ABSTRACT

Mehrzad Mehrabipour. Traffic Congestion Management in Transportation Networks through Dynamic Traffic Assignment. (Under the direction of Dr. Ali Hajbabaie).

Dynamic Traffic Assignment (DTA) is used in a wide range of applications such as congestion pricing, traffic operations, network design, evacuation planning, traffic management systems, and policy evaluations. DTA determines time-dependent link/path flows to optimize a system-level or user-level objective considering demands for origin-destination pairs. The goal of this dissertation is to develop algorithms to solve a dynamic traffic assignment problem and present an optimization-based approach to manage congestion at bottlenecks.

We first present a decomposition scheme to solve a System Optimal (SO) DTA problem with multiple origin-destination pairs in urban transportation networks. The decomposition is designed based on the Dantzig-Wolfe decomposition principle that splits the set of constraints into sub-sets through the construction of a master problem and a set of sub-problems. Each origin-destination pair is represented by an independent sub-problem with a less computational burden compared to the original problem. The master problem captures the coordination between sub-problems. The proposed methodology is implemented in two case study networks including 20 and 40 intersections. The numerical results show that the decomposition scheme converges to the optimal solution, within a 2% gap, in 22-99% less time compared to a benchmark solution capable of finding optimal solutions. This result confirms the computational efficiency of the proposed methodology.

We then present a distributed gradient-based approach to solve the SODTA problem formulated based on the cell transmission model network loading concept. Centralized algorithms to solve such a problem do not scale well with the size of the network and the number of origin-destination

pairs. While the existing decomposition algorithms scale better with the size of the problem, they still have centralized components that eventually limit their scalability. The proposed algorithm distributes the SODTA formulation into local sub-problems that find optimal values for their decision variables within an intersection region. Each sub-problem communicates with its immediate neighbors to reach a consensus on the values of common decision variables. A sub-problem receives proposed values for common decision variables from all adjacent sub-problems and incorporates them in its own offered values by weighted averaging and enforcing a gradient step to minimize its own objective function. Then, the updated values are projected onto the feasible region of the sub-problems. We prove that the proposed algorithm converges to the optimal solution of the SODTA problem in an infinite number of iterations. The algorithm is also tested on two case study networks with 20 and 40 intersections under different demand levels. The approach finds solutions with at most a 5% optimality gap with a 97% shorter runtime compared to a benchmark that finds optimal solutions. The runtime of this approach is improved by 77% with 48% fewer variables compared to the approach in the first study of the dissertation.

We next introduce an algorithm to find near-optimal solutions for the SODTA problem with multiple origins and destinations in real-time. The proposed distributed optimization and coordination algorithm decomposes the network-level traffic assignment problem into several intersection-level sub-problems that can be efficiently solved individually. The approach also uses a rolling horizon technique for temporal decomposition. As a result of this decomposition, the complexity of the problem is reduced, and the solutions can be found in real-time. The sub-problems coordinate their decisions by exchanging information with other sub-problems and push the solutions toward global optimality. The approach is tested on a case study of 20 and 40 intersections. The results are compared with a benchmark approach capable of finding the optimal solutions with a 3% maximum optimality gap. The proposed solution technique finds the solutions in real-time because the solutions are generated in less time than the duration of each time step. The algorithm is also compared with the Method of Successive Averages (MSA) and Projection Algorithm (PA) on the Nguyen-Dupuis network. MSA and PA achieved a 0.17-4.29% relative gap in 8-200 iterations in different scenarios. Our algorithm generates solutions for each time step in real-time and achieved 0.00% optimality for all scenarios.

Lastly, we present a nonlinear mathematical formulation and a solution technique for a bottleneck congestion management approach. In this approach, travelers submit their willingness to pay value for using certain parts of a transportation system with limited roadway capacity. The first part of formulation determines the allocation of travelers who submit their willingness to pay to network roads to minimize total system travel time and maximize revenue. The second part of the formulation is a user equilibrium DTA modeled with a variational inequality approach. This part models the behavior of other travelers against the allocation by a system manager in the first part. The demand for paths in the second part of the formulation is determined based on the allocation in the first part, and, thus, the decision of travelers on route choices is a factor of assignments in the first part of the formulation. The objective function of the first part of formulation is also a function of UEDTA traffic flows. A heuristic solution technique is developed using projection and genetic algorithms. The solution technique is tested on case studies with 20 and 40 intersections. The objective of the first part of formulation is improved by 15-95% over iterations under different demand profiles. The proposed approach can find solutions with at most a 3-5% gap with SODTA as a lower bound.

Traffic Congestion Management in Transportation Networks through Dynamic Traffic Assignment

by
Mehrzad Mehrabipour

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Civil Engineering

Raleigh, North Carolina
2022

Approved by

Dr. Ali Hajbabaie
Committee Chair

Dr. Leila Hajibabai

Dr. George List

Dr. Gnanamanikam Mahinthakumar

Dr. Billy Williams

BIOGRAPHY

Mehrzad Mehrabipour received her B.Sc. degree in Industrial Engineering from Shahid Bahonar University, Kerman, Iran, in 2012, and his M.Sc. degree in Industrial Engineering from Tarbiat Modares University, Tehran, Iran in 2014. She received a second M.Sc. degree in Civil Engineering from Washington State University in 2018. She received her Ph.D. in Civil Engineering from North Carolina State University.

ACKNOWLEDGEMENTS

I Would like to thank my adviser for his guidance, advice, and support throughout this dissertation. I would also like to extend my thanks to my dissertation committee members Dr. George List, Dr. Billy Williams, Dr. Leila Hajibabai, and Dr. Gnanamanikam Mahinthakumar for their support, time, and valuable feedback. My sincere gratitude to my family, friends, and mentors.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1. INTRODUCTION

1.1. Background

Merchant and Nemhauser (1978a;1978b) studied a Dynamic Traffic Assignment (DTA) problem for the first time about forty years ago. DTA is a well-studied research area to determine time-dependent traffic flows. The objective function can be the minimization of the system cost or individual users' cost, and the problems are called System Optimal (SO) DTA and User Equilibrium (UE) DTA, respectively. DTA deployment has brought many benefits to a wide range of applications over the past decades. Network design (Karoonsoontawong and Waller, 2010), traffic operations (Beard and Ziliaskopoulos, 2006), congestion pricing (Shen and Zhang, 2009), evacuation planning (Shen et al., 2007a), and traffic management systems (Peeta and Bulusu, 1999) are some of DTA applications. DTA often has many decision variables and constraints to encompass its spatial/temporal scales and the number of Origin-Destination (OD) pairs. The number of decision variables and constraints may further increase based on the utilized network loading concept. For instance, a Cell Transmission Model (CTM)-based DTA will have more decision variables and constraints than a link performance function-based DTA, as CTM divides each link into several homogenous cells. Therefore, rather than having a decision variable for each link at each time step, several decision variables are required.

According to Lo & Szeto (2002), Szeto & Lo (2004), Ukkusuri & Waller (2008), Ukkusuri et al. (2012), and Doan & Ukkusuri (2015), formulating DTA problems using the CTM network loading concept improves accuracy in modeling traffic dynamics compared to link exit functions (Ban et al., 2008; Carey, 1992, 1987; Friesz et al., 1989; Merchant and Nemhauser, 1978a; Wie et al., 1994) and point queue models (Chow, 2009; Doan et al., 2011; Ramadurai et al., 2010). However, it increases computational complexity due to significant growth in the number of decision variables and the inclusion of constraints with the min(.) function to ensure that traffic flow follows the fundamental diagram. Several studies (Beard and Ziliaskopoulos, 2006; Ziliaskopoulos, 2000) reduced the complexity of problems by linearizing these constraints, which may lead to the flow holding back problem: traffic flow does not follow the fundamental diagram (Carey and Subrahmanian, 2000; Doan and Ukkusuri, 2012; Nie, 2011).

The existing state-of-the-art solution techniques to solve CTM-based SODTA are either optimization or heuristic-based. The optimization-based approaches do not scale well with the size of networks and become computationally intractable. This is mostly due to the large number of decision variables that are included in SODTA problems (Aziz and Ukkusuri, 2012; Beard and Ziliaskopoulos, 2006; Li et al., 2003; Lin et al., 2011b). Optimization-based approaches can be categorized into centralized and decomposition approaches.

Central optimization frameworks do not scale well with the size of DTA, especially when more complicated network loading concepts, such as CTM are in use (Aziz and Ukkusuri, 2012; Beard and Ziliaskopoulos, 2006; Chiu and Zheng, 2007; Zheng and Chiu, 2011; Ziliaskopoulos, 2000). While centralized approaches have the premise of finding globally optimal solutions, they are not scalable. In fact, the largest CTM-based SODTA that is solved in the literature has less than half a million decision variables (Zheng and Chiu 2011).

Decomposition approaches offer great potential to address the scalability issue of central approaches by partitioning the problem into two or more sub-problems, each with a smaller set of decision variables, that are computationally less challenging than the original problem. However, the existing decomposition techniques for CTM-based SODTA problems are structured for only one OD pair and require significant changes to allow multiple OD pairs (Lin, Valsaraj, and Waller 2011) or have a restrictive assumption on the ratio of free-flow speed to backward propagation speed and cannot provide efficient convergence for cases with multiple OD pairs (Li, Waller, and Ziliaskopoulos 2003). Moreover, (non-spatial) decomposition approaches scale better than centralized approaches; however, they face computational complexity growth in the sub-problems as the size of the original problem grows. Besides, all existing decomposition techniques require solving a central problem (often referred to as a master problem) to aggregate the solutions found by sub-problems, and expanding the network size will increase the complexity of the master problem. Therefore, while they can provide optimality bounds and scale better than the centralized approaches, they eventually become intractable with the size of the problem (Jafari et al. 2017; Larsson and Patriksson 1992; Larsson et al. 2004; Li et al. 2003; Lin et al. 2011; Mehrabipour et al. 2019).

The heuristic approaches scale with the size of networks; however, they require a long run-time to find solutions and the quality and optimality of solutions remain unknown. Moreover, these approaches are mainly path-based, and they need to find the marginal costs of paths and links in a network to satisfy the equilibrium condition of SODTA which may lead to finding sub-optimal solutions and long run-time.

Another subject that we have studied in this dissertation is congestion pricing. Congestion pricing can alleviate traffic congestion and generate revenue for the maintenance and construction of transportation networks. Congestion pricing techniques that are widely used can be classified into tolling through currency and tolling through tokens. Tolling through dollars is a form of price regulation that requires information on travelers' demand, the value of time, arrival time, etc. to determine toll levels. This information is not accessible or very hard to obtain. The lack of accurate or complete information may lead to the failure of tolling systems through currency (Pandey et al., 2020; Schade and Schlag, 2003; Sumalee, 2001). Tolling through tokens is a form of quantity regulation that has shown more public acceptance, and equity, and can react to economic changes more rapidly, but it has still some major drawbacks. How distribute tokens among travelers can have a significant effect on the system and is not straightforward to determine. Also, the usage of tokens in a cycle can affect traffic conditions adversely (Kockelman and Kalmanje, 2005; Wu et al., 2012; Yang and Wang, 2011). A congestion pricing system that creates market competition can overcome the drawbacks of tolling systems through dollars and tokens.

1.2. Problem Statement

Dynamic traffic assignment has been researched extensively since its introduction by Merchant and Nemhauser (1978a, 1978b). Ziliaskopoulos (2000) introduced a linear program for SODTA, which was based on CTM (Daganzo, 1995, 1994a). Various researchers (e.g., Lo and Szeto, 2002; Ukkusuri and Waller, 2008) have shown that utilizing the CTM network loading concept in DTA formulation improves accuracy in modeling traffic flow dynamics compared to link exit functions (e.g., Ban et al., 2008; Carey, 1992, 1987; Friesz et al., 1989; Merchant and Nemhauser, 1978a; Wie et al., 1994) and point queue models (e.g., Chow, 2009; Doan et al., 2011; Ramadurai et al., 2010). However, the additional accuracy comes with an increase in computational complexity due to significant growth in the number of decision variables and constraints.

Existing centralized approaches to solve the CTM-based SODTA problem can find optimal solutions to the problem in theory; however, do not scale well with the size of the problem and become intractable in medium-sized problems (e.g., Beard and Ziliaskopoulos, 2006; Chiu and Zheng, 2007; Zheng and Chiu, 2011; Ziliaskopoulos, 2000). Decomposition approaches are used to address the computational complexity issue by decomposing the formulation into two or more sub-problems since they will have fewer decision variables and constraints compared to the original formulation. However, these approaches need to optimize a master problem to coordinate the actions of sub-problems (e.g., Larsson et al., 2004; Li et al., 2003; Lin et al., 2011; Mehrabipour et al., 2019). The computational complexity of the master problem increases over iterations as new extreme points generated by sub-problems are added to the master problem. In addition, the number of decision variables of each sub-problem increases with the size of the network. Even though the spatial decomposition introduced by Jafari et al. (2017) faces complexity growth neither in the master problem over iterations nor in the sub-problems by adding links and nodes, adding more nodes and links to the network makes solving the master problem very challenging.

The current heuristics like the Method of Successive Averages (MSA) (Qian et al., 2012; Sbayti et al., 2007; Tajtehranifard et al., 2018), projection approach (Doan and Ukkusuri, 2015; Lu et al., 2016), and proximal point algorithm (Zhan and Ukkusuri, 2019) lack temporal and spatial decomposition schemes to reduce the complexity of CTM-based SODTA problems. These approaches force limitations (e.g., number of paths), especially when the number of variables increases. These approaches are also iterative techniques that cannot generate solutions in real-time. They mostly focus on finding path-marginal costs which is a computationally expensive step and requires generating and sorting paths that lead to a limited number of paths for each OD. Therefore, proposing methodologies to address the discussed drawbacks is essential. These methodologies should be able to determine routes of vehicles with a reasonable amount of time and memory while capturing traffic flow more realistically.

We also study congestion pricing techniques to overcome the drawbacks of tolling systems through dollars and tokens. In current congestion pricing techniques, accuracy and accessibility of information on travelers' demand, the value of time, arrival time, etc. plays an important role to determine toll levels. Thus, the lack of accurate or complete information may lead to the failure of current tolling systems through dollars. In tolling systems through tokens, the main problems are the determination of tokens distribution and adverse effects on traffic conditions.

1.3. Research Summary

The first study develops an analytical decomposition technique to tackle the intractability of CTM-based SODTA featured with multiple OD pairs. The proposed methodology applies to spatially and temporally larger transportation networks and still provides solutions that are appropriate for operational applications. We employ the Dantzig-Wolfe technique, which constructs a Master Problem (MP) and a set of Sub-Problems (SPs), where each SP represents a single OD pair. This configuration makes SODTA's search space more manageable compared to previous efforts in the literature. The independency of SPs makes the problem well-suited for parallelism, which can expand the applicability of the proposed methodology to large-scale networks. The SPs are solved based on a single OD pair that, collectively, may not necessarily reduce the total system cost. Thus, a coordination scheme is required to push the single OD level solutions towards the global optimality, i.e., the main objective in SODTA with multiple ODs. The MP captures the unseen coordination among the SPs (that exist in the coupling constraints prior to decomposition). Hence, the convex combination of the existing solutions to SPs will be found to ensure that the combined solution does not violate the coupling constraints. Those constraints establish a set of restrictions to the total flow of all OD pairs. The algorithm iterates between SP's and MP's optimizations until the termination criteria are satisfied.

In the next study, we present a Distributed Gradient-based Approach (DGA) to overcome the drawbacks of existing decomposition approaches about having a central component. The proposed methodology distributes the network-level DTA problem into several intersection-level DTA sub-problems. Then, the approach performs three steps to update the value of decision variables iteratively at each sub-problem: (1) It first incorporates the value of common decision variables among sub-problems by taking a weighted average, (2) the approach moves the values toward the negative direction of the gradient of the objective function at each sub-problem to minimize the objective, and (3) it projects the values onto the set of constraints at each sub-problem to maintain feasibility. The approach iterates among these three steps until the values of shared decision variables from sub-problems reach consensus. It allocates a computational node to each sub-problem and does not have a central component or a master problem. Therefore, it scales better than the algorithms with dependent complexity on network nodes and links. The computational complexity of sub-problems does not depend on the number of nodes and links in the network as a result of the spatial decomposition; however, it is a function of the number of OD pairs. This intersection-level distribution is well-suited for urban network planning because increasing the network geographic area by adding more intersections (nodes and links) will not change the computational complexity of the methodology and its structure. We show that the algorithm converges to the optimal solution of the problem in an infinite number of iterations. Note that the approach can work without necessarily starting with a feasible solution. Furthermore, the approach is applicable to problems with nonlinear and quadratic objectives without requiring any further simplifications or using complicated methods since it needs either the gradient or sub-gradients of the objective function.

We next introduce a distributed and coordinated solution technique to solve the CTM-based SODTA problem to find near-optimal solutions in real-time. This solution technique distributes the network-level problem into several intersection-level sub-problems. The distribution is achieved by relaxing the constraints that represent an interrelationship between the sub-problems. Dummy source and sink cells are added to the sub-problems to ensure they are stand-alone systems. This distribution significantly reduces the complexity of the problem as intersection-level sub-problems have far fewer decision variables, are linear, and can be solved in parallel. The solution technique creates distributed coordination between the sub-problems to push the solutions towards global optimality. The coordination is achieved by exchanging information between adjacent sub-problems and re-enforcing the relaxed constraints and incorporating the required information in them. The approach also temporally decomposes the study period to enable online network updates and generating real-time solutions.

In the last chapter, we propose a formulation for a bottleneck congestion management problem and solve the formulation with a heuristic algorithm. This problem includes two types of agents: system manager and traveler. The system manager provides path options considering network bottlenecks to minimize total travel time and maximize revenue, and travelers can submit their willingness to pay to use paths with network bottlenecks and minimize their travel costs. The first part of formulation is an allocation problem with an objective function to minimize total travel time and maximize revenue. The second problem is a UEDTA that finds user-optimal dynamic traffic flows and is formulated using a variational inequality approach. The solution technique solves this formulation with the genetic algorithm as an outer loop and the projection algorithm as an inner loop.

1.4. Research Contributions

Solving the problem centrally using standard algorithms like Simplex, Dual Simplex, and Barrier methods (e.g., Chiu and Zheng (2007); Nie (2011); and Aziz and Ukkusuri (2012)), or network flow algorithm (Zheng and Chiu, 2011) can determine optimal solutions for a small number of variables. However, testing networks with larger decision spaces will require long running times or/and large memories. Our proposed methodologies address limited scalability.

In the first study, we decompose the CTM-based SODTA into several single OD sub-problems using the Dantzig-Wolfe principle. These sub-problems are independent and include a significantly fewer number of decision variables compared to the original formulation, which facilitates the application of the proposed technique to larger problems in comparison to the existing CTM-based traffic assignment approaches (Beard and Ziliaskopoulos, 2006; Chiu and Zheng, 2007; Li et al., 2003; Zheng and Chiu, 2011). In contrast with decomposition approaches that cannot guarantee optimality (Ramadurai and Ukkusuri, 2011; Xie and Jiang, 2016), our first methodology is guaranteed to converge to the global optimality in a finite number of iterations as it satisfies all the assumptions of the Dantzig-Wolfe principle (Dantzig and Wolfe, 1960). Our assumptions in this study include the (a) feasibility of the initial solution, (b) linearity of the formulation, (c) convexity of the formulation, and (d) non-singularity of the constraint coefficient matrix of the master problem. These assumptions do not limit the application of this approach to specific network problems. The approach does not have non-trivial processes for parameter tuning despite studies by (Jafari et al., 2017; Larsson et al., 2004; Larsson and Patriksson, 1995). Restrictive assumptions on the arc capacity (Ramadurai and Ukkusuri, 2011), the ratio of free-flow speed to backward propagation wave speed (Li et al. 2003), or network geometry to be acyclic (Jafari et al., 2017; Larsson et al., 2004) do not appear as well.

In the next study, a distributed gradient-based approach is developed that has a fully distributed architecture and does not require a central component or master problem. In other words, the sub-problems work cooperatively without requiring a central optimization unlike existing decomposition approaches developed by Gibert (1968); Jafari et al. (2017); Larsson and Patriksson (1992); Larsson et al. (2004); Leventhal et al. (1973); Li et al. (2003); Lin et al. (2011); and Mehrabipour et al. (2019). Therefore, the proposed approach is scalable with the size of the network and do not face computational complexity growth in the master problem over iterations or by increasing the network size. The approach also converges to the optimal solution of the SODTA problem in an infinite number of iterations. The required assumptions for convergence do not restrict the application of the approach to specific network properties such as limiting the ratio of free-flow speed to backward propagation wave speed (Li et al., 2003) and having specific network geometry (Jafari et al., 2017; Larsson et al., 2004). Moreover, the computational complexity of sub-problems is independent of the number of nodes and links in the network since we create each sub-problem by the spatial distribution of the objective function and the set of constraints unlike the previous approaches: (e.g., Gibert (1968); Larsson and Patriksson (1992); Larsson et al. (2004); Leventhal et al. (1973); Li et al. (2003); Lin et al. (2011); and Mehrabipour et al. (2019)).

Decomposition approaches can find optimal or/and high-quality solutions for cell-based SODTA problems in less time by creating smaller decision spaces from the original formulation, but the computation time is not close to real-time. The current heuristics like MSA (Qian et al., 2012; Sbayti et al., 2007; Tajtehranifard et al., 2018), projection approach (Doan and Ukkusuri, 2015; Lu et al., 2016), and proximal point algorithm (Zhan and Ukkusuri, 2019) lack temporal and spatial decomposition schemes to reduce the complexity of CTM-based SODTA problems. These approaches force limitations (e.g., number of paths) especially when the number of variables increases. These approaches are also iterative techniques that cannot generate finding solutions in real-time. They mostly focus on finding path-marginal costs that is a computationally expensive step, requires generating paths, and storing them that lead to a limited number of paths for each OD. We bridge this gap by proposing a Distributed Optimization and Coordination Algorithm that decomposes the network-level SODTA problem (DOCA-SODTA) into several intersection-level sub-problems with a link-based structure. We also present a temporal decomposition to further decrease the number of variables at each sub-problem and generate solutions in real-time. Developing a coordination scheme allows the sub-problems to share information among themselves to ensure feasibility and finding high-quality solutions.

In our last study, we develop a mathematical formulation and solution technique for a bottleneck congestion management approach. The literature lacks a decision support system for this problem which is presented in this study. Despite other congestion pricing techniques, this approach does not need to determine toll prices. The determination of price requires access to accurate and private information. Note that the toll price needs to be determined and updated constantly to react to demand and supply changes. Personalized pricing is determined through market competition. Moreover, it is shown that market competition can be adaptive to economic and network changes very quickly.

1.5. Dissertation Layout

The exposition of this doctoral dissertation is as follows.

Chapter 2 presents a review of methodologies to solve DTA problems in the literature. This chapter provides a review of central, decomposition, and heuristic-based solution techniques. It also includes literature on congestion pricing techniques.

Chapter 3 explains the problem formulation that is mainly used in this dissertation. The DTA problem is formulated as a linear program by utilizing the CTM traffic dynamics and has a system-level optimal objective.

Chapter 4 presents a decomposition scheme based on the Dantzig-Wolfe principle to solve the CTM-based SODTA problem. The problem is decomposed into several OD-based sub-problems and a master problem. The numerical experiments of this scheme on networks of 20 and 40 intersections are also presented in this chapter.

Chapter 5 presents a gradient-based methodology that distributes the SODTA problem into several independent sub-problems without requiring a master problem. We also proved the optimality properties of this approach. Then, we showed analytical insights into the approach by implementing it on a network of 20 and 40 intersections, and the results are compared with benchmark approaches

Chapter 6 presents the development of a distributed optimization and coordination algorithm to find near-optimal solutions for a cell-based SODTA problem in real-time. The proposed algorithm includes two main components: distributed optimization and distributed coordination. The tests on networks with 20 and 40 intersections are presented. The results are compared with other heuristics and a benchmark approach capable of finding the optimal solutions.

Chapter 7 introduces a formulation to minimize a total travel time function and revenue by designing a bottleneck congestion management approach for using selected paths of a transportation network.

Finally, the concluding remarks and directions for future research are presented in Chapter 8.

CHAPTER 2. LITERATURE REVIEW

Dynamic Traffic Assignment (DTA) is well-studied since early researches by Merchant and Nemhauser (1978b, 1976) about forty decades ago. In DTA, time-dependent traffic flow from a specific origin to a specific destination is determined. Two objectives can be considered: (1) minimization of the total cost and (2) minimization of individual cost. These objectives result in two sets of problems namely, System Optimal (SO) DTA and User Optimal (UE) DTA (Peeta and Ziliaskopoulos, 2001). SODTA is one of the most extensively studied DTA problems because the objective function in SODTA is more attractive for researchers than its counterpart in UEDTA due to its flexibility to be used for various applications in transportation planning and management.

DTA has been studied in a wide range of applications, either as an independent problem or embedded in other problems. Applications include eco-friendly routing problems (Aziz and Ukkusuri, 2012; Lu et al., 2016), network policy evaluation (Karoonsoontawong and Waller, 2010), congestion management (Beard and Ziliaskopoulos, 2006; Doan et al., 2011; Muñoz and Laval, 2006), and evacuation planning (Zheng and Chiu, 2011). Another classification is presented by Nie (2011) as follows: evaluating investment decisions such as network expansion and design (Karoonsoontawong and Waller, 2010), traffic management policies such as congestion pricing and information provision (Shen and Zhang, 2009), operational strategies such as signal control and ramp metering (Beard and Ziliaskopoulos, 2006; Muñoz and Laval, 2006), and large-scale evacuation planning (Shen et al., 2007a). SODTA models can also be classified into two categories based on the type of time parameters: discrete-time models and continuous-time models. Examples of SODTA discrete-time models are SODTA formulations modeled with CTM (Ziliaskopoulos, 2000). In SODTA continuous-time models, the decision-making process is conducted continuously when network conditions change (Friesz et al., 1989). Another classification for

SODTA is introduced by Doan and Ukkusuri (2015) which considers different choice dimensions. This classification includes a single bottleneck model with only departure time choice (Doan et al., 2011), route choice problem (Zheng and Chiu, 2011), and incorporating route choice and departure time choice (Shen et al., 2007b). Moreover, SODTA can be formulated as a network of multiple origins and destinations (Qian et al., 2012), a network of single-origin and destination (Ziliaskopoulos, 2000), and with multiple destinations without diverging nodes (Shen et al., 2007b).

