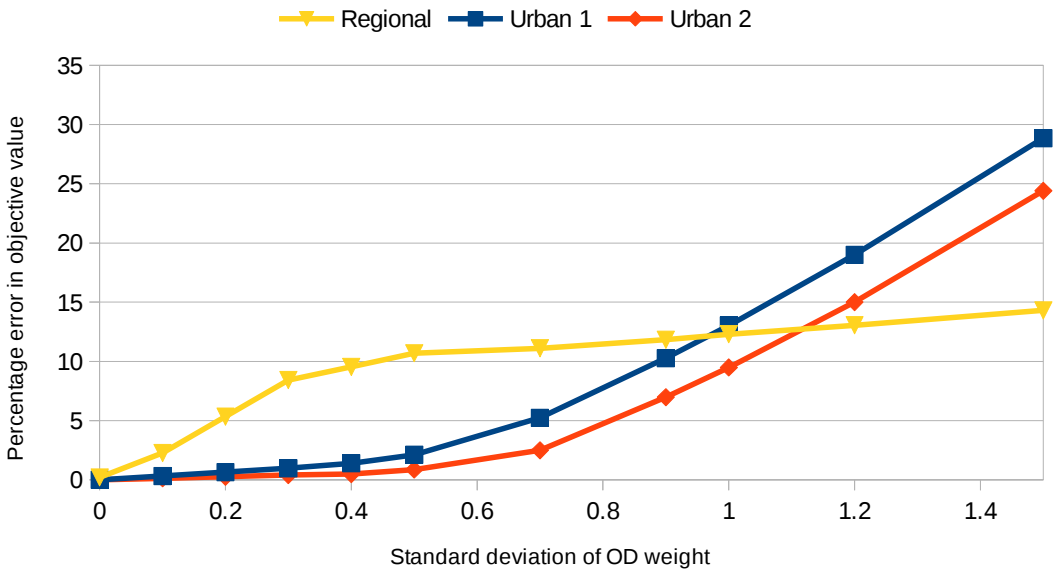
$$\text{subject to} (4.17) \text{ and } (4.18) \tag{4.78}$$

Figure 4.10 compares the difference between the objective values and budget of urban cities and regional network under centralized and decentralized implementations. Note that in the centralized case, all OD pairs have the same weight, i.e., $\gamma_w = 1$ for all urban and regional OD pairs. For decentralized implementation, we assume that OD weights come from a normal distribution with mean of 1 and different standard deviations ($\gamma_w \sim N(1, \sigma)$). The value of standard deviation (σ) indicates the variation of the objective from the standard case where all OD pairs have equal weights. As seen in Figure 4.10, errors are increasing functions of σ. This is reasonable: as the value of σ increases, deviation of the objective functions in decentralized implementation from the objective of the centralized model increases, and this results in NDPs with significantly different objectives, and, as a result, different design values.

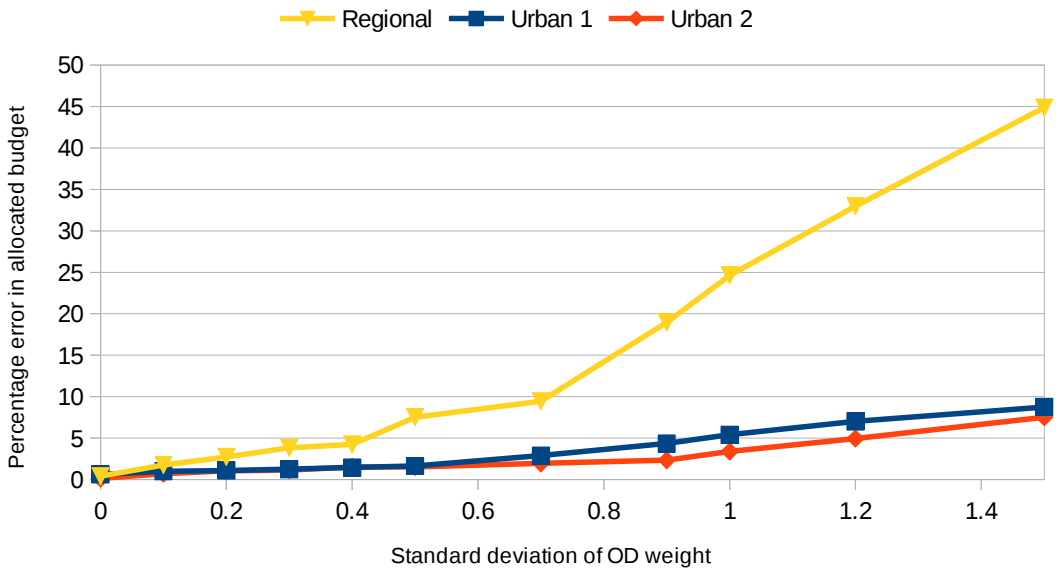
The values plotted in Figure 4.10 indicate how ignoring the concerns and priorities of the local regions for funding allocation and design decisions may result in sub-optimal plans for the system. Here still we assumed that all objectives are concerned with minimizng system travel time, while different corridors have different weights. The difference can be even more significant if objectives are distinct. For example, if a region is affected by high pollution or accident rates, implementing plans with the purpose of improving the traffic congestion may actually worsen the condition by attracting more demand to that region.