Different network loading concepts have been utilized to formulate DTA problems. The models can be categorized into exit flow function (Carey, 1992, 1987; Friesz et al., 1989; Wie et al., 1994), point queue model (Chow, 2009; Han et al., 2011; Ramadurai et al., 2010), and cell transmission model (CTM) (Chiu and Zheng, 2007; Zheng and Chiu, 2011; Ziliaskopoulos and Waller, 2000). Nie and Zhang (2005) have discussed the benefits and drawbacks of the mentioned network loading concepts.

Exit flow function-based approaches have a smaller set of decision variables as such, they can be applied to larger transportation networks. However, the flow propagation is instantaneous from the beginning of a link to its end. Moreover, the anisotropic property of traffic and queue spillbacks are not captured. The point queue model forms queues at nodes when the number of upcoming vehicles is more than the capacity of the link. This model has restrictive assumptions such as (1) the length of vehicles is zero, (2) their speed is equal to the free-flow speed, and (3) the queues are formed at the exit node.

The CTM (Daganzo, 1995, 1994b) loading concept represents each transportation link by several cells and decomposes the study period into short time intervals (e.g., seconds). Therefore, it can capture traffic flow dynamics more accurately than the other mentioned loading concepts

(Lo and Szeto, 2002; Szeto and Lo, 2004; Ukkusuri et al., 2012; Ukkusuri and Waller, 2008). As a result, it is more appropriate for operational purposes. However, the additional accuracy is achieved at the expense of a significant increase in computational complexity that limits the application of the CTM-based SODTA to networks with limited temporal and spatial scales and only a few OD pairs. In the rest of this chapter, different approaches to solve DTA problems are explained in detail. They are classified into optimization and heuristic-based approaches. Optimization-based approaches are further categorized into centralized and decomposition approaches.

2.1. Optimization-based Approaches to Solve DTA

2.1.1. Centralized Approaches

Exact analytical techniques to solve SODTA problems are mainly centralized. Merchant and Nemhauser (1978a) pioneered the research efforts with the development of a discrete-time, non-linear, non-convex macroscopic formulation for a SODTA problem with multiple origins and a single destination. In another study, they presented the optimality conditions of the formulation (Merchant and Nemhauser, 1978b).

In 1980, Ho presented an approach for solving the formulation developed by Merchant & Nemhauser (1978a). A sequence of objective functions was solved to find optimal solutions under various assumptions. Carey (1986) showed the validity of assumptions in the study of Merchant & Nemhauser (1978a), about using Kuhn-Tucker conditions for deriving the optimality conditions. The assumptions were specifically related to the formulation structure, e.g., using differentiable, continuous, and linearly-independent functions (Merchant and Nemhauser, 1978a). Later, Carey (1987) developed a convex, non-linear formulation with superior convexity properties compared to the formulation of Merchant & Nemhauser (1978a). Besides, he developed several formulations to account for multiple commodities and destinations that some led to non-convex models.

Wie et al. (1994) employed an augmented Lagrangian algorithm with the conjugate gradient method, which was an integrated penalty function and a primal-dual method. Based on Wardrop's second principle (Wardrop, 1952), the algorithm found the optimal solution when the dynamic marginal costs of all used paths were balanced. A discrete-time optimal control problem to minimize the total transportation cost was also modeled. The main drawback of the model was the inaccurate propagation of vehicles across arcs in uncongested conditions.

Exit flow functions used in all the previous formulations are only dependent on the link volumes. Therefore, they cannot capture traffic dynamics, and, since they do not propagate traffic gradually, they are not realistic and not suitable for the link analysis of traveled vehicles. Incorporating time-variant flows and their interactions through links lead to highly nonlinear, non-convex formulations (Gartner et al., 2002) that are computationally expensive (Carey and Subrahmanian, 2000). Carey & Subrahmanian (2000) presented linear convex models for a SODTA problem with multiple destinations that could capture traffic dynamics more accurately. Yet, the models were not fully dynamic since the delay of links was a function of only the link volumes, and the formulation could not capture queue spillback.

Ziliaskopoulos (2000) developed a CTM-based linear formulation to solve a SODTA problem with a single OD pair and provided the dual formulation and its interpretation. The formulation was solved centrally using Simplex for a very simplified network of 10 cells without any signalized intersection. In 2006, Beard and Ziliaskopoulos proposed a cell-based, mixed-integer linear formulation to integrate a SODTA problem with multiple OD pairs and a signal timing optimization problem. This formulation was tested centrally on a small network of 2 intersections. Even though both formulations could capture traffic dynamics, they were not scalable and became computationally intractable as the size of the problem grew. Therefore, efficient solution techniques were required.

Chiu & Zheng (2007) developed another linear formulation for SODTA with multiple OD pairs using the cell-based formulation presented by Ziliaskopoulos (2000) to determine prioritized optimal paths and departure schedules in a no-notice disaster condition. Because this formulation added another dimension to the feasible region of Ziliaskopoulos's (2000) formulation by incorporating multi-priority groups, the formulation had more computational complexity and was solved only for a simple network of 40 cells using the interior point method. This test network was unrealistic for the application of this study, and the proposed approach did not scale well with the size of the network.

Zheng & Chiu (2011) proposed a network flow algorithm to overcome the computational complexity of centralized approaches for solving Ziliaskopoulos's (2000) formulation. The researchers solved an equivalent problem, the earliest arrival flow, for the formulation and optimized path flows in a time-expanded network. The algorithm could perform faster in larger study periods compared to the primal-simplex and the interior point methods for a medium-sized network. However, all three algorithms were solved in polynomial time and became intractable with the addition of cells. Moreover, the network flow algorithm could not be applied to a SODTA problem with multiple OD pairs since the underlying assumptions of the algorithm could not be guaranteed.

Aziz & Ukkusuri (2012) developed a cell-based, non-linear, quadratic SODTA formulation with a single destination to minimize both total travel time and vehicle emissions. The formulation was solved centrally using commercial software for a similar network to Nguyen and Dupuis network with 14 nodes. Solving the formulation for a larger network required more computationally efficient algorithms. Using mesoscopic traffic flow models would increase the computational complexity even more (Lu et al., 2016).

Long et al. (2016) presented mixed-integer linear formulations for SODTA with a single destination to minimize the total emission of a network using a link transmission model. Several restrictive and unrealistic assumptions were considered to create the formulations. It was assumed that queue spillback would not occur, and all vehicles traversed a link with constant speed. The formulations were solved centrally for small networks involving 2, 6, 9, and 17 nodes. The link-based outputs could not provide enough information on different parts of a link for traffic management in the urban street network. Moreover, testing larger networks would not be possible without the development of more efficient methodologies.

Zhou et al. (2016) developed two approaches to solve a stochastic Static Traffic Assignment (STA) problem. The first approach consisted of two parts: linearize the problem and form a sub-problem; and find an approximate solution for the sub-problem. The second approach solved the Lagrangian relaxation of the dual problem with the steepest ascent method. These approaches were path-based, which could limit their application when the number of paths would increase.

Long, Wang, and Szeto, (2018) proposed twelve formulations to model SODTA using Link Transmission Model (LTM) to optimize route and departure choice decisions. Different combinations of models including First-In-First-Out (FIFO) and non-holding-back constraints were presented. The problems without FIFO constraints were solved using a commercial solver and those with FIFO constraints were optimized using the branch and bound algorithm for a network of at most twelve nodes. They showed that the link-based formulations were more computationally efficient compared to their path-based counterparts. However, solving the problem for larger cases would require efficient algorithms.

Chakraborty et al. (2018) presented a linear formulation for SODTA with a single OD pair using LTM. The formulation was applied to a six-node case study network. While the proposed formulation included 34% fewer decision variables compared to its corresponding cell-based formulation, it still did not scale well with the size of the problem.

All these approaches used a centralized solution approach, which did not scale well with the temporal and spatial scales of the CTM-based traffic assignment problems. Therefore, they have been applied to small problems that require a limited amount of available memory and time.

2.1.2. Decomposition Approaches

Decomposition techniques have shown superior performance and scalability for medium to large size network problems. To avoid path enumerations, observed in most centralized methods (except for link-based approaches), a subset of paths for each OD pair had to be included in the solution process. For instance, Gibert (1968) initiated the idea of iteratively generating paths under certain conditions. Similarly, Leventhal et al. (1973) implemented such a decomposition scheme for a non-linear, path-based STA problem using a Column Generation (CG) algorithm, which was first introduced by Ford and Fulkerson (1958) to decrease the computational complexity of a multi-commodity problem with a large number of variables. Later, Dantzig and Wolfe (1960) generalized the CG algorithm to broaden its application to any linear program.

Wollmer (1969) generalized the decomposition algorithm presented by Tomlin (1966) to solve multi-commodity network flow models with no flow-commodity restrictions. Using the Dantzig-Wolfe principle, each Sub-Problem (SP) found arc flows for one commodity, and Master Problem (MP) optimized the convex combination of available link flows to satisfy capacity constraints; however, this algorithm was not tested and evaluated numerically. Besides, Tomlin (1971) used the Dantzig-Wolfe principle to solve an integrated equilibrium STA and distribution problem. The algorithm generated new distributions at each iteration through SPs and added them to the MP, which found a new assignment. Link-path and node-link decomposition schemes were presented, where all paths were enumerated in the link-path design leading to a computationally expensive process in large-scale networks. Besides, the convergence of the algorithm was not guaranteed since the formulations did not satisfy the linearity assumption of the Dantzig-Wolfe principle. CG was further implemented to solve variational inequality formulations in Bertsekas & Gafni (1982) and complementarity formulations in Aashtiani et al. (1983).

Larsson & Patriksson (1992) proposed a decomposition algorithm using a block-diagonal structure of an arc-node formulation for the STA problem. SPs generated paths for each OD pair, and MP found the best flow assignment to the generated paths for each OD pair. The MP was consecutively solved using a scaled reduced gradient followed by a Newton method. Using these methods might lead to finding sub-optimal solutions. The algorithm was designed for a formulation using an exit flow function that was unable to propagate traffic accurately, especially with time-variant demand. Moreover, the algorithm required the storage of both paths and flows. Therefore, providing significant storage space was always a problem as experienced by Abrahamsson, (1998).

Later, Larsson and Patriksson (1995) decomposed an STA problem with flow capacity upper bounds using an augmented Lagrangian dual method. Initialization of Lagrangian multipliers and a penalty parameter in the algorithm were experimental and non-trivial. The uncapacitated sub-problems were solved using the disaggregate simplicial decomposition algorithm (Larsson and Patriksson, 1992). The solutions were not guaranteed to be optimal due to using the scaled reduced gradient method in the disaggregate simplicial decomposition algorithm. Moreover, the algorithm required the storage of paths and their corresponding storage. The solutions of the sub-problem were not feasible, and an additional heuristic algorithm was incorporated to generate feasible solutions. The algorithm was tested on small to large size networks, but the optimality properties of solutions were not discussed.

Larsson et al. (2004) proposed a CG-based algorithm to solve an STA problem with linear side constraints. The side constraints were relaxed to create a Lagrangian dual problem. Then, MP was created by dualizing the Lagrangian dual problem. MP found the best convex combination of the available extreme points for link flows. SP was a traffic assignment problem that was solved by the disaggregate simplicial decomposition algorithm (Larsson and Patriksson, 1992). Solving SPs by finding the shortest paths required the assumption of non-negative cycle times even though no approach had been presented in the presence of negative cycle times. Furthermore, MP solutions might not be feasible at each iteration because MP was modified using a Box Step method to stabilize the CG-based algorithm. Setting parameters in this algorithm to avoid negative cycles and find better quality solutions was a non-trivial process. The algorithm was tested on small to medium size networks. Moreover, the optimality of solutions was not guaranteed since SPs were solved with a gradient projection method.

In a rather similar context, the implementation of the Benders decomposition algorithm to solve traffic assignment problems was explored by Hearn (1984) and Barton et al. (1989). Their algorithms could solve simplified traffic models without capturing traffic dynamics. Besides, their decomposition approach required the enumeration of all paths in each iteration to assign traffic.

Ramadurai & Ukkusuri (2011) presented a simulation-based decomposition algorithm for a user equilibrium DTA problem formulation with a point queue model. The network was decomposed into acyclic sub-networks. Then, the algorithm found paths with minimum and maximum utilities in each sub-network according to link costs and shifted the flow between paths in certain conditions. The algorithm required a restrictive assumption on some of the arcs' capacities. The algorithm was tested on the Sioux Falls network with 30 nodes, but the quality of solutions was unspecified, and the algorithm failed to converge smoothly in some cases.

Li et al. (2003) proposed a decomposition scheme using the Dantzig-Wolfe principle based on the formulation introduced by Ziliaskopoulos (2000) with a single destination. MP found the best convex combination of available flow patterns, and SP was a minimum-cost-flow problem on a space-time network. The algorithm was tested on a 20-node small network of 62 cells with 3 origins. The algorithm could not be applied to cases with fractional ratios of free flow speed to backward propagation speed because the properties of a minimum cost flow problem would not be satisfied for the sub-problems. Note that this ratio is within the 0.2-0.5 range in realistic cases (Lin and Ahanotu, 1995). This algorithm required a large number of iterations for convergence. The first reason was that when multiple OD pairs were included, two levels of decomposition had to be considered to solve SPs. Therefore, the convergence criterion had to be met twice in each iteration. The second reason was that the capacity constraint could not be modeled as a network flow constraint and must be excluded from SP. Adding this constraint to SP could improve the convergence substantially

Moreover, Lin et al. (2011) proposed a Dantzig-Wolfe-based decomposition heuristic to calibrate the flow capacity of CTM-based DTA with a user-optimal objective function. They decomposed the dual formulation of bi-level capacity calibration into an MP and several pricing SPs. However, the solution quality was not assured, due mainly to the approximation of the dual variables, and the algorithm required substantial modifications for application to larger network problems with multiple OD pairs.

Xie & Jiang (2016) proposed an algorithm based on the Benders decomposition technique to solve a non-linear integer program for STA with additional constraints to consider charging stations for electric vehicles. The master problem generated new paths and was solved by a gradient projection method and the primal problem found the equilibrium solution and was solved using a labeling algorithm. The model used in the paper could not capture traffic dynamics, and the solution properties were not known due to the primal approximations.

Jafari et al. (2017) decomposed the UESTA problem spatially by creating sub-networks to represent sub-problems. The sub-problems found the traffic assignment solution within a sub-network given the regional demand from a master problem. The master problem solved the STA problem for an aggregated network using modified travel times obtained from the sub-problems. They needed to create the aggregated network with arbitrary arcs that contained all paths based on the original network. They applied a method of sensitivity analysis by Jafari and Boyles (2016) to find required parameters for arbitrary links and relative changes in path flows by demand fluctuations at each iteration. The approach by Boyles (2012) provided the basis for network aggregation to create the master problem and modeling sub-networks. Boyles's (2012) method could find the cost function of arbitrary arcs in the aggregated network as well. In Jafari et al.'s (2017) approach, considering more sub-networks led to increasing the complexity of the aggregated network and the master problem. Moreover, the algorithm was restricted to acyclic networks and required an initial feasible solution to satisfy UE conditions.

2.2. Heuristic Approaches to Solve DTA

In practice, the optimal solution is not easily accessible for real traffic networks either in a reasonable amount of time or in real-time. Nondifferentiable problems are also the main reasons to shift from exact methods to heuristic-based algorithms. Because of this transition, heuristics approaches are widely used to find the solutions to intractable SODTA problems. However, the optimality of the solution is not guaranteed and the algorithms require long run times.

The Method of Successive Averages (MSA) has been employed extensively as a heuristic solution technique to solve SODTA problems. This technique generates new paths iteratively and assigns traffic to each path according to experienced travel times. The main challenge of MSA is the procedure of finding a step size in path flow allocation. The step size should utilize available information on the recent cost of paths to swap flow among newly generated paths. The main application of MSA is for solving path-based traffic assignment problems. Thus, finding and storing paths is another challenge that makes MSA computationally expensive. Furthermore, MSA faces either slow, insufficient, or failure in convergence in large and congested networks, more discussion can be found in a study by Sbayti et al., (2007). Sbayti et al. (2007) improved MSA by introducing a variable step size. Their framework could be applied to different network loading models. They distributed flows among the updated set of paths considering the willingness of drivers to switch their current paths to newly added paths to the set. They also presented a vehicle-based approach with reduced storage space requirements. However, the approach required lots of iterations and time to generate solutions.

One critical step in some of the SODTA heuristic-based approaches is finding the marginal costs of the paths and links in transportation networks. The SODTA condition according to Wardrop’s second principle (Wardrop, 1952) states that if a path contains any flow, the path marginal cost of the path is equal to the minimum path marginal cost. This condition can be interpreted as the equilibrium of path marginal costs for all used paths. These marginal costs show the effect of changes in flows on the total system costs. The correctness and efficiency of algorithms to estimate the marginal costs as a key factor of convergence to the optimal solution have created methodological challenges (Doan and Ukkusuri, 2015; Ghali and Smith, 1995; Lu et al., 2016; Nie, 2011; Peeta and Mahmassani, 1995; Peeta and Zhou, 2006; Qian et al., 2012; Shen et al., 2007b, 2006).

Qian et al. (2012) embedded their proposed path marginal cost computation and a least marginal cost algorithm in MSA to solve a path-based SODTA formulation. The presented approach for the path marginal cost computation would work for point queue and kinematic wave models. Doan & Ukkusuri (2015) also presented an approach to find path marginal costs in a cell-based SODTA formulation. Their projection algorithm had better solutions and computational time in comparison with MSA by incorporating their approach for finding path marginal costs. The projection algorithm was mostly used for variational inequality problems. In this algorithm, a quadratic program was solved in each iteration to find new departure rates according to path marginal costs (Ukkusuri et al., 2012). Both heuristics developed by Qian et al. (2012) and Doan & Ukkusuri (2015) were limited to cases with a predetermined number of paths for each OD. Moreover, these approaches would become intractable with an increase in the spatial and temporal scales of problems due to the computational cost of a path-based simulation and marginal cost estimation. The approaches were also iterative and could not generate solutions in real-time.

Lu et al. (2016) presented a path-based heuristic approach using marginal costs to solve a SODTA problem and determine the greenest routes for drivers. The heuristic algorithm had a general framework that could be used for various network loading models. The algorithm updated the set of paths for each OD in each iteration using a time-dependent shortest path algorithm. They also solved the SODTA problem with a gradient projection algorithm considering an updated set of paths and an approach to find link marginal emissions. This approach needed several iterations for convergence, which did not allow finding solutions in real-time.

Tajtehranifard et al., (2018) proposed a path marginal cost approach to solve an SO problem with quasi-DTA network loading. The model used capacity constraints and residual queues in a static traffic assignment problem. However, this model could not capture queue spillbacks and consider signalized intersections. The developed path marginal cost approximation was embedded in MSA. The algorithm considered a limited number of paths for OD pairs to reduce the computational efforts in computing path marginal costs and travel times. It also had two loops and required convergence for inner and outer loops to be terminated, which might not allow finding solutions in real-time. Moreover, finding a good perturbation of flow on each path for path marginal cost approximation and appropriate step size values in MSA were non-trivial steps.

Zhan and Ukkusuri (2019) proposed a cell-based user equilibrium DTA formulation and solved it using a proximal point algorithm developed by Rockafellar (1976). The proximal point algorithm used in this study was a heuristic technique that solved variational inequality problems using the projection approach iteratively. Since the algorithm optimized a formulation with a computational complexity that was dependent on the network size in each iteration, it faced long CPU times. The algorithm also showed slow convergence and deviated from optimality mainly due to the included perturbation in variational inequality problems. As suggested by the authors, distributed techniques could help enhance the efficiency of their approach.

The discussed approaches were scalable and more efficient in run-time in comparison to the optimization-based approaches; however, they could not find near-optimal solutions in real-time. Moreover, these approaches were iterative, and they needed to find the marginal costs of paths and links in a network to satisfy the equilibrium condition of SODTA that might lead to long run-time.

2.3. Congestion Pricing

Congestion pricing and tolling systems have been studied for more than 10 decades, and preliminary ideas belong to Pigou (1912) and Vickrey (1969). Tolling has been recognized as an effective approach for managing and alleviating traffic congestion for many years. It also results in a revenue source for maintaining and constructing infrastructure. The focus of our literature is on second-best pricing which enforces tolls on a subset of links in a network.

Most studies require toll determination for finding second-best pricing solutions. The required calibration for finding cost functions and the assumption of having perfect information on current and future travel costs make these studies restrictive and sensitive to information accuracy. Verhoef (2002) proposed a mathematical formulation to determine the second-best optimal toll levels and locations to maximize social welfare. The problem was modeled using a Lagrangian function with equilibrium conditions. Shepherd and Sumalee (2004) proposed genetic-based algorithms to apply Verhoef's formulation to realistic case studies. Lawphongpanich and Hearn (2004) presented nonlinear programming formulations with equilibrium constraints and proposed a cutting constraint algorithm to find second-best prices for realistic cases. Joksimovic et al., (2005) studied the second-best pricing problem with route and departure time decisions. They presented a bi-level formulation to find optimal uniform and time-dependent toll levels. A grid-search solution technique was used to solve the formulations which required exploring all solutions. Di et al., (2016) studied a boundedly rational route choice behavior instead of a classical UE problem in second-best pricing and bi-level tolling problems.

Lin et al., (2011a) determined time-dependent tolls by developing and solving a bi-level linear program with a CTM network flow. The upper-level formulation minimized the total travel time of the system and was the difference between the arrival and departure time for flow over time. The lower level was a UEDTA formulated as a variational inequality. They used a dual variable approximation technique and the method of successive averages to solve the formulation. The combinatorial heuristic found solutions with about a 2% optimality gap. Approximating toll values was dependent on formulation structure and could not be generalized. This study could not also solve problems with multiple destinations. Sharon et al., (2017) proposed a tolling framework with three components: traffic, travel time estimation, and tolling models. The traffic model determined drivers’ routes and was a function of a system state, travel times of links, demand, and tolls. This framework was implemented using 3 network loading models including link performance functions, CTM, and microsimulation. The travel time model used a system state to estimate link travel times. The toll estimation model required tuning for two parameters and was a function of link travel time, free-flow travel time, and tolls on the previous time step. Even though this study used less assumptions and parameters compared to initial studies to determine toll, it still faced fundamental challenges. These tolling systems face public opposition and the lack of accurate or complete information may lead to their failure.

Congestion pricing through tokens is another classification of congestion pricing techniques. Kockelman and Kalmanje (2005) argued that conventional congestion pricing strategies suffer from equity issues because the strategies favor users with a higher value of time who are willing to pay higher prices for the tolls. These users are mostly associated with higher income levels. Therefore, Kockelman and Kalmanje (2005) suggested that credit-based congestion pricing has the potential to resolve equity issues. In credit-based congestion pricing, drivers receive a monthly travel credit that can be used on tolled roads. Users that do not use their credit can use it later or receive the equivalent cash value of their unused credit, whereas users that go over their credit should pay for the extra usage. To evaluate the public acceptance of this tolling strategy, Kockelman and Kalmanje (2005) conducted a survey in Austin, TX, and concluded that 25% of the respondents supported this strategy. Yang and Wang (2011) stated that distributing credits among users requires issuing credits to all eligible users such as all taxpayers, determining the charge or credit consumption for each tolled link, and trading the credits in a free market. Therefore, Yang and Wang (2011) showed that for a general network with homogenous users, there existed unique equilibrium flow patterns that could be found using standard traffic equilibrium models with the side constraint of total credit consumption. Although the strategy discussed by Kockelman and Kalmanje (2005) and Yang and Wang (2011) could resolve the equity issues, it did not necessarily improve traffic operations at bottlenecks since it was a planning-level strategy. Therefore, the response of users to this strategy and how they distributed their credit usage over a month (or any other credit allocation period) might negatively impact traffic congestion.

Su and Park (2015) developed an agent-based simulation for highway reservations. The highway reservation mechanism was proposed as an alternative to traditional fixed-rate tolling systems. In the proposed mechanism, system users bid on their desired routes with an on-ramp time interval. A reservation management center sorted all bids based on their bidding amount. Starting from higher bids, the bids were accepted or rejected based on the capacity of the link. Spatial-temporal tables were used for storing and updating highway reservations. It was assumed that the users arrived within the reserved on-ramp time interval. The case study of this paper was a 13-mile freeway with two routes which included a reserved highway and an arterial. MATSim was used for simulation, and the results of comparing the reservation system with DTA showed the travel time was more in DTA in most cases.

Basar and Cetin (2017) presented online survey results for an auction-based tolling system using the technology of autonomous and connected vehicles for efficient utilization of low capacities of roads. This system was a descending price auction including two arbitrary routes of tolled and non-tolled. The survey results were analyzed with discrete choice models to determine the effects of different factors on tolls. This tolling system showed no public rejection or unacceptance. This system was also simulated to compare fixed tolling and auction-based tolling systems. The results indicated that deployment of this system could lead to reductions in travel time by 30 minutes and an increase in revenue by a least %70 compared to fixed tolls. The two above studies presented simulation-based studies on auction-based tolling systems. However, they did not provide a procedure to generate optimal or near-optimal solutions for this system.

CHAPTER 3. SYSTEM OPTIMAL DYNAMIC TRAFFIC ASSIGNMENT FORMULATION

The SODTA problem is formulated as a linear program utilizing the CTM traffic dynamics introduced by Daganzo (1994 and 1995). Note that CTM relates flow and density in each cell using non-linear equations. We used the linearized form of the CTM-based SODTA formulation introduced by Beard and Ziliaskopoulos (2006) and modified the set of OD pairs to reduce the formulation complexity. Two indices are considered for origins and destinations by Beard and Ziliaskopoulos (2006). Instead, we used one index by creating a set consisting of tuples for OD pairs. this modification reduces the dimensionality, and, as a result, the computational complexity of the formulation.

Table 3-1 presents the sets, decision variables, and parameters used for formulating SODTA. Let C , T, C_{OD}, and $S(i)$ respectively denote the set of cells, time steps, OD pairs, and successors to cell $i \in C$. This formulation includes two sets of decision variables. The first set is the number of vehicles $x_i^{t,od}$in cell $i \in C$ at time step $t \in T$ with OD pair $(o,d) \in C_{OD}$, and the second set is the number of vehicles $y_{ij}^{t,od}$ flowing from cell $i \in C$ to successor cell $j \in S(i)$ at time step $t \in T$ with OD pair $(o,d) \in C_{OD}$. This formulation receives traffic signals as inputs, and equation (3-1) finds the variable saturation flow rate f_i^t at intersection cell $i \in C_I$ for time step $t \in T$ using pre-defined signal timing parameters. We use C_I to denote the set of intersection cells. The signal status g_i^t is a binary parameter defined for all intersection cells $i \in C_I$ and time steps $t \in T$. When the signal is green, g_i^t will be one and zero otherwise. The variable saturation flow rate is equal to the constant saturation flow rate if the signal in an intersection cell is one. More studies on signal timing optimization can be found in (Hajbabaie et al., 2021, 2018, 2011; Hajbabaie and Benekohal, 2013, 2011a; Islam et al., 2021; Medina et al., 2013; Mehrabipour, M., 2017;

Mehrabipour and Hajbabaie, 2017; Tajalli et al., 2019; Tajalli and Hajbabaie, 2021a). More studies on CTM can be found in studies by Al Islam and Hajbabaie (2021) and Islam et al., (2020).

$$f_i^t = g_i^t F_i \qquad \forall t \in T, i \in C_I \tag{3-1}$$

Table 3-1 Definition of sets, decision variables, and parameters used in chapter 3

Sets:	
T	The set of all time steps
C	The set of all network cells
C_O	The set of all source cells
C_S	The set of all sink cells
C_I	The set of all intersection cells
C_{OD}	The set of all OD pairs
$P(i)$	The set of all predecessors to cell $i \in C$
$S(i)$	The set of all successors to cell $i \in C$
Decision variables:	
$x_i^{t,od}$	The number of vehicles in cell $i \in C$ at time step $t \in T$ with OD pair $(o,d) \in C_{OD}$
$y_{ij}^{t,od}$	The number of vehicles flowing from cell $i \in C$ to downstream cell $j \in S(i)$ at time step $t \in T$ with OD pair $(o,d) \in C_{OD}$
Parameters:	
τ	The duration of each time step
$d_i^{t,od}$	The entry demand level at source cell $i \in C_O$ at time step $t \in T$ from origin o to destination d in OD pair $(o,d) \in C_{OD}$
F_i	The saturation flow rate at cell $i \in C$
M_i	The maximum number of vehicles that cell $i \in C$ can accommodate
g_i^t	A binary parameter to define signal status at intersection cell $i \in C_I$ at time step $t \in T$. Zero and one values indicate red and green signals, respectively.
f_i^t	The variable saturation flow rate of intersection cell $i \in C_I$ at time step $t \in T$

We define the minimization of total travel time as the objective function in equation (3-2). The total travel time is found by summing all vehicles $x_i^{t,od}$ in all network cells except for sink cells $i \in C \backslash C_S$ with all OD pairs $(o,d) \in C_{OD}$ over all time steps $t \in T$ and multiplying the result by τ

(i.e., the duration of each time step). We can eliminate the time step duration τ from the objective function for simplicity since it is a constant value.

$$\text{Min} \sum_{(o,d)\in C_{OD}} \sum_{t\in T} \sum_{i\in C\setminus C_S} \tau x_i^{t,od} \tag{3-2}$$

Constraints (3-3), (3-4), and (3-5) show the conservation of flow for different cell types. The increase or decrease in the number of vehicles $x_i^{t+1,od} - x_i^{t,od}$ between time steps $t \in T$ and $t + 1 \in T$ is equal to the difference of the total inflow to and total outflow of cell $i \in C$ at time step $t \in T$ for OD pair $(o, d) \in C_{OD}$. Constraints (3-3) ensures the flow conservation for all cells $i \in C$ except for source cells $i \in C_O$ and sink cells $i \in C_S$. Constraint (3-4) shows the flow conservation for source cells $i \in C_O$, and the incoming flow for these cells is the demand. We use $d_i^{t,od}$ to denote the demand at source cell $i \in C_O$ at time step $t \in T$ for OD pair $(o, d) \in C_{OD}$. Constraint (3-5) is for sink cells C_S, where there is no outflow.

$$\sum_{k\in P(i)} y_{ki}^{t,od} - \sum_{j\in S(i)} y_{ij}^{t,od} = x_i^{t+1,od} - x_i^{t,od} \qquad \forall t \in T, i \in C \setminus \{C_S, C_O\}, (o,d) \in C_{OD} \tag{3-3}$$

$$d_i^{t,od} - \sum_{j\in S(i)} y_{ij}^{t,od} = x_i^{t+1,od} - x_i^{t,od} \qquad \forall t \in T, i \in C_O, (o,d) \in C_{OD} \tag{3-4}$$

$$\sum_{i\in P(j)} y_{ij}^{t,od} = x_j^{t+1,od} - x_j^{t,od} \qquad \forall t \in T, j \in C_S, (o,d) \in C_{OD} \tag{3-5}$$

Constraint (3-6) limits the total outgoing flow $\sum_{j\in S(i)} y_{ij}^{t,od}$ from cell $i \in C$ to its successor cells $j \in S(i)$ to the occupancy $x_i^{t,od}$ of the cell at time step $t \in T$ with OD pair $(o, d) \in C_{OD}$.

$$\sum_{j\in S(i)} y_{ij}^{t,od} \le x_i^{t,od} \qquad \forall t \in T, i \in C, (o,d) \in C_{OD} \tag{3-6}$$

Constraints (3-7) and (3-8), respectively, ensure that the total outgoing flow from and the total incoming flow to a cell are limited to the constant saturation flow rate of the cell. We find the total outflow $\sum_{(o,d)\in C_{OD}}\sum_{j\in S(i)} y_{ij}^{t,od}$ or total inflow $\sum_{(o,d)\in C_{OD}}\sum_{i\in P(j)} y_{ij}^{t,od}$ at time step $t\in T$ by summing all outgoing flows from cell $i\in C$ to its successor cells $j\in S(i)$ and incoming flows to cell $j\in C$ from its predecessor cells $i\in P(j)$, respectively, over all OD pairs. We use F_i to denote the constant saturation flow rate of cell $i\in C$.

$$\sum_{(o,d)\in C_{OD}}\sum_{j\in S(i)} y_{ij}^{t,od} \le F_i \qquad \forall t\in T, i\in C \qquad (3\text{-}7)$$

$$\sum_{(o,d)\in C_{OD}}\sum_{i\in P(j)} y_{ij}^{t,od} \le F_j \qquad \forall t\in\ T, j\in C \qquad (3\text{-}8)$$

The total incoming flow $\sum_{(o,d)\in C_{OD}}\sum_{i\in P(j)} y_{ij}^{t,od}$ to cell $j\in C$ at time step $t\in T$ should be less than or equal to the available capacity $M_j-\sum_{(o,d)\in C_{OD}} x_j^{t,od}$ of that cell as shown by constraint (3-9). The total inflow $\sum_{(o,d)\in C_{OD}}\sum_{i\in P(j)} y_{ij}^{t,od}$ at time step $t\in T$ is found by summing flow $y_{ij}^{t,od}$ on all incoming links to cell $j\in C$ from its predecessor cells $i\in P(j)$ and over all OD pairs $(o,d)\in C_{OD}$. The total occupancy $\sum_{(o,d)\in C_{OD}} x_j^{t,od}$ of cell $j\in C$ at time step $t\in T$ is computed by summing the occupancy $x_j^{t,od}$ over all OD pairs $(o,d)\in C_{OD}$. Let M_j denote the maximum number of vehicles that cell $j\in C$ can accommodate. Hence, we find the available capacity by $M_j-\sum_{(o,d)\in C_{OD}} x_j^{t,od}$ for cell $j\in C$ at time step $t\in T$. We use δ to denote the ratio of free-flow speed to the backward propagation speed.

$$\sum_{(o,d)\in C_{OD}}\sum_{i\in P(j)} y_{ij}^{t,od} \le \delta(M_j-\sum_{(o,d)\in C_{OD}} x_j^{t,od}) \qquad \forall t\in T, j\in C, (o,d)\in C_{OD} \qquad (3\text{-}9)$$

Constraint (3-10) limits the total outgoing flow $\sum_{(o,d)\in C_{OD}}\sum_{j\in S(i)} y_{ij}^{t,od}$ from intersection cell $i\in C_I$ to variable saturation flow rate f_i^t at time step $t\in T$. We sum the flow $y_{ij}^{t,od}$ on all links between cell $i\in C_I$ and its successor cells $j\in S(i)$ over all OD pairs to find the total outflow $\sum_{(o,d)\in C_{OD}}\sum_{j\in S(i)} y_{ij}^{t,od}$ at time step time step $t\in T$. Equation (3-1) finds the variable saturation flow rate f_i^t given the signal status g_i^t for intersection cell $i\in C_I$ and time step $t\in T$.

$$\sum_{(o,d)\in C_{OD}} \sum_{j\in S(i)} y_{ij}^{t,od} \leq f_i^t \qquad \forall t \in T, i \in C_I \tag{3-10}$$

Constraints (3-11) and (3-12) are used to ensure that the decision variables are nonnegative.

$$x_i^{t,od} \geq 0 \qquad \forall t \in T, i \in C, (o,d) \in C_{OD} \tag{3-11}$$

$$y_{ij}^{t,od} \geq 0 \qquad \forall t \in T, i \in C \backslash C_S, j \in S(i), (o,d) \in C_{OD} \tag{3-12}$$

CHAPTER 4. SOLVING SODTA WITH THE DANTZIG-WOLFE DECOMPOSITION PRINCIPLE

This chapter presents a decomposition methodology for solving the SODTA problem that includes generating initial feasible solutions, formulating a restricted Master Problem (RMP), and formulating Sub Problems (SPs). The algorithm initiates with a set of feasible solutions to the problem (3-2)-(3-12). At each iteration, the RMP finds an optimal convex combination of currently available extreme points. Then, the objective functions of the SPs are updated based on the dual values of the RMP, and new extreme points are generated. The recently generated points are added to the RMP's set of input parameters. The iterative procedure continues until the termination criterion is satisfied. The steps of the algorithm are detailed in the following sections. Table 4-1 presents the definition of sets, decision variables, and parameters used in this chapter. The schematic diagram of the described decomposition methodology is shown in Figure 4-1. Note that this study is published by Mehrabipour et al. in 2019.

Table 4-1 Definition of sets, decision variables, and parameters used in chapter 4

Sets:	
T	The set of all time steps
C	The set of all network cells
C_O	The set of all source cells
C_S	The set of all sink cells (i.e., vehicle destinations)
C_I	The set of all intersection cells
C_{OD}	The set of all origin-destination pairs
$P(i)$	The set of all cell predecessors
$S(i)$	The sets of all cell successors
E_O	The set of all ordinary links
E_D	The set of all diverge links

Table 4-1 (continued)

C_D	The set of all diverge cells
P	The set of all paths
E	The set of extreme points
Decision variables:	
$x_i^{t,od}$	The number of vehicles in cell $i \in C$ at time $t \in T$ with $od \in C_{OD}$
$y_{ij}^{t,od}$	The flow of vehicles from cell $i \in C$ to successor cell $j \in S(i)$ at time $t \in T$ with $od \in C_{OD}$
$x_i^{t,p}$	The number of vehicles in cell $i \in C$ at time $t \in T$ for path $p \in P$
$y_{ij}^{t,p}$	The flow of vehicles from cell $i \in C$ to successor cell $j \in S(i)$ at time $t \in T$ for path $p \in P$
$\xi_i^t = \sum_{\forall p \in P} x_i^{t,p}$	The total number of vehicles in cell $i \in C$ at time $t \in T$ for all paths $p \in P$
$\psi_{ij}^t = \sum_{\forall p \in P} y_{ij}^{t,p}$	The flow of vehicles from cell $i \in C$ to successor cell $j \in S(i)$ at time $t \in T$ over all paths $p \in P$
$\eta_{ij}^t = \sum_{p \in P} x_i^{t,p}, \forall (i,j) \in p$	The total number of vehicles in cell $i \in C_D$ at time $t \in T$ that plan to go to cell $j \in S(i)$ over all paths $p \in P$ containing link (i,j)
$\gamma_{ij}^{t,p}$	The number of vehicles in cell $i \in C_D$ at time $t \in T$ that plan to go to cell $j \in S(i)$: $x_i^{t,p}$ if path $p \in P$ contains link (i,j) or 0 otherwise
$\pi_{i,j}^t$	The dual variable for cell $i \in C$ for coupling constraints j at time $t \in T$
φ_{od}	The dual variable for convexity constraints on $od \in C_{OD}$
Parameters:	
τ	Duration of each time step
$d_i^{t,od}$	The entry demand level at source cell $i \in C_O$ at time $t \in T$ for $od \in C_{OD}$
F_i	The saturation flow rate at cell $i \in C$
$f_i^t = g_i^t F_i, \forall t \in T, i \in C_I$	The variable saturation flow rate at cell $i \in C$ at time $t \in T$
M_j	The maximum number of vehicles that cell $j \in C$ can accommodate
δ	The ratio of free flow speed to backward propagation speed
g_i^t	A binary parameter to define signal status: 1 if signal is green or 0 if it is red
f_i^t	The variable saturation flow rate in intersection cell $i \in C_I$ at time $t \in T$
μ	An arbitrary small and positive number
$\mathcal{D}^{t,p}$	The entry demand level on path $p \in$ P at time $t \in T$

4.1. Step 0. Initialization

The occupancy and flow on the shortest paths can help to provide a set of feasible solutions and initialize the algorithm. We have first applied the Dijkstra algorithm by Ahuja et al. (1993) to find the shortest paths, where the cost of each link is set to travel time under free-flow conditions. Then, we have implemented a path-based CTM simulation to produce the flow on paths for the entire network. Details follow.

4.1.1. Path-based Simulation

We have implemented the CTM path-based simulation developed by Ukkusuri et al. (2012) with slight changes to obtain the values of $x_i^{t,p}$ and $y_{ij}^{t,p}$. We define $x_i^{t,p}$ as the total number of vehicles in each cell $i \in C$ at time $t \in T$ on path $p \in P$; and $y_{ij}^{t,p}$ as the flow of vehicles from cell $i \in C$ to cell $j \in S(i)$ over path $p \in P$ at each time $t \in T$.

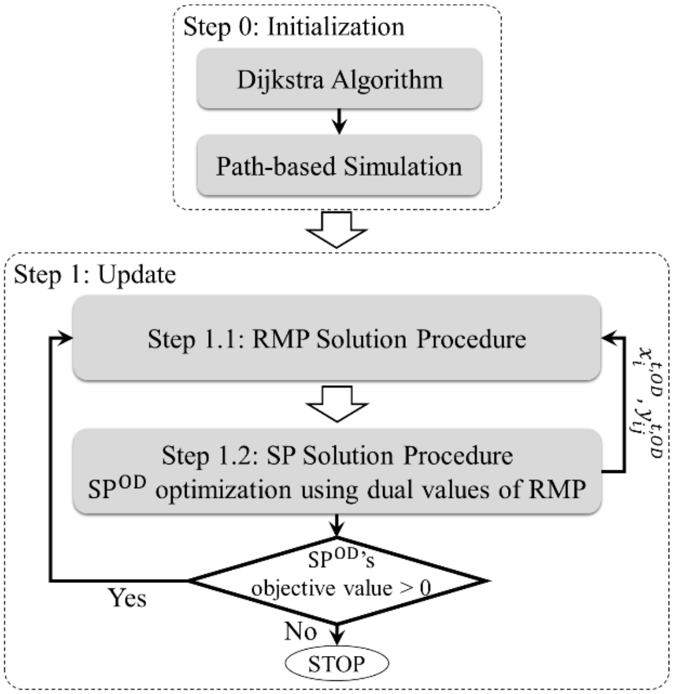

Step 0: Initialization
Dijkstra Algorithm
Path-based Simulation
Step 1: Update
Step 1.1: RMP Solution Procedure
Step 1.2: SP Solution Procedure
$\mathrm{SP}^{\mathrm{OD}}$ optimization using dual values of RMP
$x_i^{t,OD}, y_{ij}^{t,OD}$
$\mathrm{SP}^{\mathrm{OD}}$'s objective value > 0
Yes
No
STOP

Figure 4-1 The decomposition scheme for SODTA with multiple OD pairs

The summation of all vehicles in cell $i \in C$ at each time $t \in T$ over all paths is denoted by ξ_i^t, calculated by $\sum_{\forall p \in P} x_i^{t,p}$. We also define η_{ij}^t as the total number of vehicles in cell $i \in C_D$ at time $t \in T$ that plan to go to cell $j \in S(i)$ over all paths $p \in P$ containing link (i,j). In other words, η_{ij}^t represents the total number of vehicles that simultaneously need to use a link leaving a diverge cell. We have denoted the summation of $y_{ij}^{t,p}$ over all paths by ψ_{ij}^t, calculated by $\sum_{\forall p \in P} y_{ij}^{t,p}$. In addition, parameter $\mathcal{D}^{t,p}$ represents the demand of each path $p \in$ P at time $t \in T$.

The links are categorized into sets of ordinary links E_O and diverge links E_D. A representation of an ordinary link $(i,j) \in E_O$, a diverge link $(i,j) \in E_D$, and a diverge cell $i \in C_D$ is shown in Figure 4-2.

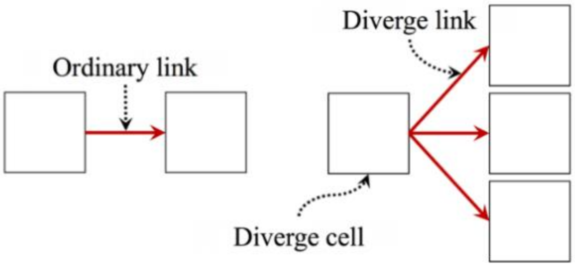


Figure 4-2 Link and cell representation in path-based simulation

To start the CTM simulation, we first assign initial values to $x_i^{t,p}$ at time $t = 0$ for each $i \in C$ on path $p \in P$. We assume the network is empty at the beginning of the study period, i.e., $x_i^{0,p} = 0, \forall i \in C, p \in P$. Then, we follow a two-stage procedure at time $t \in T$:

a. Update the flow of link (i,j) on path $p \in P$; see Equations (4-1)-(4-7).

b. Update occupancies at cell $i \in C, C_S$ and C_O on path $p \in P$; see Equations (4-8)-(4-10).

Equation (4-1) finds the total flow ψ_{ij}^t on each ordinary link $(i,j) \in E_O$ at time step $t \in T$. To extract the path-level flow on $p \in P$, the total flow ψ_{ij}^t needs to be distributed among all paths based on $\xi_i^{-t} x_i^{t,p}$, as is shown in Equations (4-2) that can be re-formulated into Equations (4-3). Equations (4-4) and (4-5) determine the flow of the diverge links under two conditions: (1)

Conditions (4-4) will be enforced when the maximum outflow of a diverge cell $i \in C_D$ at time $t \in T$ is less than or equal to the corresponding saturation flow rate F_i or (2) Conditions (4-5) will be applied, otherwise. Equation (4-6) is a compact form of (4-4) and (4-5). Equations (4-7) find the path-level flow $y_{ij}^{t,p}$ by distributing the total flow ψ_{ij}^{t} among the diverge links of a diverge cell $i \in C_D$ at time $t \in T$ using $(\eta_{ij}^{t} + \mu)^{-1} x_i^{t,p}$. Equations (4-8), (4-9), and (4-10) update the cell occupancies at each time $t \in T$ on each path $p \in P$, for each ordinary cell $i \in C \backslash \{C_S, C_O\}$, source cell $i \in C_O$, and sink cell $i \in C_S$, respectively.

$$\psi_{ij}^{t} = \min\left\{\xi_i^t, F_i, F_j, \left(M_j - \xi_j^t\right)\right\} \qquad \forall t \in T, (i,j) \in E_O \tag{4-1}$$

$$y_{ij}^{t,p} = \begin{cases} \min\left\{\xi_i^t, F_i, F_j, \left(M_j - \xi_j^t\right)\right\} \dfrac{x_i^{t,p}}{\xi_i^t}, & if\ \xi_i^t > 0 \\ 0 & O.W. \end{cases} \qquad \forall t \in T, (i,j) \in E_O, p \in P \tag{4-2}$$

$$y_{ij}^{t,p} = \min\left\{\xi_i^t, F_i, F_j, \left(M_j - \xi_j^t\right)\right\} \frac{x_i^{t,p}}{\xi_i^t + \mu} \qquad \forall t \in T, (i,j) \in E_O, p \in P \tag{4-3}$$

$$if \sum\nolimits_{j \in S(i)} \left(\min\left\{\xi_i^t, F_i, F_j, \left(M_j - \xi_j^t\right)\right\}\right) \leq F_i, \psi_{ij}^{t} = \min\left\{\xi_i^t, F_i, F_j, \left(M_j - \xi_j^t\right)\right\} \qquad \forall\ t \in T, i \in C_D, j \in S(i) \tag{4-4}$$

$$if \sum_{j \in S(i)} \left(\min\left\{\xi_i^t, F_i, F_j, \left(M_j - \xi_j^t\right)\right\}\right) > F_i \qquad \psi_{ij}^{t} = \frac{\min\left\{\xi_i^t, F_i, F_j, \left(M_j - \xi_j^t\right)\right\}}{\sum_{j \in S(i)} \left(\min\left\{\xi_i^t, F_i, F_j, \left(M_j - \xi_j^t\right)\right\}\right)} F_i \qquad \forall t \in T, i \in C_D, j \in S(i) \tag{4-5}$$

$$\psi_{ij}^{t} = \min\left\{\xi_i^t, F_i, F_j, \left(M_j - \xi_j^t\right)\right\} \min\left\{1, \frac{F_i}{\sum_{j \in S(i)} \left(\min\left\{\xi_i^t, F_i, F_j, \left(M_j - \xi_j^t\right)\right\}\right) + \mu}\right\} \qquad \forall t \in T, i \in C_D, j \in S(i) \tag{4-6}$$

$$y_{ij}^{t,p} = \psi_{ij}^{t} \; \frac{\gamma_{ij}^{t,p}}{\eta_{ij}^{t}+\mu} \qquad \forall t \in T, i \in C_D, j \in S(i) \tag{4-7}$$

$$y_{ki}^{t,p} - y_{ij}^{t,p} = x_i^{t+1,p} - x_i^{t,p} \qquad \forall\ t \in T, i \in\ C \setminus \{C_S\ , C_O\}, k \in P(i), j \in S(i), p \in P \tag{4-8}$$

$$\mathcal{D}^{t,p} - y_{ij}^{t,p} = x_i^{t+1,p} - x_i^{t,p} \qquad \forall t \in T, i \in C_O\ , j \in S(i), p \in P \tag{4-9}$$

$$y_{ki}^{t,p} = x_i^{t+1,p} - x_i^{t,p} \qquad \forall t \in T, i \in C_S, k \in P(i), p \in P \tag{4-10}$$

4.2. Step 1. Update

To implement the Dantzig-Wolfe decomposition, we have employed the block diagonal structure of SODTA that helps break down the problem into OD pairs. In the following formulation, all constraints except for Constraints (3-7)-(3-10) are defined for each $od \in C_{OD}$. Therefore, we can categorize the constraints based on OD pairs to define different sets of constraints, where each set represents one OD pair. Each set of constraints forms a sub-problem. Constraints (3-7)-(3-10) are in none of the sets or sub-problems. We include these constraints in the master problem. In other words, constraints (3-7)-(3-10) on total flows of ODs are not decomposable unless they are relaxed, leading to an RMP and several SPs. Note that constraints (3-7)-(3-10) for one OD are included in SPs for faster convergence. The RMP selects a combined optimal solution that satisfies all relaxed constraints coupling the SPs. In each iteration of the algorithm, the optimal solution to the RMP is also feasible for (3-2)-(3-12). The dual values of the RMP are added as the modification indicators of the violations of the partially relaxed constraints, i.e., penalty terms, to the objective function of the SPs. The procedures proposed to determine the optimal solutions to the MP and SPs follow.

4.2.1. Step 1.1. Restricted Master Problem

The RMP is formulated using (3-2), (3-7)-(3-10) and a set of convexity constraints. In this procedure, $x_{i,e}^{t,od}$and $y_{ij,e}^{t,od}$ are identified as parameters and λ_e^{od} is defined as the decision variable, optimized for each $od \in C_{OD}$ for the set of extreme points $e \in E$. For each $od \in C_{OD}$, a convexity constraint $\sum_{\forall e \in E} \lambda_e^{od} = 1$ will be added to the RMP (Constraints (4-16)). Thus, the RMP will be formulated as follows.

$$\min \sum_{t \in T} \sum_{i \in C \setminus C_S} \sum_{od \in C_{OD}} \sum_{e \in E} \lambda_e^{od} x_{i,e}^{t,od} \tag{4-11}$$

$$\sum_{j \in S(i)} \sum_{od \in C_{OD}} \sum_{e \in E} \lambda_e^{od}\, y_{ij,e}^{t,od} \leq F_i \qquad \forall t \in T, i \in C \setminus C_S \tag{4-12}$$

$$\sum_{i \in P(j)} \sum_{od \in C_{OD}} \sum_{e \in E} \lambda_e^{od}\, y_{ij,e}^{t,od} \leq F_j \qquad \forall t \in\ T, j \in C \setminus C_O \tag{4-13}$$

$$\sum_{od \in C_{OD}} \sum_{e \in E} (\sum_{i \in P(j)} \ \lambda_e^{od}\, y_{ij,e}^{t,od}) + \lambda_e^{od} x_{j,e}^{t} \leq M_j \qquad \forall t \in T, j \in C \setminus C_O \tag{4-14}$$

$$\sum_{j \in S(i)} \sum_{od \in C_{OD}} \sum_{e \in E} \lambda_e^{od}\, y_{ij,e}^{t,od} \leq f_i^t \qquad \forall t \in T, i \in C_I \tag{4-15}$$

$$\sum_{e \in E} \lambda_e^{od} = 1 \qquad \forall od \in C_{OD} \tag{4-16}$$

$$\lambda_e^{od} \geq 0 \qquad \forall od \in C_{OD}, e \in E \tag{4-17}$$

Objective function (4-11) aims to minimize the convex combination of the total travel time of all cells except for the sink cells $i \in\ C \setminus C_S$ over all time steps $t \in T$, all OD pairs $od \in C_{OD}$, and the set of extreme points $e \in E$. Constraints (4-12)-(4-15) compensate for the violation of the partially relaxed constraints (in the solutions of the SPs). Finally, Constraints (4-17) ensure the non-negativity of λ_e^{OD}.