4.7 Conclusion

In this study, we developed a decentralized algorithm for network design problem based on the idea of the distributed problem solving approach. The algorithm allows the urban cities (subnetworks) to perform their design problem independently while a regional agent allocates budget to urban cities by taking into account its own priorities and the impact of its decision of the performance of the urban cities. The problem was formulated as a four-level network design problem and a solution algorithm based on the sensitivity analysis heuristic was developed to solve the problem. Our test results on a hypothetical network



(a)



(b)

Figure 4.10: (a) Error in objective value and (b) budget allocation as a function of standard deviation of OD weights.

composed of two copies of the Sioux Falls network shows that the centralized solution can be replicated by assigning a weight equal to the best urban objective to each urban city. Our findings also demonstrated the importance of modeling the interactions between different entities and considering the system-level impacts of local decisions. Simulations on the Austin regional network demonstrated a computational saving of 22% compared to the centralized algorithm.

An interesting topic for future research would be combining the decentralized network modeling and design techniques. Network design requires solving many instances of the traffic assignment problem and the idea of distributed network modeling, developed in Chapter 3, can be used to reduce the complexity of this step. In addition, the focus of this chapter was on continuous instances of network design problem. Extending the model to handle discrete and mixed design problems is another interesting topic for future research.

5 Conclusion

5.1 Summary

This dissertation developed distributed algorithms for network modeling and design problems. The main idea behind distributed problem solving approach is to partition the problem into smaller subproblems and solve them locally by exchanging information between them. After the subproblems are solved, the set of local solutions are synthesized to obtain a solution to the original problem.

In Chapter 2, a new algorithm was proposed to solve the user equilibrium sensitivity analysis problem. The proposed algorithm resembles the traditional user equilibrium problem on a modified network where link performance functions are replaced with their derivatives evaluated at the user equilibrium flow. Compared to earlier bush-based sensitivity algorithms, the proposed method does not require a planarity assumption and is more stable numerically. We evaluated the validity of the derivatives on Barcelona and Austin regional networks. It was shown that the contracted network, a simplified version of the complete network in which OD paths are represented by a single link with parameters tuned according to the sensitivity analysis, can approximate the behavior of the complete network with a high accuracy. The contracted network calibrated based on the proposed user equilibrium sensitivity analysis algorithm only has to be constructed once, and can be used to evaluate the network performance for different demand scenarios easily with a good approximation.