4.2.2. Step 1.2. Sub-problems

Each SP is presented for only one OD pair; hence, the od index will be eliminated from all variables in the updated formulation (Constraints (4-20)-(4-29)). Constraints (3-7)-(3-10) are reserved for exclusive application to the total flow of all OD pairs. SPs aim to maximize the reduced cost of the RMP. To formulate the objective function, we define the dual variables of the RMP as follows. We let $\pi_{i,j}^{t}$ denote the dual variable for cell $i \in C$ for coupling constraint j at time $t \in T$. Besides, φ_{od} represents the dual variable for convexity constraints on $OD \in C_{OD}$. In other words, $\pi_{i,14}^{t}$, $\pi_{i,15}^{t}$, $\pi_{i,16}^{t}$, $\pi_{i,17}^{t}$, and φ_{od} respectively denote the dual variables of constraints (4-12)-(4-16) in the RMP. The reduced costs of the RMP are introduced as:

$$\begin{aligned} -\sum_{i\in C\setminus C_S}\sum_{t\in T} x_i^t + \sum_{i\in C\setminus C_S}\sum_{t\in T}\pi_{i,14}^t \sum_{j\in S(i)} y_{ij}^t + \sum_{i\in C\setminus C_O}\sum_{t\in T}\pi_{i,15}^t \sum_{i\in P(j)} y_{ij}^t & \\ + \sum_{i\in C\setminus C_O}\sum_{t\in T}\pi_{i,16}^t (\sum_{i\in P(j)} y_{ij}^t + x_i^t) & \qquad \forall od \in C_{OD} \qquad \text{(4-18)} \\ + \sum_{i\in C_I}\sum_{t\in T}\pi_{i,17}^t \sum_{j\in S(i)} y_{ij}^t + \varphi_{od} & \end{aligned}$$

Variable λ_e^{od} can be introduced as a basic variable in the RMP solution if (4-18) is greater than zero. We maximize (4-19) as the objective function of each SP. φ_{od} is eliminated from (4-19) since it is a constant value; however, it will be included later for the algorithm's termination. If the generated optimal solutions guarantee strictly greater than zero reduced costs, considering $-\varphi_{od}$, the solutions (i.e., $x_i^{t,od}$and $y_{ij}^{t,od}$) will be introduced to the RMP as a set of input parameters.

$$\max - \sum_{i\in C\backslash C_S}\sum_{t\in T} x_i^t + \sum_{i\in C\backslash C_S}\sum_{t\in T}\pi_{i,14}^t \sum_{j\in S(i)} y_{ij}^t + \sum_{i\in C\backslash C_O}\sum_{t\in T}\pi_{i,15}^t \sum_{i\in P(j)} y_{ij}^t$$
$$+ \sum_{i\in C\backslash C_O}\sum_{t\in T}\pi_{i,16}^t \left(\sum_{i\in P(j)} y_{ij}^t + x_i^t\right) + \sum_{i\in C_I}\sum_{t\in T}\pi_{i,17}^t \sum_{j\in S(i)} y_{ij}^t \tag{4-19}$$

$$\sum_{k\in P(i)} y_{ki}^t - \sum_{j\in S(i)} y_{ij}^t = x_i^{t+1} - x_i^t \qquad \forall t \in T, i \in C\backslash\{C_S\,, C_O\} \tag{4-20}$$

$$d_i^t - \sum_{j\in S(i)} y_{ij}^t = x_i^{t+1} - x_i^t \qquad \forall t \in T, i \in C_O \tag{4-21}$$

$$\sum_{i\in P(j)} y_{ij}^t = x_j^{t+1} - x_j^t \qquad \forall t \in T, j \in C_S \tag{4-22}$$

$$\sum_{j\in S(i)} y_{ij}^t \le x_i^t \qquad \forall t \in T, i \in C \tag{4-23}$$

$$\sum_{j\in S(i)} y_{ij}^t \le F_i \qquad \forall t \in T, i \in C \tag{4-24}$$

$$\sum_{i\in P(j)} y_{ij}^t \le F_j \qquad \forall t \in \ T, j \in C \tag{4-25}$$

$$\sum_{i\in P(j)} y_{ij}^t \le M_j - x_j^t \qquad \forall t \in T, j \in C \tag{4-26}$$

$$f_i^t = g_i^t F_i \qquad \forall t \in T, i \in C_I \tag{4-27}$$

$$x_i^t \ge 0 \qquad \forall t \in T, i \in C \tag{4-28}$$

$$y_{ij}^t \ge 0 \qquad \forall t \in T, i \in C/C_s, j \in P(i) \tag{4-29}$$

4.3. Termination Criterion

To evaluate the solution quality of the proposed algorithm, the Upper Bound (UB) and the Lower Bound (LB) are computed in each iteration. The objective value of the RMP represents the UB and hence, a feasible solution with a specified optimality gap will be derived in each iteration.

The LB is computed by subtracting the non-negative objective value of the SPs from the UB. Note that it is proven that the proposed solution technique will find the optimal solution to the problem in a finite number of iterations (Dantzig and Wolfe, 1960). However, finding the optimal solutions may require an excessive number of iterations and runtime. Therefore, we have set the termination criterion to reach a 2% gap between UB and LB or 500 iterations.

4.4. Distributed Computing Framework

Distributed and parallel computing techniques generate concurrently-solvable problem elements and utilize the capabilities offered by multi-processor machines to efficiently solve complex problems (Adeli, 2000; Adeli and Kamal, 2014). Although parallel algorithms have been applied to different traffic assignment problems, the structure of CTM-based SODTA formulations with multiple ODs has not been exploited for high-performance optimization computations. For instance, Chen and Meyer (1988) have presented a parallel algorithm for an approximate multi-commodity STA problem, where all commodities are optimized in parallel. A disaggregate simplicial decomposition algorithm presented by Larsson & Patriksson (1992) to solve STA problems is parallelized by OD pairs (Karakitsiou et al., 2004; Lotito, 2006). Parallel frameworks are also used for DTA simulations. However, these simulations are not as computationally expensive as the optimization techniques (Qu and Zhou, 2017; Rickert and Nagel, 2001). On the other hand, the literature presents extensive studies that aim to maximize the computational efficiency by parallel algorithms that utilize macro-tasking, micro-tasking, and vectorization features of super-computers; see Saleh & Adeli (1994); Saleh & Adeli (1996); Soegiarso & Adeli (1994); Saleh & Adeli (1997). Micro-tasking and vectorization are conducted at the loop level, and macro-tasking parallelizes functions (Adeli and Hung, 1993; Saleh and Adeli, 1994). Besides, Adeli and Kumar, 1995; Kumar and Adeli (1995) have implemented distributed algorithms on a cluster with a costly communication architecture among workstations. Adeli & Kamal (1992a) and Adeli & Kamal (1992b) have parallelized the computation components while maintaining a workload balance among processors.

Our proposed algorithm is partitioned into multiple tasks, as shown in Figure 4-3, where each is assigned to a new or the same Computing Processor Unit (CPU). Tasks either take place simultaneously or are implemented sequentially. According to Ziliaskopoulos, Kotzinos, and Mahmassani (1997), a task is a code script that is executed on a CPU. A cluster of workstations can be efficiently employed if the computation tasks are much more than the communication ones because the workstations are coupled loosely in clusters and the communication is slow and inefficient (Adeli and Kumar, 1998). In our algorithm, the main computational tasks are the optimization of SPs that do not require any message passing, so it is suitable to be run on clusters. The synchronization of SPs can improve the computational efficiency of the algorithm substantially because the algorithm should not wait to optimize all SPs one by one sequentially to be able to go the next step. Moreover, increasing SPs by the addition of OD pairs will not affect the required time for the optimization of SPs.

The presented algorithm is implemented on a High-Performance Computing (HPC) cluster. The number of nodes and processors that are requested from HPC to run the algorithm are one and one plus the number of ODs, respectively.

The algorithm starts with reading input data using the first processor. All the following tasks are being processed with the same processor and sequentially:

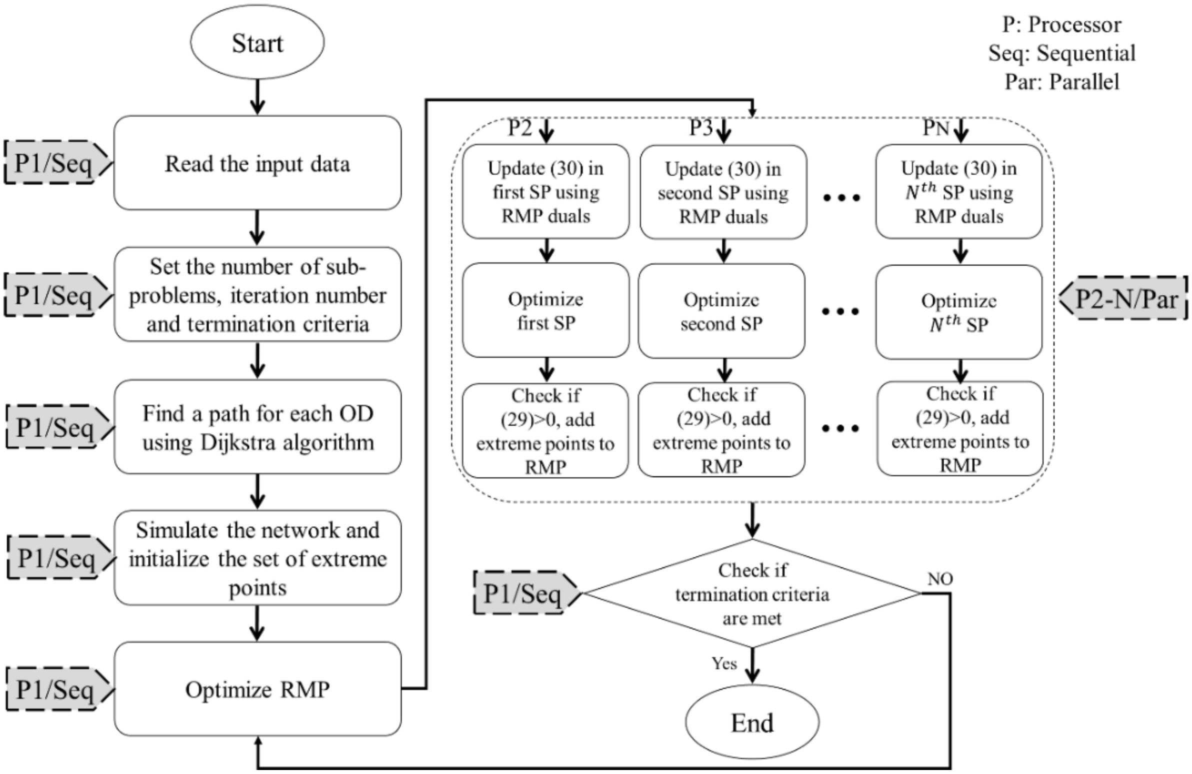
- Read the input data
- Set the number of sub-problems equal to the number of OD pairs
- Set the iteration number to zero

- Set the termination criteria (maximum number of iterations and gap between UB and LB)
- Find a path for each OD using the Dijkstra algorithm
- Simulate the network
- Initialize the set of extreme points with the outputs of the simulation
- Optimize RMP

Then, the following tasks are implemented concurrently by new threads. New threads are assigned to the set of tasks that are placed in one column:

- Update the objective functions of SPs
- Optimize SPs
- Add the extreme points generated by SPs to MP if the objective function of SP is strictly positive

All three tasks for each SP are assigned to a node, and the access of all threads to the shared arrays is delayed until all threads finish their jobs. Then, the arrays are shared among SPs to insert their outputs. Finally, the algorithm checks if the termination criteria are met using the first processor. If so, the algorithm is stopped. Otherwise, the RMP is optimized, and the algorithm continues.



Start
P: Processor
Seq: Sequential
Par: Parallel
P1/Seq
Read the input data
P1/Seq
Set the number of sub-problems, iteration number and termination criteria
P1/Seq
Find a path for each OD using Dijkstra algorithm
P1/Seq
Simulate the network and initialize the set of extreme points
P1/Seq
Optimize RMP
P2
P3
PN
Update (30) in first SP using RMP duals
Update (30) in second SP using RMP duals
Update (30) in N^{th} SP using RMP duals
Optimize first SP
Optimize second SP
Optimize N^{th} SP
P2-N/Par
Check if (29)>0, add extreme points to RMP
Check if (29)>0, add extreme points to RMP
Check if (29)>0, add extreme points to RMP
P1/Seq
Check if termination criteria are met
NO
Yes
End

Figure 4-3 The outline of the algorithm's implementation

4.5. Numerical Experiments

This section presents the characteristics of test networks and test scenarios used to evaluate the performance of the proposed methodology. Two networks include 20 (4×5) and 40 (4×10) intersections. The 20 intersection network is shown in Figure 4-4. 40 intersection network is a duplication of the 20 intersection network. Table 4-2 shows a list of scenarios with a different number of intersections, OD pairs, and the type of demand. Table 4-3 shows the demand profiles for each OD pair and all scenarios. The network loading interval is assumed to be 30 *min* with an additional 10 *min* for unloading the network. The proposed methodology is coded in Java Eclipse and run on a Linux-based cluster. In the test network of 20 intersections, we have used one node of a High-Performance Computer (HPC) with 21 cores with a total of 16.0 GB of memory. In addition, we have used one HPC node with 41 cores in the test network of 40 intersections. In both cases, the master problem is solved in one core, and each sub-problem is solved in a separate core in parallel. The CPLEX libraries are called via Java scripts to solve linear programs. The duration of each time step is six seconds, the number of cells in each link is between two to four, the total number of cells in the network is 316 for 20 intersections and 632 for 40 intersections, the free-flow speed is 25 mph, length of each cell is 220 ft, and saturation flow rate is 3 vehicle/time step/lane.

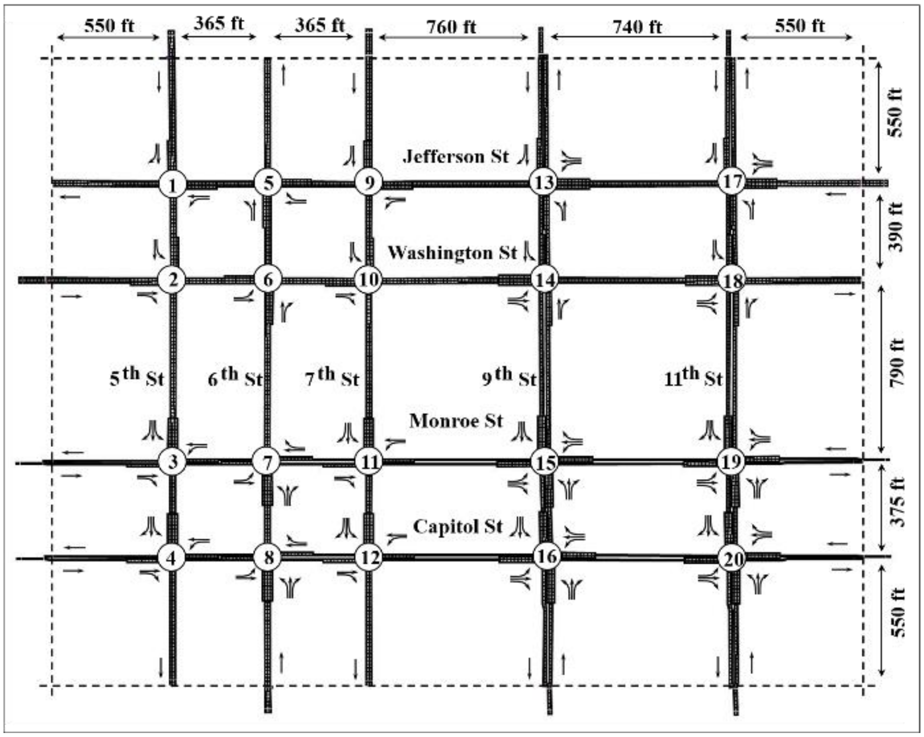


Figure 4-4 Test network 1: 20 intersections –Springfield, IL

Table 4-2 List of tested scenarios

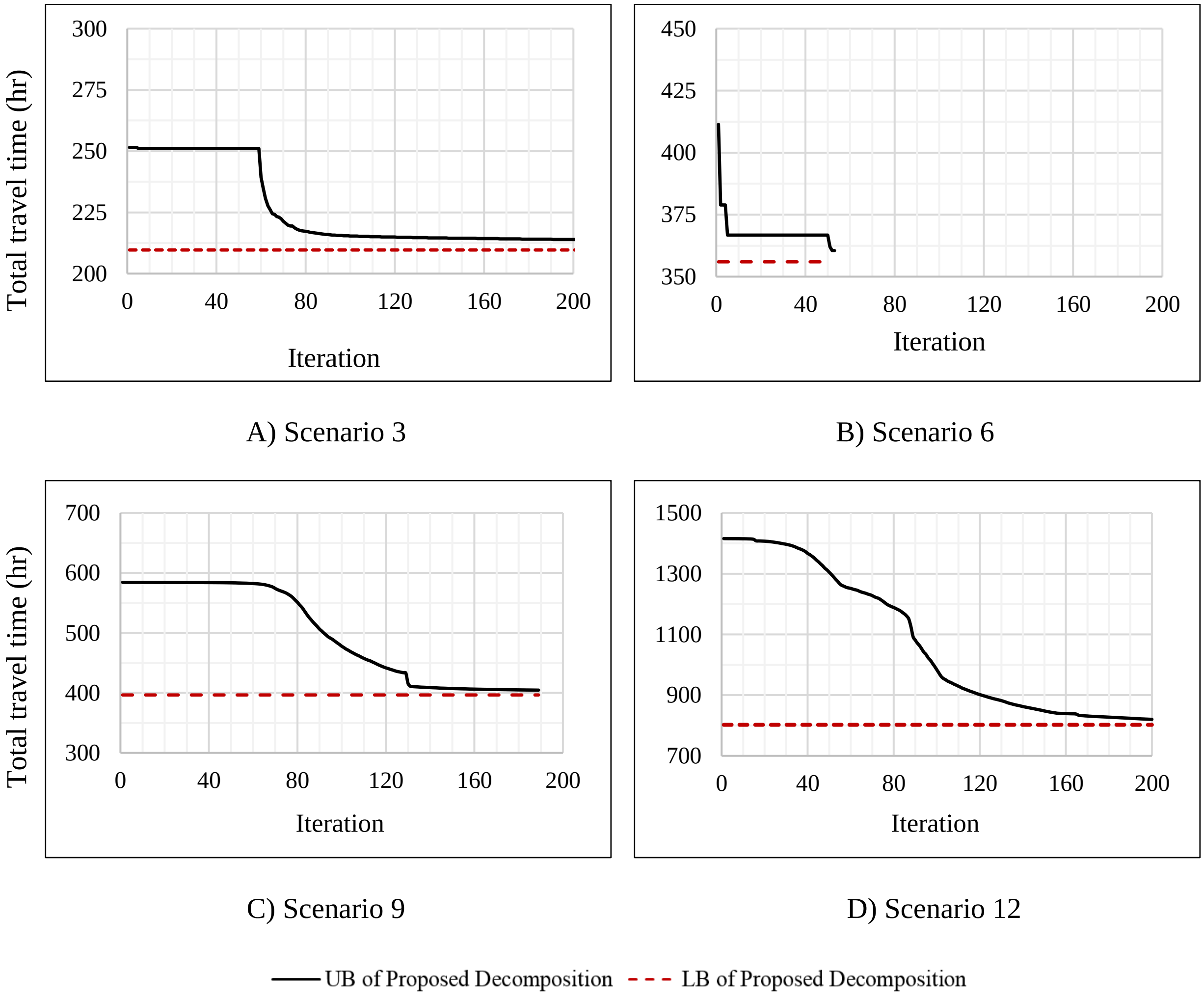
Scenario	Number of intersections	Number of OD pairs	Demand
1	20	5	Under-saturated
2	20	5	Semi-saturated
3	20	5	Over-saturated
4	20	10	Under-saturated
5	20	10	Semi-saturated
6	20	10	Over-saturated
7	20	15	Under-saturated
8	20	15	Semi-saturated
9	20	15	Over-saturated
10	40	25	Under-saturated
11	40	25	Semi-saturated
12	40	25	Over-saturated

Table 4-3 Demand profiles loaded into the test networks of 20 and 40 intersections

Network of 20 intersections							
OD/Demand (veh/hr/ln)	Under-saturated	Semi-saturated	Over-saturated	OD/Demand (veh/hr/ln)	Under-saturated	Semi-saturated	Over-saturated
1	333	500	750	9	133	200	300
2	133	200	300	10	133	200	300
3	333	500	750	11	333	500	750
4	333	500	750	12	333	500	750
5	67	100	150	13	333	500	750
6	333	500	750	14	333	500	750
7	333	500	750	15	67	100	150
8	333	500	750				
Network of 40 intersections							
OD/Demand (veh/hr/ln)	Under-saturated	Semi-saturated	Over-saturated	OD/Demand (veh/hr/ln)	Under-saturated	Semi-saturated	Over-saturated
1	267	400	600	14	320	480	720
2	333	500	750	15	333	500	750
3	267	400	600	16	160	240	360
4	320	480	720	17	160	240	360
5	320	480	720	18	67	100	150
6	320	480	720	19	160	240	360
7	200	300	450	20	120	180	270
8	333	500	750	21	120	180	270
9	333	500	750	22	320	480	720
10	333	500	750	23	160	240	360
11	200	300	450	24	120	180	270
12	320	480	720	25	67	100	150
13	200	300	450				

4.6. Results

This section presents a set of numerical experiments to show the performance of the proposed decomposition algorithm in solving the CTM-based SODTA problem. Figure 4-5 displays UB and LB over iterations for scenarios with oversaturate demand (i.e., 3, 6, 9, and 12). The proposed algorithm has achieved a 2% optimality gap in less than 200 iterations in all tested scenarios.



A) Scenario 3

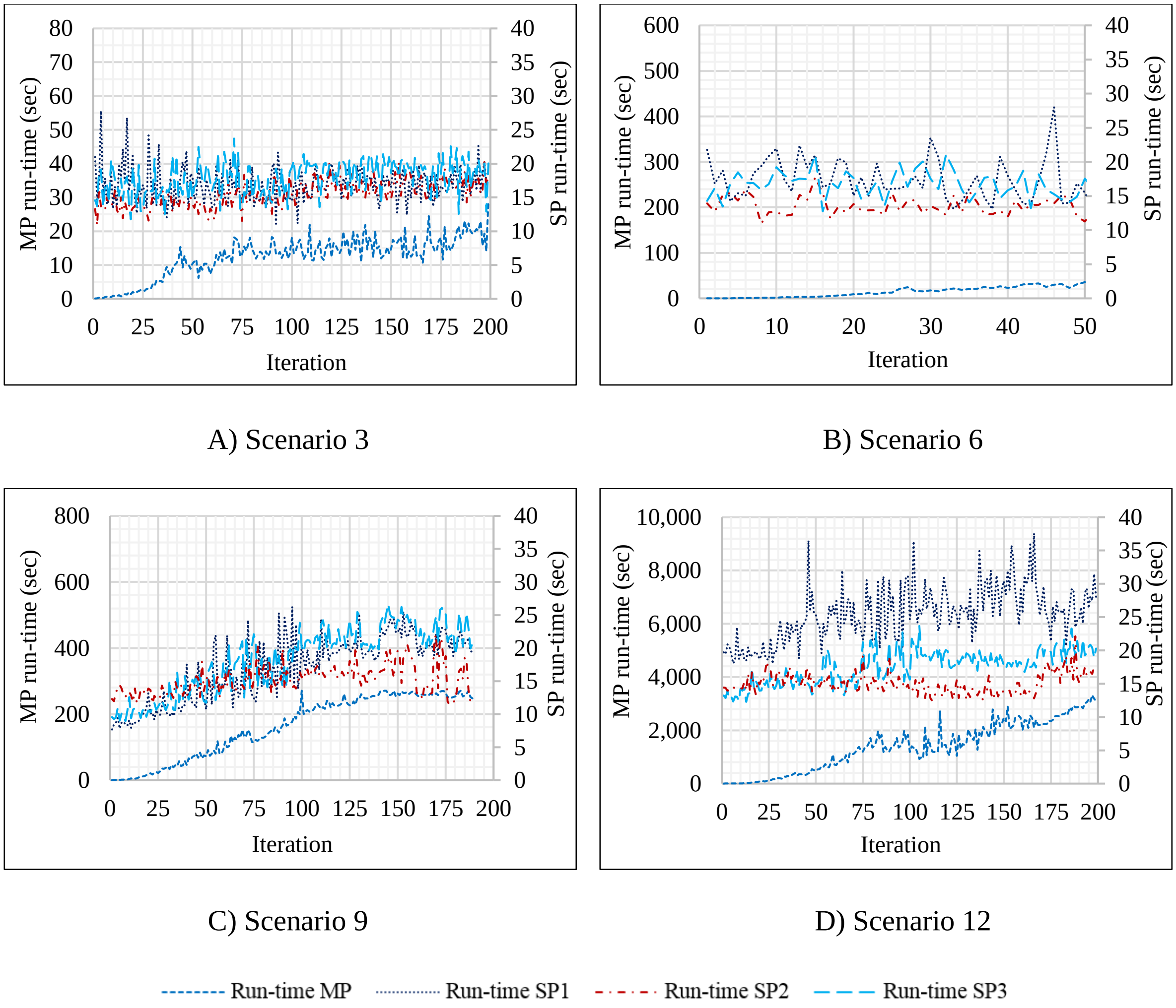
B) Scenario 6

C) Scenario 9

D) Scenario 12

Figure 4-5 UB and LB of the proposed algorithm in Scenarios 3, 6, 9, and 12

Figure 4-6 shows the runtime per iteration of the proposed algorithm to solve SODTA up to the termination point in scenarios 3, 6, 9, and 12. As expected, the average runtime of SPs does not change over iterations since their complexity is the same. On the other hand, it can be observed that the runtime of MPs is increased slightly in each iteration because new decision variables λ_e^{od} are added. The runtime of other parts of the algorithm including the SP procedure is constant. Note that the number of decision variables in scenario 12 is more than 14 million as opposed to 4 million in scenario 9. However, the total runtime in scenario 12 is only 52% more than that of scenario 9, which is a strong indicator of the scalability of the proposed solution algorithm.



A) Scenario 3

B) Scenario 6

C) Scenario 9

D) Scenario 12

Figure 4-6 Run-time of 3 SPs and MP in Scenarios 3, 6, 9, and 12

4.6.1. Benchmark Solutions

CPLEX is used to find optimal solutions to the original problem as the benchmark. CPLEX is given the flexibility to select the most appropriate method to find the optimal solutions. Figure 4-

7 shows the marginal objective value, i.e., the difference between the optimal objective values of CPLEX and the decomposition technique and optimality gap (of the decomposition technique), based on all scenarios. Note that the marginal objective values are not available in Scenarios 7-12 as the CPLEX required more memory to find solutions (CPLEX was run on a Linux-based cluster with 16 GB of memory). Among all scenarios, the maximum marginal objective value is 1.19%, and the optimality gap is reported less than 2%, which was the defined termination criteria. Note that the optimality gaps and marginal objective value are both zero for scenario 2.

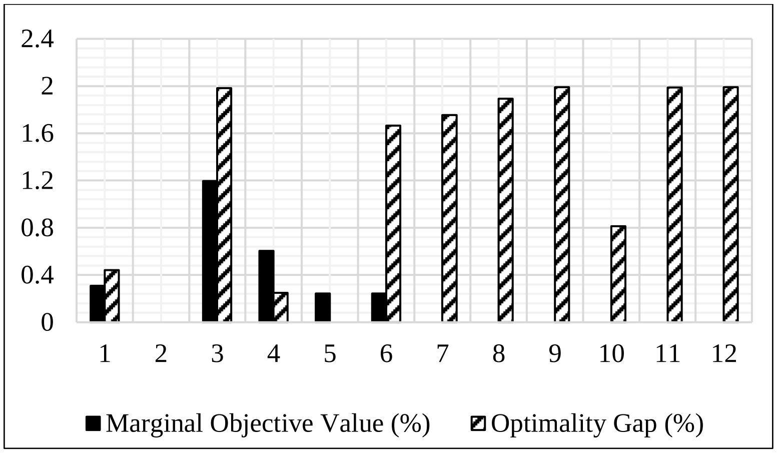


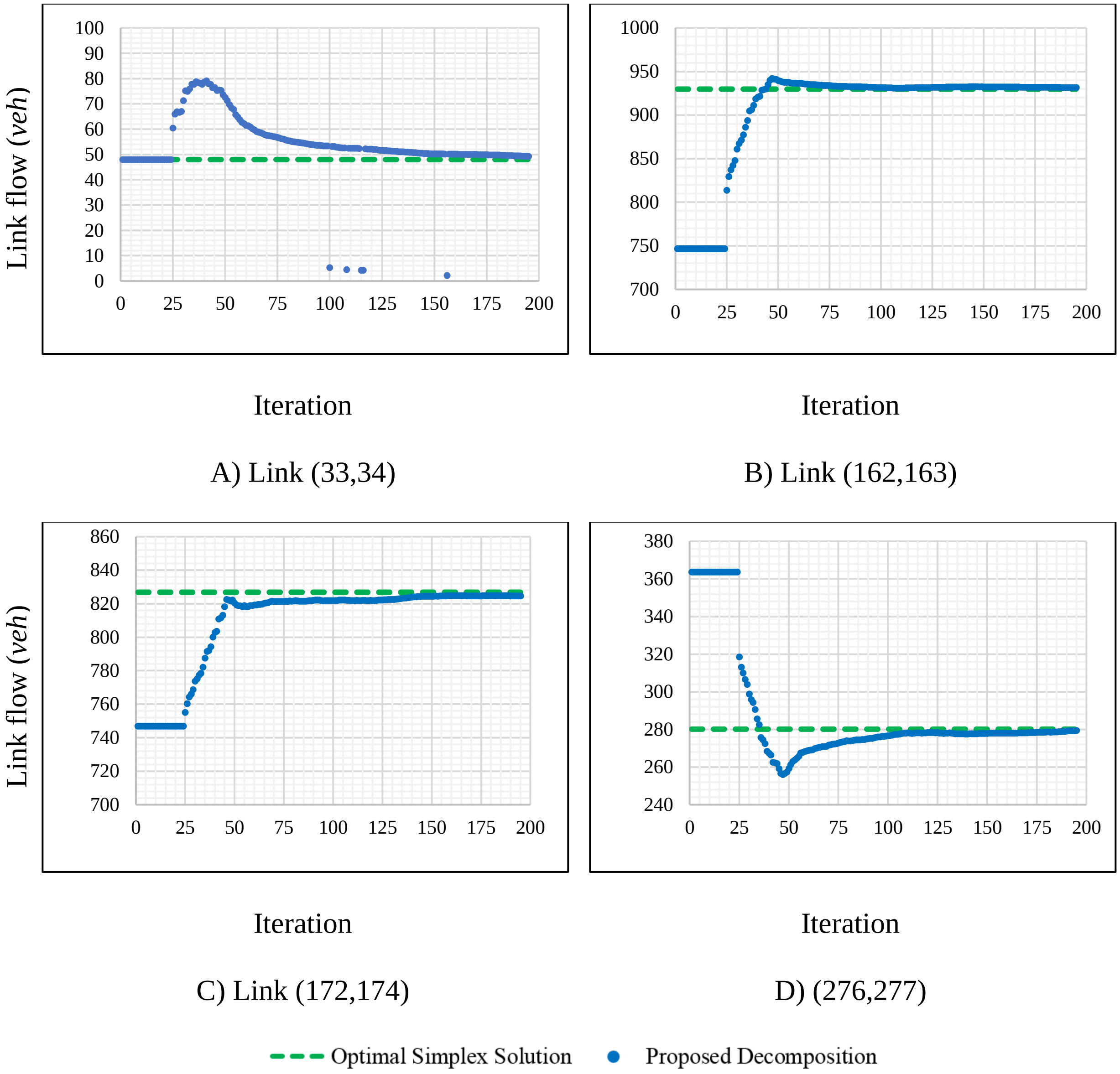
Figure 4-7 Marginal objective value and optimality gap in Scenarios 1-12

Table 4-4 presents the total travel time and the computation time for all scenarios solved with the decomposition algorithm and CPLEX. Since CPLEX fails to provide the optimal solutions in scenarios 7-12, the difference between travel time and computation times is not available. The maximum difference between the travel time found by the solutions of the proposed algorithm and the benchmark approach is 2.00%. Note that the algorithm can reach a much lower optimality gap at the cost of higher CPU times. For example, the algorithm requires 343.09 minutes of computation time to reach a 0.01% optimality gap in scenario 7. Moreover, CPLEX is more computationally expensive with similar computational resources compared to the proposed algorithm in all scenarios. In Scenarios 1-2 and 4-5, the proposed algorithm can find near-optimal solutions in the initial iterations, which implies that the initial solution obtained by the path-based simulation is very close to the optimal solution. This is not the case for CPLEX since it does not follow an initialization-update procedure.

Table 4-4 Network performance measures for Scenarios 1-12

Performance measures	Algorithm / Gap	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5	Scenario 6
Total travel time (hr)	CPLEX	82.03	140.34	210.81	145.12	221.18	354.75
	Proposed	82.28	140.34	213.88	145.38	221.73	362.00
	difference (%)	0.31	0.00	1.98	0.17	0.24	1.66
Total computation time (min)	CPLEX	1.90	3.09	58.68	46.01	69.36	790.2
	Proposed	0.06	0.06	45.61	0.12	0.12	28.179
	difference (%)	-96.84	-98.06	-22.27	-99.74	-99.83	-96.43
Performance measures	Algorithm / Gap	Scenario 7	Scenario 8	Scenario 9	Scenario 10	Scenario 11	Scenario 12
Total travel time (hr)	CPLEX	NA*	NA	NA	NA	NA	NA
	Proposed	170.16	262.59	464.92	265.70	474.00	812.52
	difference (%)	NA	NA	NA	NA	NA	NA
Total computation time (min)	CPLEX	NA	NA	NA	NA	NA	NA
	Proposed	4.56	33.71	438.18	8.81	97.67	2927.41
	difference (%)	NA	NA	NA	NA	NA	NA
*not available (NA)							

Figure 4-9 illustrates the convergence of the link flows to the optimality in Scenario 3 based on the proposed decomposition scheme. The total link flows are calculated by the accumulation of flows over all OD pairs and time steps. We have selected four CTM-based links (33,34), (162,163), (172,174) and (276,277) as network representatives. In all cases, the link flows are converged to their optimal values with smooth fluctuations by 200 iterations.



A) Link (33,34)

B) Link (162,163)

C) Link (172,174)

D) (276,277)

Figure 4-8 Total link flows in the proposed algorithm compared to the optimal Simplex solution in Scenario 3

To parallelize the proposed decomposition scheme, a multi-thread program is coded in Java-Eclipse. Detailed information on parallelization is available in section 4.4. Each SP on a single OD is assigned to a specific processor and each thread is executed on a different directory. This thread-based program allows the concurrent optimization of the SPs. The sequential and parallel modes are compared through the generation of an experimental scenario that assigns a single processor to the sequential mode and 16 processors to the parallel mode. In the parallel computing procedure, 15 threads are generated, where each handles the SP of a single OD. Besides, one thread is created for the MP. Figure 4-8 shows the runtime of the sequential and parallel modes over 5 *min*. The sequential mode can only complete 15 iterations in 5 *min*, while the parallel programming enabled by the proposed decomposition technique allows 40 iterations, which is a substantial improvement in the computation time.

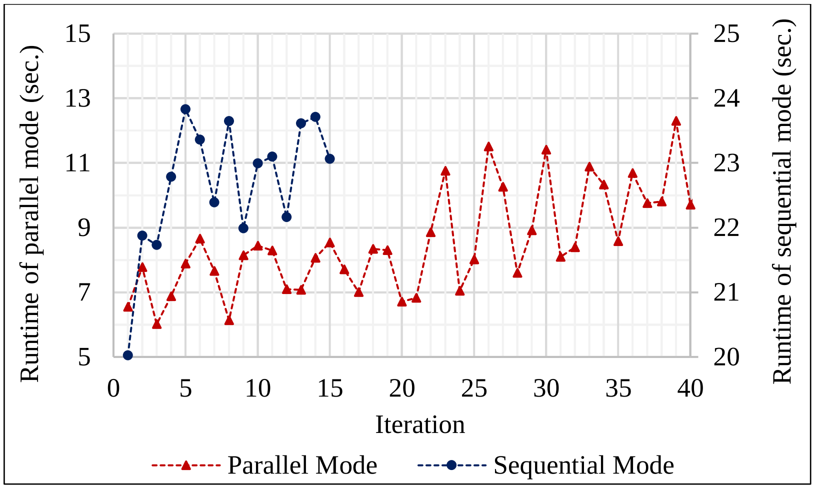


Figure 4-9 Runtime comparison of sequential and parallel architectures for 5 *min* runtime in Scenario 7

CHAPTER 5. DISTRIBUTED GRADIENT-BASED APPROACH FOR SOLVING SODTA

This chapter presents a distributed gradient-based methodology to solve the SODTA problem. The discussions are continued in five subsections: (1) the distribution of SODTA problem formulation, (2) initialization, (3) the gradient-based update, (4) termination criteria, and (5) convergence properties. In the first section, we partition the cell-based SODTA formulation among sub-problems with an intersection-level segmentation so that the number of sub-problems is equal to the number of intersections. Each sub-problem contains some parts of the objective function and constraints of the original SODTA formulation that have decision variables corresponding to the cells and links within the intersection region assigned to the sub-problem.

In the initialization step, we find initial values for the decision variables of each sub-problem. We start by generating the shortest paths for each OD pair using Dijkstra's algorithm (Ahuja et al., 1993) and sending the demand to the network through these paths using a path-based CTM simulation (Mehrabipour et al. 2019; Ukkusuri, Han, and Doan 2012). We initialize the approach with the occupancy and flow values that are the outputs of the simulation. Then, we update the value of the decision variables of each sub-problem iteratively through a distributed gradient-based step. For each sub-problem, we incorporate the proposed values by themselves and other sub-problems for the shared decision variables of the sub-problem by taking a weighted average. We find the weighted average at the current iteration for all sub-problems using the value of decision variables from either the initialization step or the previous iteration. Then, the values are moved towards the negative direction of the gradient of the objective function of each sub-problem (to minimize it), and the values of the decision variables are projected on the set of constraints at each sub-problem to maintain feasibility. The approach iterates until the conflict among the proposed values from the sub-problems is within an acceptable threshold. We will show that the methodology converges to the optimal solution of the problem by an infinite number of iterations. Table 5-1 shows the notations used in this chapter. Figure 5-1 shows the overall framework of the methodology, and more details about each part of the figure are provided in the rest of this section. Note that this study is published in IEEE Transactions in Intelligent Transportation Systems Journal (Mehrabipour and Hajbabaie, 2022a), and, for materials used in this chapter, the full credit is given to the original source © 2022 IEEE.

Table 5-1 Definition of sets, decision variables, and parameters used in chapter 5 © 2022 IEEE

Sets	
T	The set of time steps
C	The set of network cells
C_O	The set of source cells
C_S	The set of sink cells
C_I	The set of intersection cells
C_{OD}	The set of OD pairs
$P(i)$	The set of predecessors of cell $i \in C$
$S(i)$	The set of successors of cell $i \in C$
C_D	The set of diverge cells
N	The set of sub-problems
Cl	The set of links that their heads $i \in C^s$and tails $j \in S(i)$ belong to sub-problems $e \in C^{\mathscr{s} \in n^s}$ and $f \in S(e)$ such that $i = e, j = f, s \neq \mathscr{s}$
$\mathfrak{n}^s$	The set of neighbors of sub-problem $s \in N$ including itself
C^s	The set of cells in sub-problem $s \in N$
Co^s	The union of the set of cells in sub-problem $s \in N$ and the set of source cells, i.e., $C_O \cap C^s$
Cs^s	The union of the set of cells in sub-problem $s \in N$ and the set of sink cells, i.e., $C_S \cap C^s$
C_D^s	The set of diverge cells in sub-problem $s \in N$
$Ceos^s$	The set of cells in sub-problem $s \in N$ except for the set of source cells and the set of sink cells i.e., $C^s / C_O \cup C_S$

Table 5-1 (continued)

Ces^s	The set of cells in sub-problem $s \in N$ except for the set of sink cells, i.e., C^s/C_S
Ceo^s	The set of cells in sub-problem $s \in N$ except for the set of source cells, i.e., C^s/C_O
Ci^s	The union of the set of cells in sub-problem $s \in N$ and the set of intersection cells, i.e., $C_I \cap C^s$
X_s	The set of constraints of $s$ubproblem $s \in N$
X	The feasible region of the SODTA problem
X^*	The set of optimal solutions
Decision variables	
$x_i^{t,od}$	The number of vehicles in cell $i \in C$ at time step $t \in T$ for OD pair $(o,d) \in C_{OD}$
$y_{ij}^{t,od}$	The number of vehicles flowing from cell $i \in C$ to downstream cell $j \in S(i)$ at time step $t \in T$ for OD pair $(o,d) \in C_{OD}$
Parameters	
τ	The duration of time step $t \in T$
$\mathcal{D}_i^{t,od}$	The entry demand level at source cell $i \in C_O$ at time step $t \in T$ for OD pair $(o,d) \in C_{OD}$
F_i	The saturation flow rate at cell $i \in C$
M_i	The maximum number of vehicles that cell $i \in C$ can accommodate
δ	The ratio of free-flow speed to the backward propagation speed
g_i^t	A binary parameter to define signal status at intersection cell $i \in C_I$ at time step $t \in T$. Zero and one values indicate red and green signals, respectively
$R_{ij}^{t,od}$	The turning ratio of the link between diverge cell $i \in C_D$ and its successor cell $j \in S(i)$ at time step $t \in T$ for OD pair $(o,d) \in C_{OD}$
f_i^t	The variable saturation flow rate of intersection cell $i \in C_I$ at time step $t \in T$
Notations and Terms	
K	The total number of iterations
k	The iteration counter
Ӽ_s^k	The column vector of values for the decision variables of sub-problem $s \in N$ at iteration $k \in K$
$\boldsymbol{x}$ and $\boldsymbol{z}$	The column vectors
$\boldsymbol{x}'$	The transpose of vector x
$\boldsymbol{x}'\mathbf{z}$	The dot product of two vectors x and z
$\boldsymbol{z}^* \in \mathrm{X}^*$	A realization of the vector of optimal solution

Table 5-1 (continued)

$\|\boldsymbol{x}\| = (\boldsymbol{x}'\boldsymbol{x})^{\frac{1}{2}}$	The standard Euclidean norm
$\boldsymbol{P}_X[\boldsymbol{x}]$	The projection of vector x on set X
$\mathcal{F}$	The objective function of the SODTA problem
$\mathcal{F}_s(\boldsymbol{x})$	The objective function of sub-problem $s \in N$ at vector x
$\mathcal{F}^*$	The optimal objective function
$G = (N, A)$	The information exchange graph with a set of nodes N and a set of directional links A
$(s, \mathscr{s}) \in A$	A link in information exchange graph $G = (N, A)$
$\mathscr{w}_{s\mathscr{s}}$	The weight on the link going from node s to node $\mathscr{s}$ in information exchange graph $G = (N, A)$
θ and θ'	The lower bound and the upper bound for weight values $\mathscr{w}_{s\mathscr{s}}: s\mathscr{s} \in N$
$\mathscr{w}$	The weighted graph Laplacian matrix with weight $\mathscr{w}_{s\mathscr{s}}$ entries assigned to links $(s, \mathscr{s}) \in N$ in information exchange graph $G = (N, A)$
$\boldsymbol{\mathcal{G}}_s$	The gradient of the objective function $\mathcal{F}_s$ of sub-problem $s \in N$
$\mathcal{G}_{s,i}^{t,od}$	The gradient of objective function of sub-problem $s \in N$ with respect to $x_i^{t,od}$
$\mathcal{G}_{s,ij}^{t,od}$	The gradient of objective function of sub-problem $s \in N$ with respect to $y_{ij}^{t,od}$
α^k and γ^k	The step sizes at iteration $k \in K$
ε	The consensus error used for termination criterion
$\mathscr{p}_i^{t,od}$	The auxiliary parameter for vehicles in cell $i \in C$ at time step $t \in T$ with OD pair $(o, d) \in C_{OD}$
$\mathscr{q}_{ij}^{t,od}$	The auxiliary parameter for vehicles going from cell $i \in C$ to downstream cell $j \in S(i)$ at time step $t \in T$ with OD pair $(o, d) \in C_{OD}$
$\mathbb{A}\backslash\mathbb{B}$	All elements in set $\mathbb{A}$ except for the ones in set $\mathbb{B}$

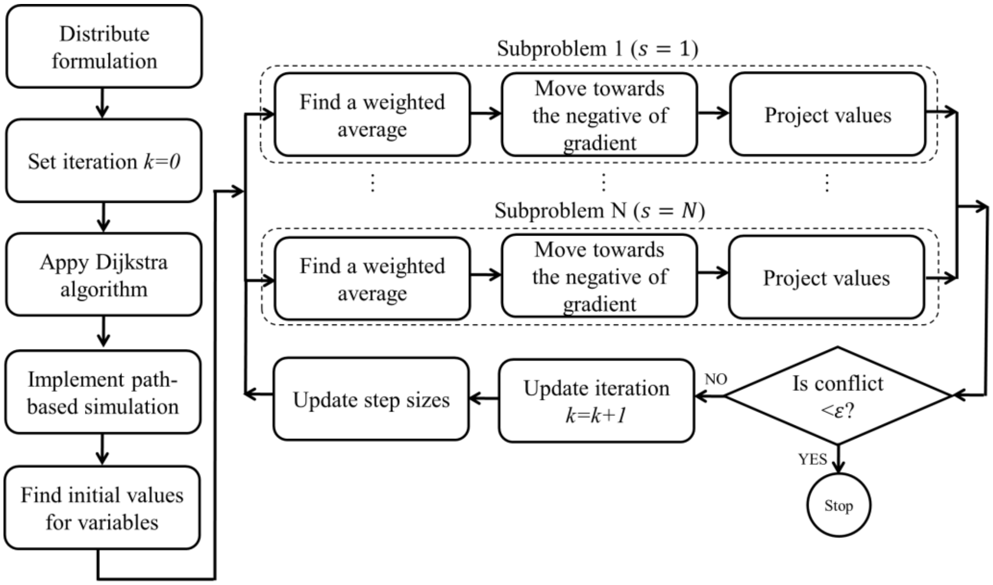


Figure 5-1 The framework of the distributed gradient-based methodology © 2022 IEEE

5.1. The Distribution of SODTA Formulation

The first step of the methodology is to decompose the network-level SODTA problem formulation into intersection-level sub-problems. Each sub-problem will have decision variables associated with one intersection. The summation of the objective functions from each sub-problem is equivalent to the objective function of the original SODTA problem, and the union of the constraints from each sub-problem is equivalent to the original constraint set. Note that the sub-problems do not share any constraints. This intersection-level decomposition is well-suited for solving problems in urban street networks since assigning more intersections to the network will not change the architecture of the methodology and its computational complexity. A general formulation for sub-problem $s \in N$ can be written by redefining all previous sets for each sub-problem $s \in N$. A general formulation for sub-problem $s \in N$ is shown below. This formulation differs from the original formulation in the set of cells for which the constraints are defined.

$$\min \sum_{(o,d)\in C_{OD}} \sum_{t\in T} \sum_{i\in Ces^s} x_i^{t,od}$$

$$\sum_{k\in P(i)} y_{ki}^{t,od} - \sum_{j\in S(i)} y_{ij}^{t,od} = x_i^{t+1,od} - x_i^{t,od} \qquad \forall t \in T, i \in Cexos^s, (o,d) \in C_{OD} \tag{5-1}$$

$$\mathcal{D}_i^{t,od} - \sum_{j\in S(i)} y_{ij}^{t,od} = x_i^{t+1,od} - x_i^{t,od} \qquad \forall t \in T, i \in Co^s, (o,d) \in C_{OD} \tag{5-2}$$

$$\sum_{i\in P(j)} y_{ij}^{t,od} = x_j^{t+1,od} - x_j^{t,od} \qquad \forall t \in T, j \in Cs^s, (o,d) \in C_{OD} \tag{5-3}$$

$$\sum_{j\in S(i)} y_{ij}^{t,od} \leq x_i^{t,od} \qquad \forall t \in T, i \in Ces^s, (o,d) \in C_{OD} \tag{5-4}$$

$$\sum_{(o,d)\in C_{OD}} \sum_{j\in S(i)} y_{ij}^{t,od} \leq F_i \qquad \forall t \in T, i \in Ces^s \tag{5-5}$$

$$\sum_{(o,d)\in C_{OD}} \sum_{i\in P(j)} y_{ij}^{t,od} \leq F_j \qquad \forall t \in T, j \in Ceo^s \tag{5-6}$$

$$\sum_{(o,d)\in C_{OD}} \sum_{i\in P(j)} y_{ij}^{t,od} \leq \delta(M_j - \sum_{\forall (o,d)\in C_{OD}} x_j^{t,od}) \qquad \forall t \in T, j \in Ceo^s \tag{5-7}$$

$$\sum_{(o,d)\in C_{OD}} \sum_{j\in S(i)} y_{ij}^{t,od} \leq f_i^t \qquad \forall t \in T, i \in Ci^s \tag{5-8}$$

$$x_i^{t,od} \geq 0 \qquad \forall t \in T, i \in C^s, (o,d) \in C_{OD} \tag{5-9}$$

$$y_{ij}^{t,od} \geq 0 \qquad \forall t \in T, i \in C^s, j \in S(i), (o,d) \in C_{OD} \tag{5-10}$$

5.2. Initialization

The approach starts with initial values for all decision variables at the first iteration. The initial solutions do not have to be feasible for the original SODTA formulation. We first implement

Dijkstra’s algorithm (Ahuja et al., 1993) to generate the shortest paths for all OD pairs. Note that we assume that the network is under free-flow conditions. Then, we use the path-based CTM simulation introduced by Ukkusuri et al. (2012) to find occupancy $x_i^{t,p}$ for cell $i \in C^s$ at time step $t \in T$ with path $p \in P$ and flow $y_{ij}^{t,p}$for links between cell $i \in C^s$ and its successor cell $j \in$ S(i) at time step $t \in T$ with path $p \in P$ for all sub-problems $s \in N$. Note that we assume each OD pair is associated with one path for simplicity. This assumption is not restrictive.

5.3. Distributed Gradient Update

This procedure updates the decision variables of all sub-problems in iteration $k + 1 \in K$ using three main steps:

Step 1: Each sub-problem optimizes the values of its decision variables. Therefore, decision variables that are in common between several sub-problems will have various values. This step sets the value of these decision variables in each sub-problem equal to their weighted average. The weights in each sub-problem are determined such that they satisfy the required conditions for convergence.

Step 2: The approach moves the computed values for sub-problem $s \in N$ from *Step 1* towards the negative direction of the gradient of the objective function of the corresponding sub-problem to minimize the total travel time within the region assigned to the sub-problem.

Step 3: The approach projects the decision variable values in sub-problem $s \in N$ from *Step 2* onto the set of constraints of the sub-problem to make the values feasible for that sub-problem.

Note that the formulation is not changed for updating the flow and occupancy values. The adjustment of the flow values is handled by the gradient update procedure. We describe these steps with a mathematical representation in the rest of this section. We first introduce four definitions for the information exchange graph (Definition 1), neighbors of a sub-problem

(Definition 2), the gradient of a function (Definition 3), and the projection operator (Definition 4).