The proposed network contraction technique then was used to develop a spatially decomposition

algorithm for traffic assignment problem in Chapter 3. The proposed algorithm, named DSTAP, partitions the network into non-overlapping subnetworks and distributes the assignment task between a master problem and subproblems. The master problem solves the assignment problem on the master network, a simplified version of the complete network in which all sub-networks are replaced with some artificial links approximating the travel times between the subnetwork OD pairs. The parameters of these artificial links are estimated by solving the sensitivity analysis problem on subnetworks. Each subproblem is responsible for solving the assignment problem over one subnetwork equipped with some artificial links to model the paths outside of the subnetwork boundaries. The artificial links and regional demand assigned to subproblems are updated iteratively until a measure of convergence based on the maximum excess cost is obtained. We proved that DSTAP is an exact algorithm, and provided bounds on the maximum excess cost of the flow assignment on the complete network. The proposed bound is a function of maximum excess costs on the master problem and subproblems of DSTAP. Natural applications for the DSTAP algorithm are networks with clear boundaries which can be partitioned easily. Examples of such networks are statewide or national assignment problems, or cities with rivers or other geographic features.

In Chapter 4, as the second part of this dissertation, we developed a distributed algorithm for the network design problem. The proposed decentralized algorithm, referred to as allocation-design problem, was concerned with allocating funding between different urban cities and some regional projects. The urban cities have full authority over their jurisdiction, and the regional agent may influence the urban design projects indirectly through the funding allocated to them. The problem was formulated as a four-level network design problem, and a solution algorithm based on a sensitivity analysis-base heuristic was developed to solve the problem. We discussed how the proposed decentralized network design algorithm can replicate the network design problem over the complete network, and showed the computational advantages of the decentralized algorithm. Also the advantage of the proposed algorithm in capturing the system-level effects of the local decisions was numerically illustrated.

5.2 Future Work

The work presented in this dissertation can be extended in different ways. In the case of the DSTAP algorithm, we assumed that the network is already partitioned into subnetworks. Defining the partitions, however, is a major question for future research. Partitioning the network into smaller subnetworks would result in smaller subproblems, but it may increase the size of the master network, as number of artificial links and OD pairs are expected to be an increasing function of number of partitions. This, in fact, may hinder the computational advantages of the DSTAP algorithm. It would be interesting to study this prop-

erty more rigorously and investigate if there is an optimal number of partitions which would result in the best run time. Defining a proper objective function seems to play a critical role for the partitioning task. The following objectives may be considered: (1) to minimize the number of boundary points, (2) to minimize interactions between subnetwork, and (3) to minimize the size of regional OD matrix. By combining the partitioning and DSTAP algorithms, we can automate the whole process: the combined algorithm would partition the network and solve the assignment problem over the subnetworks and master network such that the run time is minimized.

Combining the decentralized network modeling and design techniques developed in this work is also a topic of interest for future research. Network design requires solving many instances of the traffic assignment problem, and the idea of distributed network modeling, developed in Chapter 3, may be used to reduce the complexity of this step. In addition, the focus of this chapter was on continuous instances of network design problems. Extending the model to handle discrete and mixed design problems is worth investigation.

Another interesting and important extension of the current work would be extending the idea of distributed modeling for dynamic models, which incorporate traffic flow dynamics. Despite much research on dynamic models in recent decades, model complexity and computational burden for large-scale networks are still obstacles to deployment in practice. In this dissertation and for the static case, interactions between different players were modeled by introducing some artificial links with parameters tuned based on linearizations to the equilibrium solution for each sub-network. Perhaps the same idea can be extended for dynamic models: each sub-network can be equipped with artificial links representing a simple and aggregate model of neighboring sub-networks. Each artificial link relates the mean-space flow, speed, and density of the aggregated area (there is a direct connection between these artificial links and macroscopic fundamental diagram (MFD) discussed in the literature in recent years). Developing efficient algorithms to estimate the parameters of the MFDs governing the artificial links, and evaluating the performance of the proposed modeling approach are major tasks to be studied.

References

M. Abdulaal and L. J. LeBlanc. Continuous equilibrium network design models. *Transportation Research Part B: Methodological*, 13(1):19–32, 1979.

R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

R. E. Allsop. Some possibilities for using traffic control to influence trip distribution and route choice. In *Transportation and traffic theory, proceedings*, volume 6, 1974.

J. X. Ban, H. X. Liu, M. C. Ferris, and B. Ran. A general mpcc model and its solution algorithm for continuous network design problem. *Mathematical and Computer Modelling*, 43(5):493–505, 2006.