Definition 1 (Information exchange graph) The information exchange graph $G = (N, A)$ contains nodes and directional links that belong to sets N and A, respectively. Node $s \in N$ represents a sub-problem, and link $(s, \mathscr{s}) \in A$ shows the transfer of information from sub-problem $s \in N$ to sub-problem $\mathscr{s} \in N$. If there is a variable in common between sub-problems $s, \mathscr{s} \in N$, directional links $(s, \mathscr{s}) \in A$ and $(\mathscr{s}, s) \in A$ are needed. There is also a self-arc at each node i.e., $\{(s, \mathscr{s}): s = \mathscr{s}, \forall s, \mathscr{s} \in N\}$. This arc represents the use of information generated by sub-problem $s \in N$ for computations of sub-problem $s \in N$. Weight value $\mathscr{w}_{s\mathscr{s}}$ for the link going from node $s \in N$ to node $\mathscr{s} \in N$ is also assigned to link $(s, \mathscr{s}) \in A$ to be used for incorporating information from sub-problem $s \in N$ for decision variables in sub-problem $\mathscr{s} \in N$. Assumption 1 determines the value of weights and is required to prove the convergence of the approach. We can use identical values for wights on all links $(s, \mathscr{s}) \in A$ if $s \neq \mathscr{s}$. Weights on self-arcs $s \in N$ satisfy equation $\mathscr{w}_{ss} = -\sum_{\mathscr{s} \in C^{\mathfrak{n}^s}: s \neq \mathscr{s}} \mathscr{w}_{\mathscr{s}s}$. Numerical examples for the weights are provided for a simple example in this section and a test network in the result section.

Definition 2 (The neighbors of a sub-problem) The neighbors of sub-problem $s \in N$ are sub-problems $\mathscr{s} \in N: \mathscr{s} \neq s$ that offer estimation for at least one decision variable of sub-problem $s \in N$. In other words, sub-problem $\mathscr{s} \in N$ is a neighbor of sub-problem $s \in N$ if and only if $\mathscr{w}_{\mathscr{s}s} > 0$. We define the set of neighbors of sub-problem $s \in N$ including itself by $\mathfrak{n}^s$.

In general, any node $s \in N$ may be connected to any other node $\mathscr{s} \in N$ in information exchange graph G even if the sub-problems (nodes) are not immediate neighbors in the original (physical) network if their corresponding sub-problems share a decision variable. This fact will not affect the distributed structure of the methodology because the approach uses the information from the previous iteration for the exchange process, not the current iteration. However, the structure of SODTA formulation and intersection-based distribution lead to the presence of links only between immediate neighbors in the information exchange graph. The reason is that the immediate neighbors share decision variables corresponding to the links between any two intersections (regions).

Definition 3 (Gradient of a function) Let $\mathcal{F}_s(\boldsymbol{x})$, $\mathcal{G}_s \in \mathbb{R}^n$, and X_s respectively denote the objective function value of sub-problem $s \in N$ given vector $\boldsymbol{x}$, the gradient of the objective function $\mathcal{F}_s(\boldsymbol{x})$, and the feasible region of sub-problem $s \in N$. The gradient $\mathcal{G}_s$ satisfies inequality (5-11) for all vectors $\boldsymbol{z}, \boldsymbol{x} \in X_s$.

$$\mathcal{F}_s(\boldsymbol{z}) + {\mathcal{G}_s}'(\boldsymbol{x} - \boldsymbol{z}) \le \mathcal{F}_s(\boldsymbol{x}) \tag{5-11}$$

Definition 4 (Projection operator) We use the projection operator $\boldsymbol{P}_X[\boldsymbol{z}]$ to find the projection of vector $\boldsymbol{z}$ onto a closed convex set X using Euclidean norm as shown in (5-12).

$$\boldsymbol{P}_X[\boldsymbol{z}] = argmin_{x \in X} \|\boldsymbol{z} - \boldsymbol{x}\| \tag{5-12}$$

A projection is a linear transformation, and the projection operator is used to map any vector onto a closed convex set. By solving $\arg\min\limits_{z \in X} \|z - x\|$, we can apply this operator to map vector of $x \in X$ on set X.

We now describe all three steps for iteration $k + 1 \in K$ assuming that the value of occupancy and flow decision variables are available (either from the initialization step or previous iteration $k \in K$ for all sub-problems $s \in N$). We first update the value of decision variables as described in *Steps 1* and *2*. We define auxiliary parameters $\mathscr{p}_i^{t,od}: t \in T, i \in C^s, (o,d) \in C_{OD}$ and $\mathscr{q}_{ij}^{t,od}: t \in T, i \in C^s, j \in S(i), (o,d) \in C_{OD}$ for updating the value of cell occupancy and flow decision variables, respectively. If a decision variable appears in only one sub-problem, we find its auxiliary parameter using either equation (5-13) or (5-14). Equations (5-13) and (5-14) find the auxiliary parameters for occupancy and flow variables, respectively. Including a weighted average in these equations is not required because the decision variable is optimized exclusively. We only need to move the value of the decision variable from the initialization step or previous iteration $k \in K$ at sub-problem $s \in N$ towards the negative direction of the gradient of the objective function. We denote the gradient of the objective function of sub-problem $s \in N$ by $\boldsymbol{\mathcal{G}}_s$ and the step size by γ^{k+1}. The gradient of objective function of sub-problem $s \in N$ respect to $x_i^{t,od}$ and $y_{ij}^{t,od}$ are shown by $\mathcal{G}_{s,i}^{t,od}$ and $\mathcal{G}_{s,ij}^{t,od}$, respectively.

We use equation (5-15) for the occupancy variable and (5-16) for the flow variable to find their auxiliary parameters when a decision variable appears in more than one sub-problem. For instance, the weighted average for decision variable $x_i^{t,od}: t \in T, i \in C^s, (o,d) \in C_{OD}$ at sub-problem $s \in N$ is $\sum_{s,\mathscr{s} \in n^s, j \in C^{n^s}: i=j} \mathscr{w}_{s\mathscr{s}} x_j^{t,od}$, which takes weighted average of the generated values for this variable by itself and its neighbors from previous iteration $k \in K$. Then, the value of $x_i^{t,od}$ generated by sub-problem $s \in N$ at iteration $k \in K$ is added to term $\alpha^{k+1} \sum_{s,\mathscr{s} \in n^s, j \in C^{n^s}: i=j} \mathscr{w}_{s\mathscr{s}} x_j^{t,od} - \gamma^{k+1} \mathcal{G}_s$ to find auxiliary parameter $\mathscr{p}_i^{t,od}$. We also use α^{k+1} to denote the step size used for consensus among decision variables at iteration $k+1 \in K$. The same approach is used to find other auxiliary parameters at iteration $k+1 \in K$ as shown in equation (5-16).

If $x_i^{t,od}: t \in T, i \in C^s, (o,d) \in C_{OD}$ is in one sub-problem $s \in N$

$$\mathscr{p}_i^{t,od} = x_i^{t,od} - \gamma^{k+1} \mathcal{G}_{s,i}^{t,od} \tag{5-13}$$

If $y_{ij}^{t,od}: t \in T, i \in C^s, j \in S(i), (o,d) \in C_{OD}$ is in one sub-problem $s \in N$

$$\mathscr{q}_{ij}^{t,od} = y_{ij}^{t,od} - \gamma^{k+1} \mathcal{G}_{s,ij}^{t,od} \tag{5-14}$$

If $x_i^{t,od}: t \in T, i \in C^s, (o,d) \in C_{OD}$ is in

$$\mathscr{p}_i^{t,od} = x_i^{t,od} + \tag{5-15}$$

more than one sub-problem $s \in N$	$\alpha^{k+1} \sum_{s,\mathscr{s} \in n^s, j \in C^{n^s}: i=j} \mathscr{w}_{s\mathscr{s}} x_j^{t,od} -$ $\gamma^{k+1} \mathcal{G}_{s,i}^{t,od}$	
If $y_{ij}^{t,od}: t \in T, i \in C^s, j \in S(i), (o,d) \in C_{OD}$ is in more than one sub-problem $s \in N$	$\mathscr{q}_{ij}^{t,od} = y_{ij}^{t,od} +$ $\alpha^{k+1} \sum_{s,\mathscr{s} \in n^s, j \in C^{n^s}: i=j} \mathscr{w}_{s\mathscr{s}} y_{ij}^{t,od} -$ $\gamma^{k+1} \mathcal{G}_{s,ij}^{t,od}$	(5-16)

Then, we project the value of auxiliary parameters onto the constraints set of sub-problem $s \in N$ as discussed in *Step 3* and using the projection operator described in Definition 4. In fact, we find new values for decision variables at iteration $k+1 \in K$ by solving the following optimization program for sub-problem $s \in N$.

$$\min \sum_{(o,d) \in C_{OD}} \sum_{t \in T} \sum_{i \in C^s} \left\| x_i^{t,od} - \mathscr{p}_i^{t,od} \right\|^2 + \sum_{(o,d) \in C_{OD}} \sum_{t \in T} \sum_{i \in C^s, j \in S(i)} \left\| y_{ij}^{t,od} - \mathscr{q}_{ij}^{t,od} \right\|^2 \tag{5-17}$$

s.t.

Constraints (3-2)- (3-12) for sub-problem $s \in N$

We continue the procedure of updating the value of auxiliary parameters and decision variables for sub-problem $s \in N$ over iterations $k \in K$ until the termination criterion is satisfied.

Equation (5-18) shows the update procedure in vector notation. The vector of decision variables in sub-problem $s \in N$ at iteration $k+1$ is denoted by $\mathcal{X}_s^{k+1}$, and X_s is the feasible region of sub-problem $s \in N$. Let $\mathscr{w}_{s\mathscr{s}}$ and $\boldsymbol{\mathcal{G}}_s$ denote the weight for link $(s, \mathscr{s}) \in N$ in information exchange graph $G = (N, A)$ and the gradient of the objective function of sub-problem $s \in N$, respectively. Step sizes at iteration $k+1 \in K$ are α^{k+1} and γ^{k+1}.

$$\mathcal{X}_s^{k+1} = P_{X_s}[\mathcal{X}_s^k + \alpha^{k+1} \sum_{\mathscr{s} \in n^s} \mathscr{w}_{\mathscr{s}s} \mathcal{X}_{\mathscr{s}}^k - \gamma^{k+1} \boldsymbol{\mathcal{G}}_s] \tag{5-18}$$

Figure 5-2 shows a small network of four cells that is distributed to two sub-problems. We also illustrate the three steps of the update procedure in Figure 5-3 using vector notation to visualize this procedure for this simple example. Each sub-problem contains those constraints and parts of the objective function that have the decision variables corresponding to cells and links within the region assigned to that sub-problem. Note that the constraints and objective function can be distributed following a different structure as long as the explained conditions in section 5.1 are satisfied. In this simple example, the sub-problems share the decision variables corresponding to the link between cells 2 and 3. Adding more links between cells 2 and 3 will not change the information exchange graph nor the performance of the approach because this one link forces the flow decision variables to appear in both sub-problems, and these sub-problems share information using two directional arcs in the information exchange graph.

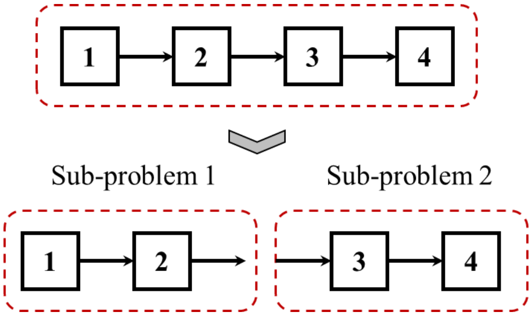
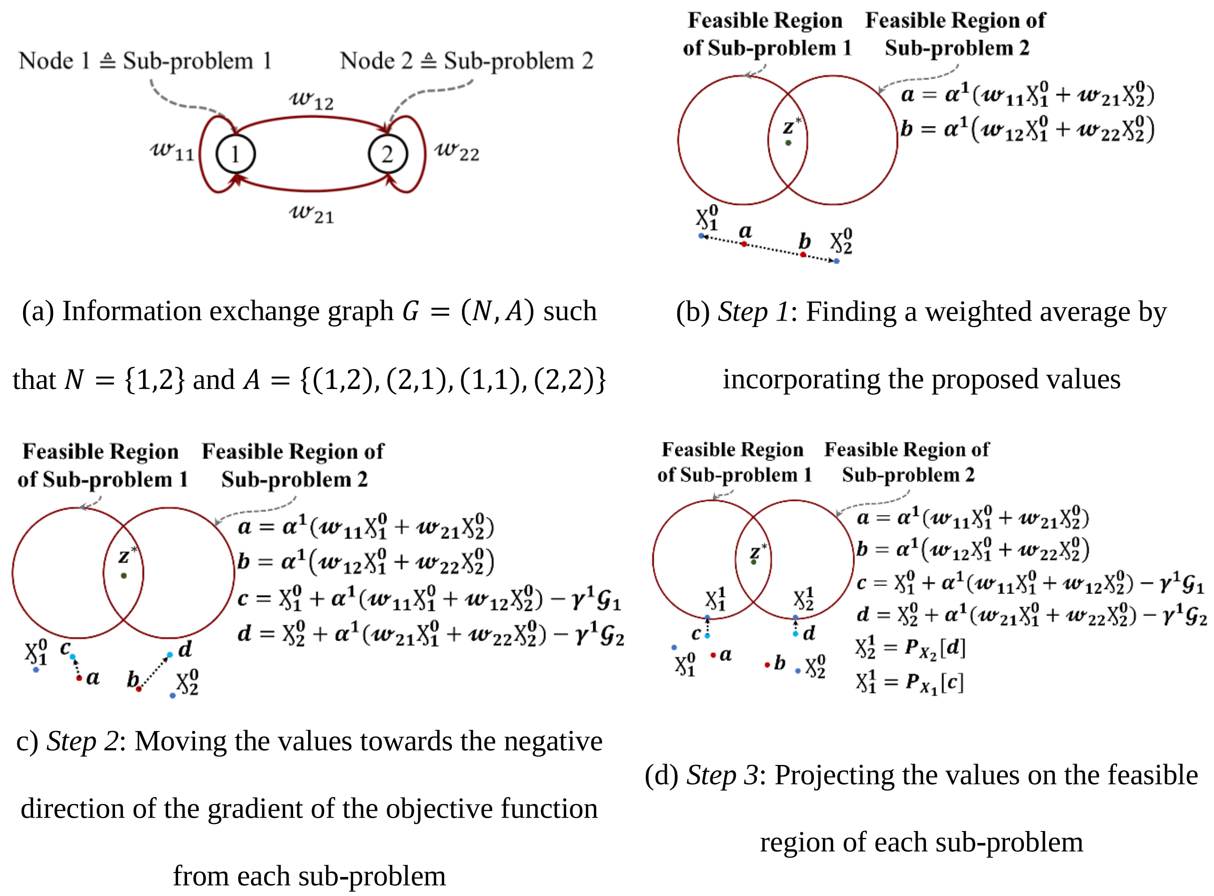


Figure 5-2 The distribution of a link with 4 cells to sub-problems 1 and 2 © 2022 IEEE

Figure 5-3.a shows information exchange graph $G = (N, A)$, where $N = \{1,2\}$ and $A = \{(1,2), (2,1), (1,1), (2,2)\}$. In Figure 5-3.b-d, each red circle represents the feasible region of one sub-problem, and vector $\boldsymbol{z}^* \in \mathrm{X}^*$ denotes the vector of optimal solutions. Vector $\boldsymbol{z}^* \in \mathrm{X}^*$ is within the feasible region of the original problem as well. The number of variables in common among sub-problems will not change this region. Note that sub-problems do not need to share all decision variables. In other words, if there is at least one variable in common, the sub-problems share information.

The vector of initial values for decision variables for sub-problems 1 and 2 at iteration $0 \in K$ are denoted by $\mathfrak{X}_1^0$ and $\mathfrak{X}_2^0$, respectively. Figure 5-3.b shows the incorporation of the value of decision variables and the computation of weighted average values for sub-problems 1 and 2 using parameters a and b, respectively, as described in *Step 1*. For example, we explain how to compute the value of a. We first multiply weight $\mathcal{w}_{11}$ on self-arc $(1,1)$ on node 1 in information exchange graph G with the vector of values for decision variables $Ӽ_1^0$ in sub-problem 1 at iteration $0 \in K$, that is $\mathcal{w}_{11}Ӽ_1^0$, and find term $\mathcal{w}_{21}Ӽ_2^0$ the same way. We then find the value of a by summing terms $\mathcal{w}_{11}Ӽ_1^0$ and $\mathcal{w}_{21}Ӽ_2^0$ and multiplying the result by step size α^1. The value of b is found following the same procedure. Figure 5-3.c presents *Step* 2 of the update procedure. We first sum the values of a and b with $Ӽ_1^0$ and $Ӽ_2^0$, respectively. Then, we add $-\gamma^1\mathcal{G}_1$ and $-\gamma^1\mathcal{G}_2$ to the value of $Ӽ_1^0 + a$ and $Ӽ_2^0 + b$ to move them towards the negative direction of the objective function gradient and find c and d, respectively. Figure 5-3.d presents the third step, where we project the value of c and d on the feasible region of sub-problems 1 and 2 to find the vector of new values for decision variables $Ӽ_1^1$ and $Ӽ_2^1$ at iteration $1 \in K$, respectively. In the next section, we explain the required assumption for determining the value of step sizes and weights.


Node 1 ≜ Sub-problem 1
Node 2 ≜ Sub-problem 2
w_{12}
w_{11}
w_{22}
w_{21}
1
2
Feasible Region of Sub-problem 1
Feasible Region of Sub-problem 2
z^*
$a = \alpha^1(w_{11}X_1^0 + w_{21}X_2^0)$
$b = \alpha^1(w_{12}X_1^0 + w_{22}X_2^0)$
$c = X_1^0 + \alpha^1(w_{11}X_1^0 + w_{12}X_2^0) - \gamma^1 \mathcal{G}_1$
$d = X_2^0 + \alpha^1(w_{21}X_1^0 + w_{22}X_2^0) - \gamma^1 \mathcal{G}_2$
$X_2^1 = P_{X_2}[d]$
$X_1^1 = P_{X_1}[c]$
X_1^0
X_2^0
X_1^1
X_2^1
a
b
c
d

(a) Information exchange graph $G = (N, A)$ such that $N = \{1,2\}$ and $A = \{(1,2), (2,1), (1,1), (2,2)\}$

(b) *Step 1*: Finding a weighted average by incorporating the proposed values

c) *Step 2*: Moving the values towards the negative direction of the gradient of the objective function from each sub-problem

(d) *Step 3*: Projecting the values on the feasible region of each sub-problem

Figure 5-3 The procedure of distributed gradient-based update for an iteration © 2022 IEEE

5.4. Convergence Properties

We prove that the solution of Distributed Gradient-based Approach (DGA) converges to the optimal solution of the SODTA formulation. We use six lemmas and two assumptions to provide the required conditions of a known convergence theorem for sequences proposed by Polyak (1987) and use the results of this theorem to show the convergence of our approach.

We first show that the feasible region of the SODTA formulation is a closed and convex set in Lemma 1. We also prove that the gradient of the objective function for all sub-problems is convex and uniformly bounded in this lemma. Then, we prove a sequence of upper bounds for various terms in Lemmas 2-5. The upper bounds help simplify the main inequality in Lemma 6 that is pertinent to the inequality in the convergence results proposed by Polyak (1987). Lemma 1 and the two assumptions are also necessary to achieve the desired results and satisfy the conditions of Polyak's (1987) theorem. Assumption 1 determines rules for the weight values in the information exchange graph and the graph connectivity, and Assumption 2 determines the step size rules. We finally use the results of Polyak's (1987) theorem and the assumptions to prove the convergence to optimal solutions in Theorem 1.

Lemma 1 Given the feasible region of sub-problems $X_s \subseteq \mathbb{R}^n$, $s = 1, \ldots, N$ and the feasible region X of SODTA formulation that is equal to the intersection of the feasible region of sub-problems, i.e., $X = \bigcap_{s \in N} X_s$, the following features hold for set X and gradient $\boldsymbol{\mathcal{G}}_s$ of the objective function $\mathcal{F}_s$ of all sub-problems $s \in N$.

(1) Set X is convex.

(2) Set X is closed.

(3) Gradient $\boldsymbol{\mathcal{G}}_s$ of objective function $\mathcal{F}_s$ for all sub-problems $s \in N$ is convex over $\mathbb{R}^n$, and

(4) Gradient $\boldsymbol{\mathcal{G}}_s$ for all sub-problems $s \in N$ is uniformly bounded, i.e., $\{\|\boldsymbol{\mathcal{G}}_s\| \leq L: L > 0\}$.

Proof. We show the proof for each part of this lemma in separate sections below:

(1) Set X is convex because the SODTA formulation consists of linear constraints that create half-spaces, and half-spaces are convex as well as the combination of convex half-spaces (Rockafellar, 1970).

(2) Set X is closed because it contains all the boundary points by having equality and less-than-or-equal-to constraints (MacCluer, 2006).

(3) Since set X is closed and convex, the SODTA objective function $\mathcal{F}$ and objective function $\mathcal{F}_s$ of each sub-problem $s \in N$ are convex and continuous as well as their gradients.

(4) Gradient $\boldsymbol{\mathcal{G}}_s$ of objective function $\mathcal{F}_s$ at sub-problem $s \in N$ is one if the derivative is

respect to $x_i^{t,od}$ for Ces^s, $t \in T$ and $(o,d) \in C_{OD}$, and all other elements of gradient $\mathcal{G}_s$ are zero, knowing that the objective function $\mathcal{F}_s$ at sub-problem $s \in N$ is the minimization of $\sum_{(o,d)\in C_{OD}} \sum_{t\in T} \sum_{i\in\, Ces^s} x_i^{t,od}$. Therefore, gradient vector $\boldsymbol{\mathcal{G}}_s$ of all sub-problems $s \in N$ is uniformly bounded by 1. ■

Weierstrass theorem states that assuming function $\mathcal{F}$ is a continuous real function on a compact metric space X, it can be concluded that there are points $r \in X$ and $l \in X$ such that $\sup_{r\in X} \mathcal{F}(r) = \mathcal{F}(r)$ and $\inf_{l\in X} \mathcal{F}(l) = \mathcal{F}(l)$ (theorem 4.16 form Rudin (1976)). Weierstrass theorem concludes that there exists $\mathcal{F}(l) \leq \mathcal{F}(x) \leq \mathcal{F}(r)$ for $\forall x \in X$. In our case, the set X is bounded, and SODTA objective function $\mathcal{F}$ is convex and as a result continuous over $\mathbb{R}^n$. Therefore, we can conclude that optimal objective function $\mathcal{F}^*$ is finite, and the minimum or optimal value can be achieved at some point in the set. The other conclusion of this statement is having a non-empty optimal set X^*. Moreover, since set $X \subseteq \mathbb{R}^n$ and the objective function $\mathcal{F}_s$ at sub-problem $s \in N$ is convex over $\mathbb{R}^n$, the gradient $\mathcal{G}_s$ of $\mathcal{F}_s$ can be derived at any point in set X.

Lemma 2 Set $X \subseteq \mathbb{R}^n$ is nonempty, closed, and convex according to Lemma 1. Therefore, we can conclude that inequality (5-19) holds for all $x \in \mathbb{R}^n$ and $z \in X$.

$$\|P_X[x] - z\|^2 \leq \|x - z\|^2 - \|P_X[x] - x\|^2 \tag{5-19}$$

Proof. the proof of this lemma exactly follows the proof that is presented in Nedic et al. (2010) - Lemma1.b**.** ■

Definition 5 (Error vector) We define $\boldsymbol{v}_s^{k+1} = \mathfrak{X}_s^k + \alpha^{k+1} \sum_{\mathcal{s}\in \mathfrak{n}^s} \mathcal{w}_{\mathcal{s}s} \mathfrak{X}_{\mathcal{s}}^k$. Then, the error vector is the difference between the projection $\boldsymbol{P}_{X_s}$ of $\boldsymbol{v}_s^{k+1} - \gamma^{k+1}\boldsymbol{\mathcal{G}}_s$ onto set X_s and $\boldsymbol{v}_s^{k+1} - \gamma^{k+1}\boldsymbol{\mathcal{G}}_s$ for sub-problem $s \in N$ and iteration $k+1 \in K$ as shown in (5-20).

$$\boldsymbol{e}_s^{k+1} = P_{X_s}[\boldsymbol{v}_s^{k+1} - \gamma^{k+1}\boldsymbol{\mathcal{G}}_s] - \boldsymbol{v}_s^{k+1} + \gamma^{k+1}\boldsymbol{\mathcal{G}}_s \tag{5-20}$$

Lemma 3 Inequality (5-21) holds for all sub-problems $s \in N$, solution vectors $\boldsymbol{x} \in X$, and iterations $k \in K$.

$$\|\text{Ӽ}_s^{k+1} - \boldsymbol{x}\|^2 \leq \|\boldsymbol{v}_s^{k+1} - \gamma^{k+1}\boldsymbol{\mathcal{G}}_s - \boldsymbol{x}\|^2 - \|\boldsymbol{e}_s^{k+1}\|^2 \tag{5-21}$$

Proof. We first use Definition 5 to substitute vector $\text{Ӽ}_s^{k+1}$with $\boldsymbol{P}_{X_s}[\boldsymbol{v}_s^{k+1} - \gamma^{k+1}\boldsymbol{\mathcal{G}}_s]$ as it is shown in (5-22).

$$\|\text{Ӽ}_s^{k+1} - \boldsymbol{x}\|^2 = \left\|P_{X_s}[\boldsymbol{v}_s^{k+1} - \gamma^{k+1}\boldsymbol{\mathcal{G}}_s] - \boldsymbol{x}\right\|^2 \tag{5-22}$$

Then, we use Lemma 2 and Definition 5 for the error vector to find an upper bound for error vector to find an upper bound for $\left\|P_{X_s}[\boldsymbol{v}_s^{k+1} - \gamma^{k+1}\boldsymbol{\mathcal{G}}_s] - \boldsymbol{x}\right\|^2$ as shown in (5-23).

$$\left\|P_{X_s}[\boldsymbol{v}_s^{k+1} - \gamma^{k+1}\boldsymbol{\mathcal{G}}_s] - \boldsymbol{x}\right\|^2 \leq \|\boldsymbol{v}_s^{k+1} - \gamma^{k+1}\boldsymbol{\mathcal{G}}_s - \boldsymbol{x}\|^2 - \|\boldsymbol{e}_s^{k+1}\|^2 \tag{5-23}$$

If we substitute $\|\text{Ӽ}_s^{k+1} - \boldsymbol{x}\|^2$with $\left\|P_{X_s}[\boldsymbol{v}_s^{k+1} - \gamma^{k+1}\boldsymbol{\mathcal{G}}_s] - \boldsymbol{x}\right\|^2$in (5-23), we find the desired result. ■

Definition 6 (Joint state vectors) we define joint state vectors similar to the definitions determined by Srivastava et al. (2010) to show the update procedure and the rest of the proofs for all sub-problems in a compact form. The column vectors $\mathbf{x}^{k+1} = (\text{Ӽ}_1^{k+1\prime}, \text{Ӽ}_2^{k+1\prime}, \ldots, \text{Ӽ}_N^{k+1\prime})'$, $\mathbf{e}^{k+1} = (\mathbf{e}_1^{k+1\prime}, \mathbf{e}_2^{k+1\prime}, \ldots, \mathbf{e}_N^{k+1\prime})'$,$\mathbf{v}^{k+1} = (\boldsymbol{v}_1^{k+1\prime}, \boldsymbol{v}_2^{k+1\prime}, \ldots, \boldsymbol{v}_N^{k+1\prime})'$, $\boldsymbol{z}^* = (\boldsymbol{z}^*{}_1{}', \boldsymbol{z}^*{}_2{}', \ldots, \boldsymbol{z}^*{}_N{}')'$, and $\boldsymbol{\mathcal{G}} = (\boldsymbol{\mathcal{G}}_1{}', \boldsymbol{\mathcal{G}}_2{}', \ldots, \boldsymbol{\mathcal{G}}_N{}')'$ represent joint vectors of Ӽ_s^{k+1}, $\mathbf{e}_s^{k+1}$, $\boldsymbol{v}_s^{k+1}$, $\boldsymbol{z}_s^*$, and $\boldsymbol{\mathcal{G}}_s$ for all sub-problems $s \in N$, respectively. Let us also define $\boldsymbol{\mathscr{w}} = \mathscr{w} \otimes I_N$ in which $N \times N$ dimensional matrix $\mathscr{w}$ contains weight $\mathscr{w}_{s\mathscr{s}}$ on all links $(s, \mathscr{s}) \in N$ as entries and matrix I_N denotes $N \times N$ dimensional identity matrix.