H. Bar-Gera. Origin-based algorithm for the traffic assignment problem. *Transportation Science*, 36(4): 198–417, 2002.

H. Bar-Gera. Primal method for determining the most likely route flows in large road networks. *Transportation Science*, 40(3):269–286, 2006.

H. Bar-Gera. Traffic assignment by paired alternative segments. *Transportation Research Part B: Methodological*, 44(8):1022–1046, 2010.

H. Bar-Gera. Transportation test problems, April 2013. URL http://www.bgu.ac.il/~bargera/tntp/.

H. Bar-Gera, F. Hellman, and M. Patriksson. Computational precision of traffic equilibria sensitivities in automatic network design and road pricing. *Transportation Research Part B: Methodological*, 57(4): 485–500, 2013.

J. F. Bard. An algorithm for solving the general bilevel programming problem. *Mathematics of Operations Research*, 8(2):260–272, 1983.

R. R. Barton, D. W. Hearn, and S. Lawphongpanich. The equivalence of transfer and generalized benders decomposition methods for traffic assignment. *Transportation Research Part B: Methodological*, 23(1): 61–73, 1989.

M. S. Bazaraa, H. D. Sherali, and C. M. Shetty. *Nonlinear programming: theory and algorithms*. John Wiley & Sons, 2013.

M. J. Beckmann, C. B. McGuire, and C. B. Winsten. Studies in the economics of transportation. Yale University Press, New Haven, 1956.

D. P. Bertsekas. On the Goldstein-Levitin-Polyak gradient projection method. *Automatic Control, IEEE Transactions on*, 21(2):174–184, 1976.

D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999.

P. H. L. Bovy and G. R. M. Jansen. Network aggregation effects upon equilibrium assignment outcomes. *Transportation Science*, 17(3):240–262, 1983.

D. Boyce, K. S. Chon, M. E. Ferris, Y. J. Lee, K. T. Lin, and R. W. Eash. Implementation and evaluation of combined models of urban travel and location on a sketch planning network. Technical report, University of Illinois at Urbana-mpaign, 1985.

D. Boyce, B. Ralevic-Dekic, and H. Bar-Gera. Convergence of traffic assignments: how much is enough? *Journal of Transportation Engineering*, 130(1):49–55, 2004.

S. D. Boyles. Bush-based sensitivity analysis for approximating subnetwork diversion. *Transportation Research Part B: Methodological*, 46(1):139–155, 2012.

S. D. Boyles. Improved bush-based sensitivity analysis in network equilibrium. In *Transportation Research Board 92nd Annual Meeting*, number 13-4519, 2013.

S. D. Boyles, A. Voruganti, and S. Waller. Quantifying distributions of freeway operational metrics. *Transportation Letters*, 3(1):21–36, 2011.

D. Braess. Uber ein Paradoxon aus der Verkehrsplanung. *Unternehmensforschung*, 12(1):258–268, 1969.

R. Cassidy, M. Kirby, and W. Raike. Efficient distribution of resources through three levels of government. *Management Science*, 17(8):B–462, 1971.

Y. Chan. A method to simplify network representation in transportation planning. *Transportation Research*, 10(3):179–191, 1976.

K. Chang, Z. Khatib, and Y. Ou. Effects of zoning structure and network detail on traffic demand modeling. *Environment and Planning B*, 29(1):37–52, 2002.

R. Chen and R. R. Meyer. Parallel optimization for traffic assignment. *Mathematical Programming*, 42 (1-3):327–345, 1988.

S.-W. Chiou. Optimization of area traffic control for equilibrium network flows. *Transportation Science*, 33(3):279–289, 1999.

S.-W. Chiou. Bilevel programming for the continuous transport network design problem. *Transportation Research Part B: Methodological*, 39(4):361–383, 2005.

S.-W. Chiou. A generalized iterative scheme for network design problem. *Applied Mathematics and Computation*, 188(2):1115–1123, 2007.

S.-W. Chiou. A hybrid approach for optimal design of signalized road network. *Applied Mathematical Modelling*, 32(2):195–207, 2008.

S.-W. Chiou. A subgradient optimization model for continuous road network design problem. *Applied Mathematical Modelling*, 33(3):1386–1396, 2009.

H. Cho, T. E. Smith, and T. L. Friesz. A reduction method for local sensitivity analyses of network equilibrium arc flows. *Transportation Research Part B: Methodological*, 34(1):31–51, 2000.

R. D. Connors and D. P. Watling. Assessing the demand vulnerability of equilibrium traffic networks via network aggregation. *Networks and Spatial Economics*, pages 1–29, 2014.

O. Damberg and A. Migdalas. Distributed disaggregate simplicial decomposition—a parallel algorithm for traffic assignment. In *Network optimization*, pages 172–193. Springer, 1997.

G. B. Dantzig, R. P. Harvey, Z. F. Lansdowne, D. W. Robinson, and S. F. Maier. Formulating and solving the network design problem by decomposition. *Transportation Research Part B: Methodological*, 13(1): 5–17, 1979.

R. Davis and R. G. Smith. Negotiation as a metaphor for distributed problem solving. *Artificial intelligence*, 20(1):63–109, 1983.

S. Dempe and A. B. Zemkoho. Bilevel road pricing: theoretical analysis and optimality conditions. *Annals of Operations Research*, 196(1):223–240, 2012.

S. Dempe, V. Kalashnikov, G. A. Pérez-Valdés, and N. Kalashnykova. Bilevel programming problems. *Energy Systems. Springer, Berlin*, 2015.

R. B. Dial. *Probabilistic assignment: A multipath traffic assignment model which obviates path enumeration*. PhD thesis, University of Washington, Seattle, Washington, 1970.

R. B. Dial. Minimal-revenue congestion pricing part I: A fast algorithm for the single-origin case. *Transportation Research Part B: Methodological*, 33(3):189–202, 1999a.

R. B. Dial. Accurate traffic equilibrium: how to bobtail frank-wolfe. Technical report, Volpe National Transportation Research Center, Cambridge, MA, 1999b.

R. B. Dial. A path-based user-equilibrium traffic assignment algorithm that obviates path storage and enumeration. *Transportation Research Part B: Methodological*, 40(10):917–936, 2006.

L. Dubkin, I. Rabinovich, and I. Vakhutinsky. *Iterative aggregation theory*. Marcel Dekker, Inc., 1987.

R. W. Eash, K. S. Chon, Y. J. Lee, and D. Boyce. Equilibrium traffic assignment on an aggregated highway network for sketch planning. *Transportation Research Record*, 944:30–37, 1983.

T. A. Edmunds and J. F. Bard. Algorithms for nonlinear bilevel mathematical programs. *IEEE transactions on Systems, Man, and Cybernetics*, 21(1):83–89, 1991.

J. E. Falk and J. Liu. On bilevel programming, part i: general nonlinear cases. *Mathematical Programming*, 70(1-3):47–72, 1995.

R. Z. Farahani, E. Miandoabchi, W. Y. Szeto, and H. Rashidi. A review of urban transportation network design problems. *European Journal of Operational Research*, 229(2):281–302, 2013.

M. Florian, I. Constantin, and D. Florian. A new look at projected gradient method for equilibrium assignment. *Transportation Research Record: Journal of the Transportation Research Board*, 2090:10–16, 2009.

M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3 (1-2):95–110, 1956.

T. L. Friesz. Transportation network equilibrium, design and aggregation: key developments and research opportunities. *Transportation Research Part A: General*, 19(5):413–427, 1985.

T. L. Friesz and P. T. Harker. Properties of the iterative optimization-equilibrium algorithm. *Civil Engineering Systems*, 2(3):142–154, 1985.

T. L. Friesz, R. L. Tobin, H.-J. Cho, and N. J. Mehta. Sensitivity analysis based heuristic algorithms for mathematical programs with variational inequality constraints. *Mathematical Programming*, 48(1-3): 265–284, 1990.

T. L. Friesz, G. Anandalingam, N. J. Mehta, K. Nam, S. J. Shah, and R. L. Tobin. The multiobjective equilibrium network design problem revisited: a simulated annealing approach. *European Journal of Operational Research*, 65(1):44–57, 1993.