Let V denote the total number of variables. We now redefine update equation (5-18) as equation (5-24) using Definition 6.

$$\mathbf{x}^{k+1} = [I_{NV} + \alpha^{k+1}\boldsymbol{\mathscr{w}}]\mathbf{x}^k - \gamma^{k+1}\boldsymbol{\mathcal{G}} + \mathbf{e}^{k+1} \tag{5-24}$$

Assumption 1 Let us use θ and θ' to show a lower and an upper bound for weights $\mathscr{w}_{s\mathscr{s}}$ on all links $(s, \mathscr{s}) \in N$, respectively. The information exchange graph $G = (N, A)$ and weight $\mathscr{w}_{s\mathscr{s}}$ on link $(s, \mathscr{s}) \in N$ should satisfy the following conditions:

(1) Information exchange graph $G = (N, A)$ should be connected.

(2) If weight $\mathscr{w}_{s\mathscr{s}} > 0$ for link $(s, \mathscr{s}) \in N$, weight $\mathscr{w}_{\mathscr{s}s} > 0$ on link $(\mathscr{s}, s) \in N$ for all $s, \mathscr{s} \in N$

(3) Weights satisfy $\mathscr{w}_{s\mathscr{s}} = \mathscr{w}_{\mathscr{s}s}$ for all $s, \mathscr{s} \in N$

(4) If weight $\mathscr{w}_{s\mathscr{s}} \neq 0$, we have $\theta \leq \mathscr{w}_{s\mathscr{s}} \leq \theta'$ for all $s, \mathscr{s} \in N$ such that $\theta, \theta' > 0$

(5) Weights on self-arcs $s \in N$ satisfy equation $\mathscr{w}_{ss} = -\sum_{\mathscr{s} \in C^{\mathfrak{n}^{s}} : s \neq \mathscr{s}} \mathscr{w}_{\mathscr{s}s}$

Assumption 1 part 1 implies that there exists at least one path between each pair of sub-problems. This assumption ensures that the information is propagated from sub-problem $s \in N$ to sub-problem $\mathscr{s} \in N$ through an immediate link $(s, \mathscr{s})$ or a path $\mathcal{P}$ denoted by $\{(s, q), (q, l), \ldots, (h, \mathscr{s}) : s, q, l \ldots, h, \mathscr{s} \in N$ and $\mathscr{w}_{sq}, \mathscr{w}_{ql}, \ldots, \mathscr{w}_{h\mathscr{s}} > 0\}$.

Assumption 1 part 1 and Assumption 1 part 2 ensure the strong connectivity of the network. Assumption 1 part 2 also ensures mutual connectivity with a symmetric pattern that is required to prove Lemma 4. Assumption 1 part 3 states that non-negative weights $\mathscr{w}_{\mathscr{s}s}$ on links $(\mathscr{s}, s) \in N$ are bounded which is important to prove Lemma 6, and the received information at each sub-problem has a meaningful effect on finding the decision variables of sub-problems $s \in N$. Finally, Assumption 1 part 4 leads to having a row sum zero in the weighted graph Laplacian matrix $\mathscr{w}$. This characteristic is also important to prove Lemma 4. Note that these satisfying assumptions do not have any relation with the algorithm performance. In other words, the value of weights will not affect the convergence rate. They are provided to guarantee conditions to reach an agreement among subproblems and optimality.

Lemma 4 If parts 1 and 2 of Lemma 1, and 5 of Assumption 1 hold, inequality (5-25) will be satisfied.

$$\|\boldsymbol{v}^{k+1}-\boldsymbol{z}\|^2 \leq [(1+(\alpha^{k+1})^2\tau^2)+2(\gamma^{k+1})^2(1+(\alpha^{k+1})^2\tau^2)]\|\mathbf{x}^k-\boldsymbol{z}\|^2 - (2\theta\alpha^{k+1}+4\theta\alpha^{k+1}(\gamma^{k+1})^2)\sum_{ST^k}\|\text{Ꭓ}_s^k-\text{Ꭓ}_{\mathscr{s}}^k\|^2+(\gamma^{k+1})^2 \qquad \text{(5-25)}$$

Proof. We obtain inequality (5-26) by using joint state representation and substituting vector $\boldsymbol{z}$ with vector $\boldsymbol{x}$ in inequality (5-21) derived in Lemma 3.

$$\|\mathbf{x}^{k+1}-\boldsymbol{z}\|^2 \leq \|\boldsymbol{v}^{k+1}-\gamma^{k+1}\boldsymbol{\mathcal{G}}-\boldsymbol{z}\|^2-\|\mathbf{e}^{k+1}\|^2 \qquad \text{(5-26)}$$

We can write $\boldsymbol{v}^{k+1}=[I_{NV}+\alpha^{k+1}\boldsymbol{\mathscr{w}}]\mathbf{x}^k$ by applying joint vectors. Then, we find an equivalent expression for $\|\boldsymbol{v}^{k+1}-\gamma^{k+1}\boldsymbol{\mathcal{G}}-\boldsymbol{z}\|^2$ by substituting $[I_{NV}+\alpha^{k+1}\boldsymbol{\mathscr{w}}]\mathbf{x}^k$ with $\boldsymbol{v}^{k+1}$ in (5-26) and some mathematical simplifications, as shown in (5-27).

$$\|\boldsymbol{v}^{k+1}-\gamma^{k+1}\boldsymbol{\mathcal{G}}-\boldsymbol{z}\|^2=\|[I_{NV}+\alpha^{k+1}\boldsymbol{\mathscr{w}}]\mathbf{x}^k-\gamma^{k+1}\boldsymbol{\mathcal{G}}-\boldsymbol{z}\|^2=\|[I_{NV}+\alpha^{k+1}\boldsymbol{\mathscr{w}}]\mathbf{x}^k-\boldsymbol{z}\|^2+\|-\gamma^{k+1}\boldsymbol{\mathcal{G}}\|^2+2([I_{NV}+\alpha^{k+1}\boldsymbol{\mathscr{w}}]\mathbf{x}^k-\boldsymbol{z})'(-\gamma^{k+1}\boldsymbol{\mathcal{G}}) \qquad \text{(5-27)}$$

We derive inequality (5-28) by substituting $([I_{NV}+\alpha^{k+1}\boldsymbol{\mathscr{w}}^{k+1}]\mathbf{x}^k-\boldsymbol{z})'(-\gamma^{k+1}\boldsymbol{\mathcal{G}})$ with their individual second norms to the power of two.

$$\|\boldsymbol{v}^{k+1}-\gamma^{k+1}\boldsymbol{\mathcal{G}}-\boldsymbol{z}\|^2 \leq \|[I_{NV}+\alpha^{k+1}\boldsymbol{\mathscr{w}}]\mathbf{x}^k-\boldsymbol{z}\|^2+\|-\gamma^{k+1}\boldsymbol{\mathcal{G}}\|^2+2\|[I_{NV}+\alpha^{k+1}\boldsymbol{\mathscr{w}}]\mathbf{x}^k-\boldsymbol{z}\|^2\|-\gamma^{k+1}\boldsymbol{\mathcal{G}}\|^2 \qquad \text{(5-28)}$$

We next use an upper bound for $\|[I_{NV}+\alpha^{k+1}\boldsymbol{\mathscr{w}}]\mathbf{x}^k-\boldsymbol{z}\|^2$ derived by Srivastava et al. (2010), as shown in (5-29), where τ^2 denotes the maximum value in matrix $\mathscr{w}$ to the power of two. Let ST^k denote the spanning tree derived based on information exchange graph $G=(N,A)$ at iteration $k\in K$. There is an arc between nodes $s,\mathscr{s}\in N$ in this spanning tree if $\mathscr{w}_{s\mathscr{s}}>0$. Spanning tree ST^k always exits since having a non-empty feasible region for the SODTA problem ensures that information exchange graph $G=(N,A)$ is connected. Assumption 1 part 1, 2, and 3 are also critical for finding this upper bound.

$$\|[I_{ND}+\alpha^{k+1}\boldsymbol{\mathscr{w}}]\mathbf{x}^k-\boldsymbol{z}\|^2 \leq (1+(\alpha^{k+1})^2\tau^2)\|\mathbf{x}^k-\boldsymbol{z}\|^2-2\theta\alpha^{k+1}\sum_{ST^k}\|\text{Ꭓ}_s^k- \qquad \text{(5-29)}$$

$\chi_{\mathscr{s}}^{k}||^2$

We have bounded gradient $\mathcal{G}_s$ for the objective function at sub-problem $s \in N$ from Lemma 1 part 4, i.e., $||\boldsymbol{\mathcal{G}}|| \leq \boldsymbol{L} = \mathbf{1}$. We find the desired inequality by substituting the upper bound for $||[I_{ND} + \alpha^{k+1}\boldsymbol{\mathscr{w}}]\mathbf{x}^k - \boldsymbol{z}||^2$ from inequality (5-29) and the upper bound $L = 1$ for gradient joint vector $\boldsymbol{\mathcal{G}}$ in inequality (5-28). ■

Lemma 5 If parts 1 and 2 of Lemma 1 hold for any vector $\chi_s^k \in X_s$ and vector $\chi_{\mathscr{s}}^k \in X_{\mathscr{s}}$, inequality (5-30) is valid for all sub-problems $s, \mathscr{s} \in N$. Let B denote a uniform upper bound on the norms of the vectors in set X_s for all sub-problems $s \in N$, and parameter σ shows the radius given in Lemma 1 part 1.

$$||\chi_s^k - P_X[\chi_s^k]|| \leq \frac{B}{\sigma}\sum_{\mathscr{s}\in N}||\chi_s^k - \chi_{\mathscr{s}}^k|| \tag{5-30}$$

Proof. The proof of this lemma exactly follows the proof that is presented by Srivastava et al. (2010) -Lemma 4, see also the study by Gubin, Polyak, and Raik (1967) for the techniques used to prove this lemma. ■

Lemma 6 Inequality (5-31) holds for any vector $\boldsymbol{z}^* \in \mathrm{X}^*$ using Assumption 1 and Lemma 1. Note that X^* denotes the set of optimal solutions, and $\boldsymbol{z}^*$ is a vector of optimal values for decision variables.

$$\begin{aligned} ||\mathbf{x}^{k+1} - \boldsymbol{z}^*||^2 \leq {} & [(1 + (\alpha^{k+1})^2\tau^2) + 2(\gamma^{k+1})^2(1 + (\alpha^{k+1})^2\tau^2)]||\mathbf{x}^k - \boldsymbol{z}||^2 - \\ & (\theta\alpha^{k+1} + 4\theta\alpha^{k+1}(\gamma^{k+1})^2)\sum_{ST^k}||\chi_s^k - \chi_{\mathscr{s}}^k||^2 + (\gamma^{k+1})^2(1 + N) + \\ & \frac{(\gamma^{k+1})^2((N-1)\frac{NB+\sigma}{\sigma})^2}{\alpha^{k+1}\theta} + 2\gamma^{k+1}\alpha^{k+1}\sum_{s\in N}||\sum_{\mathscr{s}\in N}\mathscr{w}_{\mathscr{s}s}\chi_{\mathscr{s}}^k|| - ||\mathbf{e}^{k+1}||^2 - \\ & 2\gamma^{k+1}\left(\mathcal{F}(\boldsymbol{\rho}^k) - \mathcal{F}(\boldsymbol{z}^*)\right) \end{aligned} \tag{5-31}$$

Proof. We use similar techniques by Srivastava et al. (2010). We derived $||\chi_s^{k+1} - \boldsymbol{x}||^2 \leq ||\boldsymbol{v}_s^{k+1} - \gamma^{k+1}\boldsymbol{\mathcal{G}}_s - \boldsymbol{x}||^2 - ||\boldsymbol{e}_s^{k+1}||^2$ in (5-21). If we substitute $\boldsymbol{z}^*$ with x and substitute this expression $\|\boldsymbol{v}_s^{k+1} - \gamma^{k+1}\boldsymbol{\mathcal{G}}_s - \boldsymbol{z}^*\|^2$ with its multiplication $(\boldsymbol{v}_s^{k+1} - \gamma^{k+1}\boldsymbol{\mathcal{G}}_s - \boldsymbol{z}^*)\,'(\boldsymbol{v}_s^{k+1} - \gamma^{k+1}\boldsymbol{\mathcal{G}}_s - \boldsymbol{z}^*)$ in the inequality of Lemma 3, we find inequality (5-32).

$$\|\chi_s^{k+1} - \boldsymbol{z}^*\|^2 \leq \|\boldsymbol{v}_s^{k+1} - \boldsymbol{z}^*\|^2 - 2\gamma^{k+1}\boldsymbol{\mathcal{G}}_s{}'(\boldsymbol{v}_s^{k+1} - \boldsymbol{z}^*) + (\gamma^{k+1})^2\|\boldsymbol{\mathcal{G}}_s\|^2 - \|\boldsymbol{e}_s^{k+1}\|^2 \tag{5-32}$$

We can write inequality (5-33) knowing that gradient $\boldsymbol{\mathcal{G}}_s$ of the objective function $\mathcal{F}_s$ at sub-problem $s \in N$ is bounded, i.e., $\|\boldsymbol{\mathcal{G}}_s\| \leq 1$, from Lemma 1 part 4 and $\boldsymbol{\mathcal{G}}_s{}'(\boldsymbol{v}_s^{k+1} - \boldsymbol{z}^*) \geq \boldsymbol{\mathcal{F}}_s(\boldsymbol{v}_s^{k+1}) - \boldsymbol{\mathcal{F}}_s(\boldsymbol{z}^*)$. Let $\boldsymbol{\mathcal{F}}_s(\boldsymbol{x})$ denote the objective function of sub-problem $s \in N$ at vector x.

$$\|\chi_s^{k+1} - \boldsymbol{z}^*\|^2 \leq \|\boldsymbol{v}_s^{k+1} - \boldsymbol{z}^*\|^2 + (\gamma^{k+1})^2 - 2\gamma^{k+1}[\boldsymbol{\mathcal{F}}_s(\boldsymbol{v}_s^{k+1}) - \boldsymbol{\mathcal{F}}_s(\boldsymbol{z}^*)] - \|\boldsymbol{e}_s^{k+1}\|^2 \tag{5-33}$$

We sum both sides of inequality (5-33) over all sub-problems $s \in N$, as shown in inequality (5-34).

$$\sum_{s\in N}\|\chi_s^{k+1} - \boldsymbol{z}^*\|^2 \leq \sum_{s\in N}\|\boldsymbol{v}_s^{k+1} - \boldsymbol{z}^*\|^2 + N(\gamma^{k+1})^2 - 2\gamma^{k+1}\sum_{s\in N}[\boldsymbol{\mathcal{F}}_s(\boldsymbol{v}_s^{k+1}) - \mathcal{F}_s(\boldsymbol{z}^*)] - \sum_{s\in N}\|e_s^{k+1}\|^2 \tag{5-34}$$

We now use joint vectors $\mathbf{x}^{k+1} = (\chi_1^{k+1\prime}, \chi_2^{k+1\prime}, \dots, \chi_N^{k+1\prime})'$, $\mathbf{e}^{k+1} = (\mathbf{e}_1^{k+1\prime}, \mathbf{e}_2^{k+1\prime}, \dots, \mathbf{e}_N^{k+1\prime})'$, $\mathbf{v}^{k+1} = (\boldsymbol{v}_1^{k+1\prime}, \boldsymbol{v}_2^{k+1\prime}, \dots, \boldsymbol{v}_N^{k+1\prime})'$, and $\boldsymbol{z}^* = ({\boldsymbol{z}^*}_1{}', {\boldsymbol{z}^*}_2{}', \dots, {\boldsymbol{z}^*}_N{}')'$ from Definition 5 instead of using the summation operator in $\sum_{s\in N}\|\chi_s^{k+1} - \boldsymbol{z}^*\|^2$, $\sum_{s\in N}\|\boldsymbol{v}_s^{k+1} - \boldsymbol{z}^*\|^2$, and $\sum_{s\in N}\|\boldsymbol{e}_s^{k+1}\|^2$ in inequality (5-34), respectively to derive inequality (5-35).

$$\|\mathbf{x}^{k+1} - \boldsymbol{z}^*\|^2 \leq \|\mathbf{v}^{k+1} - \boldsymbol{z}^*\|^2 + N(\gamma^{k+1})^2 - 2\gamma^{k+1}\sum_{s\in N}[\mathcal{F}_s(\boldsymbol{v}_s^{k+1}) - \mathcal{F}_s(\boldsymbol{z}^*)] - \|\mathbf{e}^{k+1}\|^2 \tag{5-35}$$

We substitute the upper bound for $\|\boldsymbol{v}^{k+1} - \boldsymbol{z}\|^2$derived in Lemma 4 and use $\boldsymbol{z}^*$ instead of $\boldsymbol{z}$ to derive inequality (5-36).

$$\|\mathbf{x}^{k+1} - \boldsymbol{z}^*\|^2 \leq [(1 + (\alpha^{k+1})^2\tau^2) + 2(\gamma^{k+1})^2(1 + (\alpha^{k+1})^2\tau^2)]\|\mathbf{x}^k - \boldsymbol{z}\|^2 - \tag{5-36}$$

$$(2\theta\alpha^{k+1} + 4\theta\alpha^{k+1}(\gamma^{k+1})^2)\sum_{ST^k}\|\mathcal{X}_s^k - \mathcal{X}_{\mathscr{s}}^k\|^2 + (\gamma^{k+1})^2 + N(\gamma^{k+1})^2 -$$

$$2\gamma^{k+1}\sum_{s\in N}[\mathcal{F}_s(\boldsymbol{v}_s^{k+1}) - \mathcal{F}_s(\boldsymbol{z}^*)] - \|\mathbf{e}^{k+1}\|^2$$

We define new parameter $\boldsymbol{\rho}^k = \sum_{s\in N}\boldsymbol{P}_X[\mathcal{X}_s^k]/N$. Because projection $\boldsymbol{P}_X[\mathcal{X}_s^k]$ belongs to X for all sub-problems $s \in N$ and iterations $k \in K$, and X is convex, the convex combination of projections $\boldsymbol{P}_X[\mathcal{X}_s^k]$ for all sub-problems $s \in N$, that is $\sum_{s\in N}\boldsymbol{P}_X[\boldsymbol{x}_s^k]/N$, belongs to X as well, i.e., $\sum_{s\in N}\boldsymbol{P}_X[\boldsymbol{x}_s^k]/N \in X$. By adding and subtracting $\mathcal{F}_s(\boldsymbol{\rho}^k)$ in the right side of inequality (5-36) and using $\mathcal{F}(\boldsymbol{x}) = \sum_{s\in N}\mathcal{F}_s(\boldsymbol{x})$, we derive inequality (5-37).

$$\|\mathbf{x}^{k+1} - \boldsymbol{z}^*\|^2 \leq [(1 + (\alpha^{k+1})^2\tau^2) + 2(\gamma^{k+1})^2(1 + (\alpha^{k+1})^2\tau^2)]\|\mathbf{x}^k - \boldsymbol{z}\|^2 - \qquad \text{(5-37)}$$

$$(2\theta\alpha^{k+1} + 4\theta\alpha^{k+1}(\gamma^{k+1})^2)\sum_{ST^k}\|\mathcal{X}_s^k - \mathcal{X}_{\mathscr{s}}^k\|^2 + (\gamma^{k+1})^2 + N(\gamma^{k+1})^2 -$$

$$2\gamma^{k+1}\sum_{s\in N}[\mathcal{F}_s(\boldsymbol{v}_s^{k+1}) - \mathcal{F}_s(\boldsymbol{\rho}^k)] - \|\mathbf{e}^{k+1}\|^2 - 2\gamma^{k+1}\Big(\mathcal{F}(\boldsymbol{\rho}^k) - \mathcal{F}(\boldsymbol{z}^*)\Big)$$

Based on convexity and boundedness properties in Lemma 1, we have $|\mathcal{F}_s(v_s^{k+1}) - \mathcal{F}_s(\rho^k)| \leq \|v_s^{k+1} - \rho^k\|$, and we can derive inequality (5-38).

$$\|\mathbf{x}^{k+1} - \boldsymbol{z}^*\|^2 \leq [(1 + (\alpha^{k+1})^2\tau^2) + 2(\gamma^{k+1})^2(1 + (\alpha^{k+1})^2\tau^2)]\|\mathbf{x}^k - \boldsymbol{z}\|^2 - \qquad \text{(5-38)}$$

$$(2\theta\alpha^{k+1} + 4\theta\alpha^{k+1}(\gamma^{k+1})^2)\sum_{ST^k}\|\mathcal{X}_s^k - \mathcal{X}_{\mathscr{s}}^k\|^2 + (\gamma^{k+1})^2 + N(\gamma^{k+1})^2 +$$

$$2\gamma^{k+1}\sum_{s\in N}\|\boldsymbol{v}_s^{k+1} - \boldsymbol{\rho}^k\| - \|\mathbf{e}^{k+1}\|^2 - 2\gamma^{k+1}\Big(\mathcal{F}(\boldsymbol{\rho}^k) - \mathcal{F}(\boldsymbol{z}^*)\Big)$$

We finally use the upper bound derived by Srivastava et al. (2010) for $2\gamma^{k+1}\sum_{s\in N}\|\boldsymbol{v}_s^{k+1} - \boldsymbol{\rho}^k\|$ in inequality (5-38) to derive the desired result. The upper bound is $\frac{(\gamma^{k+1})^2((N-1)\frac{NB+\sigma}{\sigma})^2}{\alpha^{k+1}\theta} + \theta\alpha^{k+1}\sum_{ST^k}\|\mathcal{X}_s^k - \mathcal{X}_{\mathscr{s}}^k\|^2 + 2\gamma^{k+1}\alpha^{k+1}\sum_{s\in N}\|\sum_{\mathscr{s}\in N}\mathscr{w}_{\mathscr{s}s}\mathcal{X}_{\mathscr{s}}^k\|$. ■

Assumption 2 The step sizes α^k and γ^k should satisfy these conditions: (1)$\sum_{k=1}^{\infty}\alpha^k = \infty$ and $\sum_{k=1}^{\infty}\gamma^k = \infty$, (2) $\sum_{k=1}^{\infty}(\alpha^k)^2 < \infty$ and $\sum_{k=1}^{\infty}(\gamma^k)^2 < \infty$, (3) $\sum_{k=1}^{\infty}(\alpha^k)^2(\gamma^k)^2 < \infty$, (4) $\sum_{k=1}^{\infty}\frac{(\gamma^k)^2}{\alpha^k} < \infty$, and (5) $\sum_{k=1}^{\infty}\min(\alpha^k, \gamma^k) = \infty$.

Let us assume $\alpha^k = \frac{1}{k^{\mathfrak{u}}}$ and $\gamma^k = \frac{1}{k^{\mathfrak{v}}}$ as an example for step sizes. Assuming $0.5 < \mathfrak{u}, \mathfrak{v} \leq 1$ satisfies assumptions 2.1, 2.2, and 2.3. Considering $1 + \mathfrak{u} < 2\mathfrak{v}$ and $\mathfrak{u} < \mathfrak{v}$ ensures assumptions 2.4 and assumptions 2.5, respectively. Therefore, the above five conditions are sufficient to determine the step sizes, and there exist step sizes that satisfy all conditions. More details can be found in the study by Srivastava (2012).

We now prove the convergence of the distributed approach to optimality using a deterministic version of supermartingale convergence proposed by Polyak (1987) and also applied by Nedić and Olshevsky (2014). We provide proof of convergence in Theorem 1. Theorem 1 includes similar techniques proposed by Srivastava (2012).

Theorem 1 Vectors Ꭓ_s^{k+1} generated by sub-problems $s \in N$ using the proposed distributed gradient-based methodology converge to a common optimal vector $\boldsymbol{z}^* \in X^*$, i.e., $\lim_{k\longrightarrow\infty} \text{Ꭓ}_s^{k+1} = \boldsymbol{z}^*, \forall s \in N$ if Assumptions 1 and 2 hold.

Proof of Theorem 1. We start with the inequality derived in Lemma 6 and show that this inequality satisfies all the required conditions in the following Lemma from Polyak (1987) using Assumption 1 and Assumption 2.