R. G. Gallager. A minimum delay routing algorithm using distributed computation. *Communications, IEEE Transactions on*, 25(1):73–85, 1977.

Z. Gao, J. Wu, and H. Sun. Solution algorithm for the bi-level discrete network design problem. *Transportation Research Part B: Methodological*, 39(6):479–495, 2005.

N. H. Gartner. Optimal traffic assignment with elastic demands: a review part ii. algorithmic approaches. *Transportation Science*, 14(2):192–208, 1980.

G. Gentile. Local user cost equilibrium: a bush-based algorithm for traffic assignment. *Transportmetrica A: Transport Science*, 10(1):15–54, 2014.

A. E. Haghani and M. S. Daskin. Network design application of an extraction algorithm for network aggregation. *Transportation Research Record*, 944:37–46, 1983.

M. A. Hall. Properties of the equilibrium state in transportation networks. *Transportation Science*, 12(3): 208–216, 1978.

P. T. Harker and T. L. Friesz. Bounding the solution of the continuous equilibrium network design problem. In *Proceedings of the Ninth International Symposium on Transportation and Traffic Theory*, pages 233–252. VNU Science Press, 1984.

D. W. Hearn. Practical and theoretical aspects of aggregation problems in transportation planning. *Transportation Planning Models*, pages 257–287, 1984.

R. Hooke and T. A. Jeeves. "direct search"solution of numerical and statistical problems. *Journal of the ACM (JACM)*, 8(2):212–229, 1961.

A. J. Horowitz. *Statewide travel forecasting models*, volume 358. Transportation Research Board, 2006.

Y. Ishizuka and E. Aiyoshi. Double penalty method for bilevel optimization problems. *Annals of Operations Research*, 34(1):73–88, 1992.

E. Jafari and S. D. Boyles. Improved bush-based methods for network contraction. *Transportation Research Part B: Methodological*, 83:298–313, 2016.

E. Jafari, M. Gemar, N. Ruiz-Juri, and J. Duthiel. An investigation of centroid connector placement for advanced traffic assignment models with added network detail. *Accepted for Publication in Transportation Research Record: Journal of the Transportation Research Board*, 2015.

R. Jayakrishnan, W. Tsai, J. Prasker, and S. Rajadhyakfchsha. A faster path-based algorithm for traffic assignment. *Transportation Research Record*, 1443:75–83, 1994.

J. Jeon, S.-Y. Kho, D.-K. Kim, and J.-S. Lee. Interactions of aggregated zoning and network systems: A case study of seoul city. *Journal of the Eastern Asia Society for Transportation Studies*, 8:404–419, 2010.

M. Josefsson and M. Patriksson. Sensitivity analysis of separable traffic equilibrium equilibria with application to bilevel optimization in network design. *Transportation Research Part B: Methodological*, 41 (1):4–31, 2007.

A. Karakitsiou, A. Mavrommati, and A. Migdalas. Efficient minimization over products of simplices and its application to nonlinear multicommodity network problems. *Operational Research*, 4(2):99–118, 2004.

C. D. Kolstad and L. S. Lasdon. Derivative evaluation and computational experience with large bilevel mathematical programs. *Journal of Optimization Theory and Applications*, 65(3):485–499, 1990.

R. N. Lass, J. B. Kopena, E. A. Sultanik, D. N. Nguyen, C. P. Dugan, P. J. Modi, and W. C. Regli. Coordination of first responders under communication and resource constraints. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 3*, pages 1409–1412. International Foundation for Autonomous Agents and Multiagent Systems, 2008.

L. J. LeBlanc, E. K. Morlok, and W. P. Pierskalla. An efficient approach to solving the road network equilibrium traffic assignment problem. *Transportation Research*, 9(5):309–318, 1975.

C. Li, H. Yang, D. Zhu, and Q. Meng. A global optimization method for continuous network design problems. *Transportation Research Part B: Methodological*, 46(9):1144–1158, 2012.

P. A. Lotito. Issues in the implementation of the DSD algorithm for the traffic assignment problem. *European journal of operational research*, 175(3):1577–1587, 2006.

S. Lu. Sensitivity of static traffic user equilibria with perturbations in arc cost function and travel demand. *Transportation science*, 42(1):105–123, 2008.

S. Lu and Y. Nie. Stability of user-equilibrium route flow solutions for the traffic assignment problem. *Transportation Research Part B: Methodological*, 44(4):609–617, 2010a.

S. Lu and Y. M. Nie. Stability of user-equilibrium route flow solutions for the traffic assignment problem. *Transportation Research Part B: Methodological*, 44(4):609–617, 2010b.

P. Luathep, A. Sumalee, W. H. Lam, Z.-C. Li, and H. K. Lo. Global optimization method for mixed transportation network design problem: a mixed-integer linear programming approach. *Transportation Research Part B: Methodological*, 45(5):808–827, 2011.

Z.-Q. Luo, J.-S. Pang, and D. Ralph. *Mathematical Programs with Equilibrium Constraints*. Cambridge University Press, 1996.

T. V. Mathew and S. Sharma. Capacity expansion problem for large urban transportation networks. *Journal of Transportation Engineering*, 135(7):406–415, 2009.

Q. Meng and H. Yang. Benefit distribution and equity in road network design. *Transportation Research Part B: Methodological*, 36(1):19–35, 2002.

Q. Meng, H. Yang, and M. Bell. An equivalent continuously differentiable model and a locally convergent algorithm for the continuous network design problem. *Transportation Research Part B: Methodological*, 35(1):83–105, 2001.

M. Mitradjieva and P. O. Lindberg. The stiff is moving-conjugate direction frank-wolfe methods with applications to traffic assignment. *Transportation Science*, 47(2):280–293, 2013.

T. Miyagi and T. Suzuki. A ramsey price equilibrium model for urban transit system: A bilevel programming approach with transportation network equilibrium constraints. volume 2: Modelling transport systems. In *World Transport Research. Proceedings of the 7th World Conference on Transport Research*, 1996.

Y. Nie. A class of bush-based algorithms for the traffic assignment problem. *Transportation Research Part B: Methodological*, 44(1):73–89, 2010.

T. Oates, M. N. Prasad, and V. R. Lesser. Cooperative information-gathering: a distributed problem-solving approach. In *Software Engineering. IEE Proceedings-[see also Software, IEE Proceedings]*, volume 144, pages 72–88. IET, 1997.

M. Patriksson. Sensitivity analysis of traffic equilibria. *Transportation Science*, 38(3):258–281, 2004.

M. J. Powell. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The computer journal*, 7(2):155–162, 1964.

Y. Qiu and T. L. Magnanti. Sensitivity analysis for variational inequalities defined on polyhedral sets. *Mathematics of Operations Research*, 14(3):410–432, 1989.

D. F. Rogers, R. D. Plante, R. T. Wong, and J. R. Evans. Aggregation and disaggregation techniques and methodology in optimization. *Operations Research*, 39(4):553–582, 1991.

H. Sbayti, M. El-Fadel, and I. Kaysi. Effect of roadway network aggregation levels on modeling of traffic-induced emission inventories in beirut. *Transportation Research Part D: Transport and Environment*, 7 (3):163–173, 2002.

P. A. Steenbrink. *Optimization of Transport Networks*. J. Wiley and Sons Limited, 1974.

C. Suwansirikul, T. L. Friesz, and R. L. Tobin. Equilibrium decomposed optimization: a heuristic for the continuous equilibrium network design problem. *Transportation science*, 21(4):254–263, 1987.

R. L. Tobin and T. L. Friesz. Sensitivity analysis for equilibrium network flow. *Transportation Science*, 22(4):242–250, 1988.

S. TXDOT. Texas Statewide Analysis Model- Third Version SAM-V3 2013. Technical report, Texas Department of Transportation, October 2013. Model development, validation report and User manual.

H. Von Stackelberg. *The theory of the market economy*. Oxford University Press, 1952.

D. Z. Wang and H. K. Lo. Global optimum of the linearized network design problem with equilibrium flows. *Transportation Research Part B: Methodological*, 44(4):482–492, 2010.

S. Wang, Q. Meng, and H. Yang. Global optimization methods for the discrete network design problem. *Transportation Research Part B: Methodological*, 50:42–60, 2013.

S. Wong and H. Yang. Reserve capacity of a signal-controlled road network. *Transportation Research Part B: Methodological*, 31(5):397–402, 1997.

I. Wright, Y. Xiang, L. Waller, J. Cross, E. Norton, and D. Van Vliet. The practical benefits of the saturn origin-based assignment algorithm and network aggregation techniques. *European Transport Conference*, 2010.

C. Xie, K. M. Kockelman, and S. T. Waller. Maximum entropy method for subnetwork origin-destination trip matrix estimation. *Transportation Research Record: Journal of the Transportation Research Board*, 2196(1):111–119, 2010.

C. Xie, K. M. Kockelman, and S. T. Waller. A maximum entropy-least squares estimator for elastic origin–destination trip matrix estimation. *Transportation Research Part B: Methodological*, 45(9):1465–1482, 2011.

H. Yang. Sensitivity analysis for the elastic-demand network equilibrium problem with applications. *Transportation Research Part B: Methodological*, 31(1):55–70, 1997.

H. Yang and M. G. Bell. Traffic restraint, road pricing and network equilibrium. *Transportation Research Part B: Methodological*, 31(4):303–314, 1997.

H. Yang and M. G. Bell. Transport bilevel programming problems: recent methodological advances. *Transportation Research Part B: Methodological*, 35(1):1–4, 2001.

H. Yang and M. G. Bell. Sensitivity analysis of network traffic equilibrium revisited: The corrected approach. *Mathematics in Transport (B. Heydecker, ed.)*, pages 373–411, 2007a.

H. Yang and M. G. Bell. Sensitivity analysis of network traffic equilibrium revisited: The corrected approach. In *4th IMA International Conference on Mathematics in Transport*, 2007b.

H. Yang and S. Yagar. Traffic assignment and signal control in saturated road networks. *Transportation Research Part A: Policy and Practice*, 29(2):125–139, 1995.

H. Yang, T. Sasaki, Y. Iida, and Y. Asakura. Estimation of origin-destination matrices from link traffic counts on congested networks. *Transportation Research Part B: Methodological*, 26(6):417–434, 1992.

H. Yang, S. Yagar, Y. Iida, and Y. Asakura. An algorithm for the inflow control problem on urban freeway networks with user-optimal flows. *Transportation Research Part B: Methodological*, 28(2):123–139, 1994.

H. Yang, X. Zhang, and Q. Meng. Modeling private highways in networks with entry–exit based toll charges. *Transportation Research Part B: Methodological*, 38(3):191–213, 2004.

N. D. Yen. Lipschitz continuity of solutions of variational inequalities with a parametric polyhedral constraint. *Mathematics of Operations Research*, 20(3):695–708, 1995.

W. Yeoh and M. Yokoo. Distributed problem solving. *AI Magazine*, 33(3):53, 2012.

Y.-x. Yuan. A review of trust region algorithms for optimization. In *ICIAM*, 2000.

W. I. Zangwill. *Nonlinear programming: a unified approach*, volume 196. Prentice-Hall Englewood Cliffs, NJ, 1969.

X. Zhou, S. Erdogan, and H. Mahmassani. Dynamic origin—destination trip demand estimation for subarea analysis. *Transportation Research Record: Journal of the Transportation Research Board*, (1964): 176–184, 2006.

P. H. Zipkin. Bounds for aggregating nodes in network problems. *Mathematical programming*, 19(1): 155–177, 1980.

R. Zivan, S. Okamoto, and H. Peled. Explorative anytime local search for distributed constraint optimization. *Artificial Intelligence*, 212:1–26, 2014.

G. Ziyou and S. Yifan. A reserve capacity model of optimal signal control with user-equilibrium route choice. *Transportation Research Part B: Methodological*, 36(4):313–323, 2002.