Polyak (1987)- Lemma 11- Chapter 2.2: Let Θ^{k}, Γ^k, Ψ^k , Ω^k and Φ^k be sequences of variables that satisfy inequality (5-39) for all iterations $k \in K$. Then, if the scalar sequences Θ^{k}, Γ^k, Ψ^k , Ω^k and Φ^k are nonnegative, and these inequalities $\sum_{k=1}^{\infty} \Gamma^k < \infty$ and $\sum_{k=1}^{\infty} \Phi^k < \infty$ hold for all iterations $k \in K$, the sequence Ψ^k converges to a non-negative value, and we have $\sum_{k=1}^{\infty} \Omega^k < \infty$.

$$\Theta^{\mathrm{k+1}} \leq (1 + \Gamma^k)\Psi^k - \Omega^k + \Phi^k \qquad \text{(5-39)}$$

We now use the results of Polyak (1987)- Lemma 11. Each part in inequality (5-39) represents one sequence in Polyak (1987)- Lemma 11, and we can match inequality (5-31) with inequality

(5-39) as follows:

- $\Theta^{\mathrm{k+1}} \triangleq \|\mathbf{x}^{k+1} - \boldsymbol{z}^{*}\|^2$,
- $\Gamma^k \triangleq 2\tau^2(\alpha^{k+1})^2(\gamma^{k+1})^2 + 2(\gamma^{k+1})^2 + (\alpha^{k+1})^2\tau^2$,
- $\Psi^k \triangleq \|\mathbf{x}^k - \boldsymbol{z}\|^2$,
- $\Omega^k \triangleq (\theta\alpha^{k+1} + 4\theta\alpha^{k+1}(\gamma^{k+1})^2)\sum_{ST^k}\|\mathfrak{X}_s^k - \mathfrak{X}_{\mathscr{s}}^k\|^2 + 2\gamma^{k+1}\Big(\mathcal{F}(\boldsymbol{\rho}^k) - \mathcal{F}(\boldsymbol{z}^*)\Big)$, and
- $\Phi^k \triangleq (\gamma^{k+1})^2(1+N) + \frac{(\gamma^{k+1})^2((N-1)\frac{NB+\sigma}{\sigma})^2}{\alpha^{k+1}\theta} + 2\gamma^{k+1}\alpha^{k+1}\sum_{s\in N}\|\sum_{\mathscr{s}\in N}\mathscr{w}_{\mathscr{s}s}\mathfrak{X}_{\mathscr{s}}^k\|$.

It is evident that the sequences Θ^{k}, Γ^k, Ψ^k , Ω^k and Φ^k are nonnegative and satisfy the first condition in Polyak (1987)- Lemma 11. We then show that $\sum_{k=1}^{\infty}\Gamma^k \triangleq \sum_{k=1}^{\infty} 2\tau^2(\alpha^{k+1})^2(\gamma^{k+1})^2 + 2(\gamma^{k+1})^2 + (\alpha^{k+1})^2\tau^2 < \infty$ and $\sum_{k=1}^{\infty}\Phi^k \triangleq (\gamma^{k+1})^2 + N(\gamma^{k+1})^2 + \frac{(\gamma^{k+1})^2((N-1)\frac{NB+\sigma}{\sigma})^2}{\alpha^{k+1}\theta} + 2\gamma^{k+1}\alpha^{k+1}\sum_{s\in N}\|\sum_{\mathscr{s}\in N}\mathscr{w}_{s\mathscr{s}}\mathfrak{X}_{\mathscr{s}}^k\| < \infty$ to ensure that the second condition in Polyak (1987)- Lemma 11 holds.

We know $\sum_{k=1}^{\infty}(\alpha^k)^2 < \infty$, $\sum_{k=1}^{\infty}(\gamma^k)^2 < \infty$, and $\sum_{k=1}^{\infty}(\alpha^k)^2(\gamma^k)^2 < \infty$ from Assumption 2, and τ^2 is a constant value. Therefore, inequality $\sum_{k=1}^{\infty}\Gamma^k < \infty$ holds. We then show that $\sum_{k=1}^{\infty}\Phi^k < \infty$. We have $\sum_{k=1}^{\infty}(\gamma^k)^2 < \infty$ and $\sum_{k=1}^{\infty}\frac{(\gamma^k)^2}{\alpha^k} < \infty$ from Assumption 2. Because all sets $X_s \subseteq \mathbb{R}^n$, $s = 1, \ldots, N$ are closed according to Lemma 1 part 2, and the weights are bounded using Assumption 1, $\sum_{s\in N}\|\sum_{\mathscr{s}\in N}\mathscr{w}_{s\mathscr{s}}\mathfrak{X}_{\mathscr{s}}^k\|$ is bounded. We also know that $\sum_{k=1}^{\infty}\alpha^k\gamma^k < \infty$ by Assumption 2. Hence, we can also conclude that $\sum_{k=1}^{\infty}\Phi^k < \infty$. Now, we can use the results of Polyak (1987)- Lemma 11 and infer that $\Omega^k < \infty$ and sequence $\|\mathbf{x}^k - \boldsymbol{z}^*\|^2$ converges to $\boldsymbol{z}^* \in X^*$. We first focus on sequence Ω^k. So, we can write inequalities in (5-40).

$$\sum_{k=1}^{\infty}\min(\alpha^{k+1},\gamma^{k+1})\,[(\theta + 4\theta(\gamma^{k+1})^2)\sum_{ST^k}\left\|\mathfrak{X}_i^k - \mathfrak{X}_j^k\right\|^2 + 2\Big(\mathcal{F}(\rho^k) - \qquad (5\text{-}40)$$

$\mathcal{F}(z^*)\Big)] < (\theta\alpha^{k+1} + 4\theta\alpha^{k+1}(\gamma^{k+1})^2)\sum_{ST^k}\left\|\text{Ҳ}_i^k - \text{Ҳ}_j^k\right\|^2 + 2\gamma^{k+1}\Big(\mathcal{F}(\rho^k) -$

$\mathcal{F}(z^*)\Big) < \infty.$

Since $\sum_{k=1}^{\infty}\min\,(\alpha^k\,,\gamma^k) = \infty$ according to Assumption 2, we have $\lim_{k\to\infty}\sum_{ST^k}\|\text{Ҳ}_s^k - \text{Ҳ}_{\mathscr{s}}^k\|^2 = 0$ and $\lim_{k\to\infty}\mathcal{F}(\rho^k) = \mathcal{F}(z^*)$ for at least one subsequence. The number of nodes $s \in N$ and arcs $(s, \mathscr{s}) \in A{:}\, s, \mathscr{s} \in N$ of information exchange graph $G = (N, A)$ are finite, so we can derive finite number of spanning trees ST^k over iterations $k \in K$, and spanning tree ST^k repeats often over iterations. We can write $\lim_{k\to\infty}\sum_{ST^k}\|\text{Ҳ}_s^k - \text{Ҳ}_{\mathscr{s}}^k\|^2 = 0$ and $\lim_{k\to\infty}\|\text{Ҳ}_s^k - \text{Ҳ}_{\mathscr{s}}^k\|^2$=0 for any spanning tree ST^k at iteration $k \in K$ for all sub-problems $s, \mathscr{s} \in N$ using the connectivity of information exchange graph $G = (N, A)$ and having finite set of arcs and nodes.

We use inequality $\lim_{k\to\infty}\sum_{s\in N}\|\text{Ҳ}_s^k - \boldsymbol{\rho}^k\|^2 \;\leq \lim_{k\to\infty} 2(\frac{NB+\sigma}{N\sigma})\sum_{s<\mathscr{s}}\|\text{Ҳ}_s^k - \text{Ҳ}_{\mathscr{s}}^k\|^2$ in the study by Srivastava et al. (2010). Since the right-hand side of this inequality is 0, we have $\lim_{k\to\infty}\sum_{s\in N}\|\text{Ҳ}_s^k - \boldsymbol{\rho}^k\|^2 = 0$ and $\lim_{k\to\infty}\|\text{Ҳ}_s^k - \boldsymbol{\rho}^k\|^2$for all sub-problems $s \in N$. We use $\lim_{k\to\infty}\mathcal{F}(\boldsymbol{\rho}^k) = \mathcal{F}(\boldsymbol{z}^*)$ to infer the sequence $\boldsymbol{\rho}^k$ converges to optimal vector $\boldsymbol{z}^* \in X^*$. We also know $\lim_{k\to\infty}\|\text{Ҳ}_s^k - \boldsymbol{\rho}^k\|^2 = 0$. Therefore, $\boldsymbol{x}_s^k$ and $\boldsymbol{\rho}^k$ converge to the same vector that is $\boldsymbol{z}^* \in X^*$for all sub-problems $s \in N$. The sequences of Ҳ_s^k for all sub-problems $s \in N$ converge to a common point that is also optimal vector $\boldsymbol{z}^* \in X^*$ knowing that the sequence $\|\mathbf{x}^k - \boldsymbol{z}^*\|^2$ converges to $\boldsymbol{z}^* \in X^*$ using the results of Polyak (1987) - Lemma 11.■

5.5. Termination Criterion

The methodology finds the optimal solution when there is no disagreement among the values of decision variables found by different sub-problems. Note that only some of the flow variables for the boundary links between intersections belong to more than one sub-problem. Therefore, the optimal solution is found when sub-problems are in agreement on the value of these decision variables, or in other words, when the left-hand side of inequality (5-41) is zero. Since urban streets have a different number of lanes, the value of flow decision variables should be normalized to compute the disagreement by dividing the flow variable by the maximum capacity of receiving or sending cells. We set the termination criterion to reach a disagreement of at most ε. If the termination criterion is not met, we update step sizes α^{k+1} and γ^{k+1} and go back to the distributed gradient-based update step.

$$\sum_{t\in T}\sum_{od\in OD}\sum_{i\in C^{s},j\in S(i),e\in C^{\mathscr{s}\in n^{S}},f\in S(e):\, i=e,j=f,s\neq\mathscr{s}}\left| y_{ij}^{t,od}\Big/\max(M_i,M_j) - y_{ef}^{t,od}\Big/\max(M_e,M_f)\right| \leq \varepsilon \tag{5-41}$$

This termination criterion is checked using a distributed communication paradigm. Each sub-problem checks inequality (5-41), and if it satisfies the termination criterion, it stops updating the value of variables. The sub-problems compute the termination at the same time. The value of ε does not dependent on the network size since it is determined for each intersection. Finding a feasible solution is possible regardless of the value of ε. Moreover, we select the duration of the study period so that it is long enough for all vehicles can exit the network.

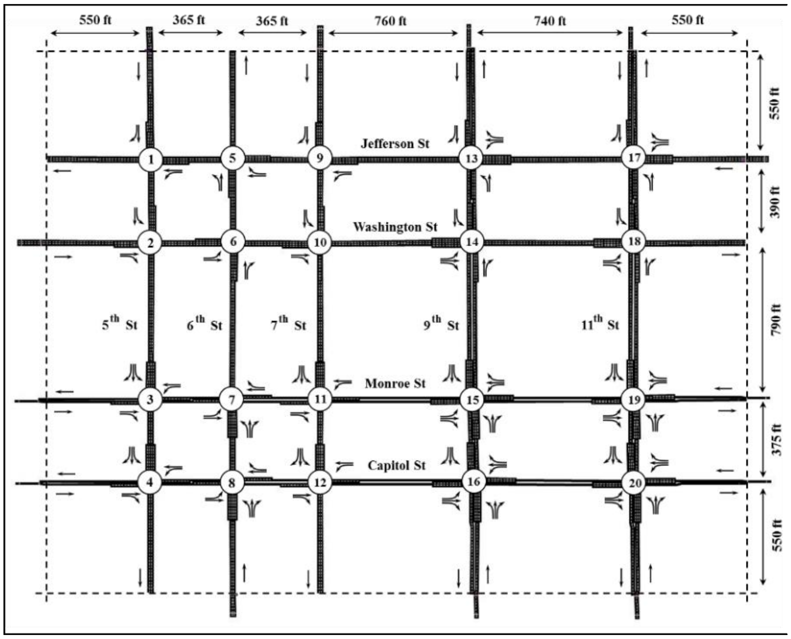
The vector of optimal solutions for the subproblems may not be a feasible solution to SODTA. Therefore, a feasible solution is found using CTM simulation. The input to the CTM is turning ratios that are found using the output of projections on the feasible region of subproblems. The output of the simulation is feasible flows for the entire network.

5.6. Test Network

We tested DGA on a portion of downtown Springfield network in Illinois. The network consists of 20 intersections with one-way and two-way streets. All intersections are signalized with predefined signal timing parameters. We considered 15 OD pairs and three demand profiles for this test network, as presented in Figure 5-4. A test network with 40 (4×10) intersections and 25 ODs is also tested to show how the approach scales. This network is created by duplicating the network of 20 intersections. Table 5-2 shows three demand profiles for this network. According to Assumption 1, one set of values for weights can be $\mathscr{w}_{s\mathscr{s}} = \mathscr{w}_{\mathscr{s}s} = 1$ for all $s, \mathscr{s} \in N$ in which $s \neq \mathscr{s}$ and $\mathscr{w}_{ss} = -\sum_{\mathscr{s} \in C^{\mathfrak{n}^s}: s \neq \mathscr{s}} \mathscr{w}_{\mathscr{s}s}$ for weights on self-arcs $s \in N$. We also set the step size rules as $\alpha^k = {}^1/_{k^{0.55}}$ and $\gamma^k = {}^1/_{k}$, that satisfy Assumption 2. For 20 intersection network, the termination is set to reach a disagreement of 0.5 for each sub-problem. It should be noted that networks with 20 and 40 intersections respectively result in 4,218,000 and 14,120,00 decision variables, representing a large optimization program.

OD / (veh/hr/ln)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Profile 1	333	133	333	333	67	333	333	333	133	133	333	333	333	333	67
Profile 2	500	200	500	500	100	500	500	500	200	200	500	500	500	500	100
Profile 3	750	300	750	750	150	750	750	750	300	300	750	750	750	750	150

The case study of 20 intersections –Springfield, IL



CTM characteristics	
The duration of each time step (sec)	6
The number of cells in each link	2,3 and 4
The total number of cells	316
The total number of links	387
Free-flow speed (mph)	25
Cell length (ft)	220
The capacity of cells except for source and sink cells (veh)	9, 24, and 36
Saturation flow rate except for source and sink cells (veh/ts/ln)	3,6, and 9
Capacity and Saturation flow rate for source and sink cells	1000

sec: second, mph: mile per hour, ft: feet, veh: vehicle, ts: time step

Figure 5-4 Case study of 20 intersections, CTM characteristics, and demand patterns © 2022 IEEE

Table 5-2 Demand profiles for the test network of 40 intersections

Network of 40 intersections with 25 ODs							
OD/Demand (veh/hr/ln)	Under-saturated	Semi-saturated	Over-saturated	OD/Demand (veh/hr/ln)	Under-saturated	Semi-saturated	Over-saturated
1	333	500	750	14	333	500	750
2	267	400	600	15	267	400	600
3	27	40	60	16	27	40	60
4	40	60	90	17	40	60	90
5	267	400	600	18	267	400	600
6	67	100	150	19	67	100	150
7	320	480	720	20	320	480	720
8	200	300	450	21	200	300	450
9	120	180	270	22	120	180	270
10	200	300	450	23	200	300	450
11	120	180	270	24	120	180	270
12	200	300	450	25	320	480	720
13	120	180	270				

5.7. Results

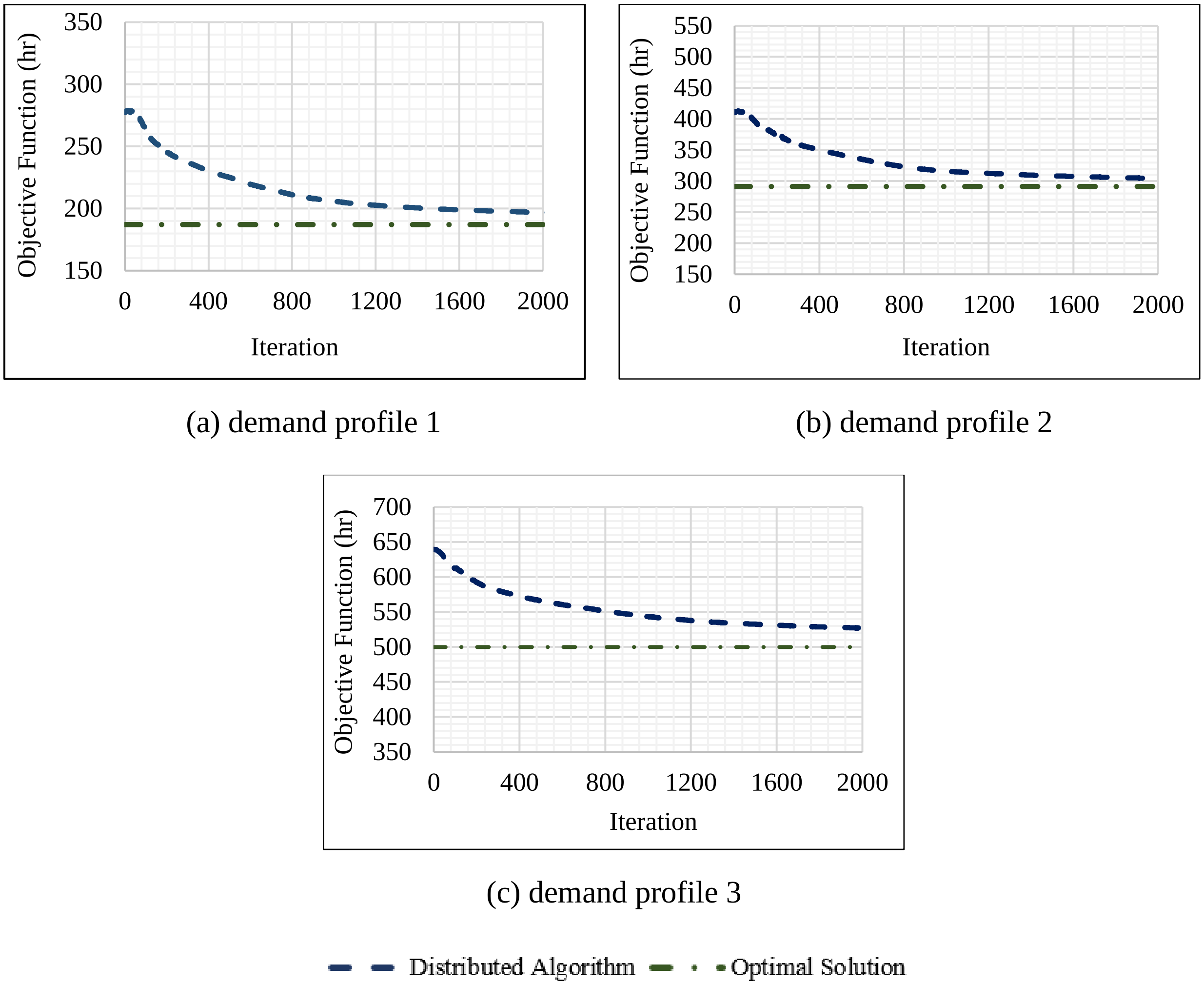
We applied DGA and a central approach to the case study network of 20 intersections. Table 5-3 shows the total travel time and the total computation time by our approach and the central approach under three different demand patterns. The SODTA formulation has more than 4 million decision variables in this case study, and we could not find the optimal solution by running CPLEX for cases with less than 150 GB of memory. DGA reduced the decision variables of each sub-problem to 243,600 on average. Each pair of sub-problems shares 6,000 variables for one-directional streets and 12,000 variables for two-directional streets. DGA required only 5 GB of memory to generate the solutions. Note that the distributed approach is implemented with a parallel architecture. It also found the solutions with at most a 5% optimality gap in less than 2.01 hours, which translates to about 97 % shorter runtime than CPLEX.

Table 5-3 Objective function and computation time for DGA and a central approach for the network of 20 intersections © 2022 IEEE

Network Performance Measures	Approach/Gap	Required Memory (GB)	Demand Profile 1	Demand Profile 2	Demand Profile 3
Objective Function: Total Travel Time (hr)	Optimal Solution (from CPLEX)	150	187.06	291.29	499.85
	Proposed Approach	5	197.34	307.30	524.84
	Difference (%)	-	5.49	5.50	5.01
Run-time (hr)	Optimal Solution (from CPLEX)	150	70.54	73.59	90.56
	Proposed Approach	5	1.60	1.81	2.01
	Difference (%)	-	97.73	97.54	97.78

Figure 5-5. a-c shows the objective value of DGA and the optimal objective value for three demand profiles over iterations. The distributed approach reduces the value of the objective function towards the optimal value over iterations. Note that the optimal solution from CPLEX had the well-known holding-back problem due to the linearization of the minimization functions

in CTM. However, the solutions from the distributed approach did not have this issue since we found the solutions by simulating the network using non-linear CTM equations as presented by Mohebifard and Hajbabaie, (2019). In all three figures, the optimality gap was at most 5% when we stopped the approach.



(a) demand profile 1

(b) demand profile 2

(c) demand profile 3

Figure 5-5 The objective functions value over iterations for network of 20 intersections © 2022 IEEE

5.7.1. Comparison with a Danzig-Wolfe Decomposition-based Algorithm (DWDA)

We compared the solutions and performance of the DGA to an OD-based decomposition approach developed by Mehrabipour et al. (2019). Table 5-4 shows different characteristics and performance measures for both approaches. Danzig-Wolfe Decomposition-based Algorithm

(DWDA) has a master problem and several sub-problems that are solved iteratively with a stopping criterion of a 5% gap between the upper bound and lower bound of the approach. Each sub-problem has all decision variables and constraints for a SODTA formulation with one OD pair. Since we have 15 ODs, the number of sub-problems is 15. New extreme points are generated by sub-problems and added to the master problem solution pool. Therefore, the complexity of the master problem increases over iterations.

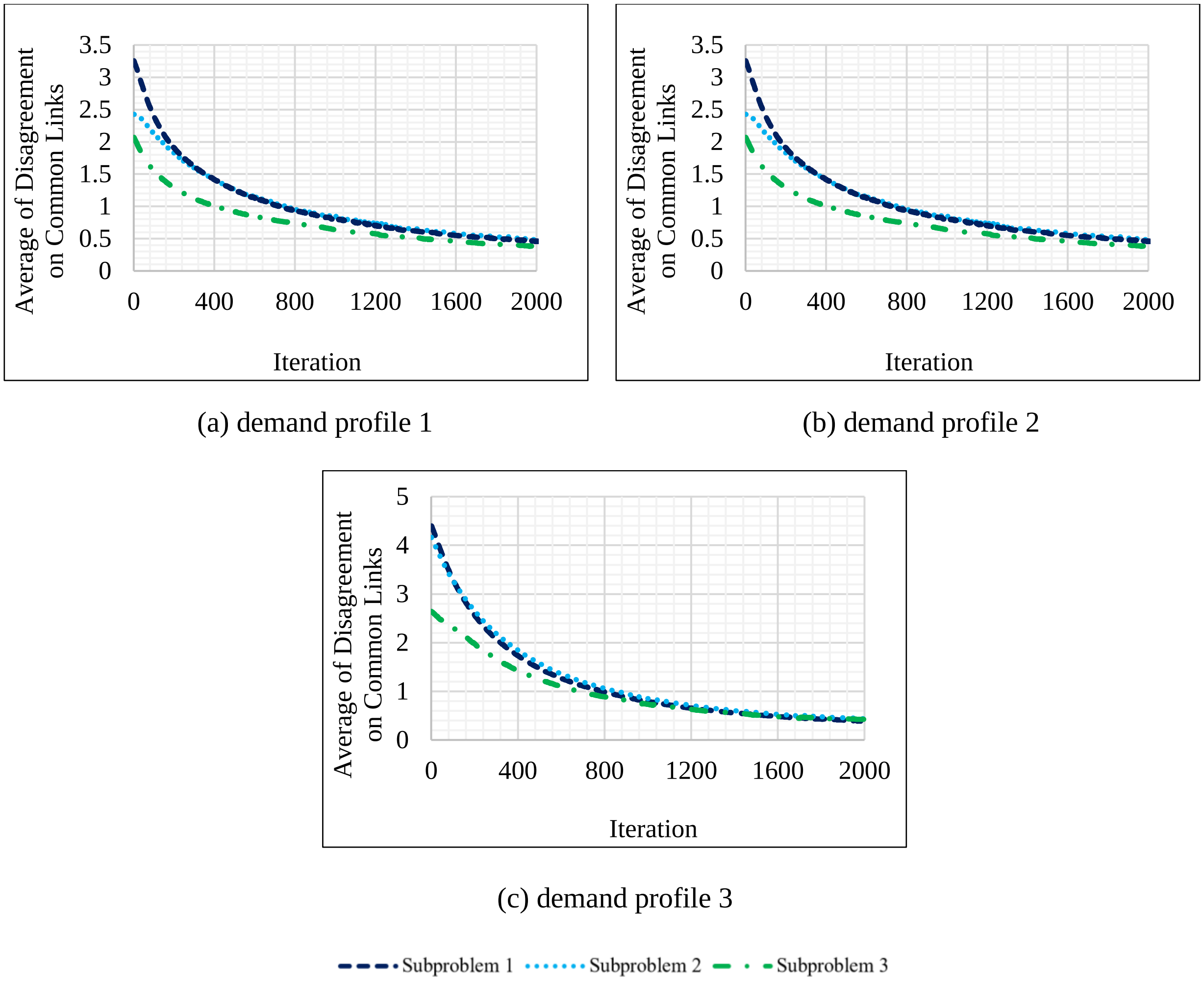
DGA has an intersection-based decomposition, and the number of sub-problems is 20 due to having 20 intersections in the network. This approach does not have any central component or master problem. The number of decision variables differs slightly in each sub-problem of DGA depending on the number of nodes and links and is at least 48% less than the number of variables in the sub-problems of DWDA. The number of iterations in DWDA is at most 101 while this number is 1715 for DGA. Even though we have more number of iterations in DGA, the computation time of each iteration is much less. The total run-time of DGA is 74% more than DWDA in an undersaturated demand pattern. However, when most of the decision variables have non-zero values in the oversaturated condition, the runtime of DGA is improved by 77% compared to DWDA. Moreover, since we have 48% fewer variables in DGA, we only required 5 GB of memory though we needed at least 20 GB of memory to run DWDA.

Table 5-4 Comparison of our approach with the Danzig-Wolfe Decomposition algorithm for the network of 20 intersections © 2022 IEEE

Demand	Approach/Gap	Number of variables in sub-problems	Iterations	Objective Function	Optimality gap (%)	Run-time (hr)	Memory (GB)
1	DWDA	281,200	11	197.69	0.05	0.92	20
	DGA	≤ 146,880	1706	197.34	0.05	1.60	5
	Diff. (%)	-48	15409	0.18	0.00	74	-75
2	DWDA	281,200	53	308.42	0.05	2.34	20
	DGA	≤ 146,880	1708	307.30	0.05	1.81	5
	Diff. (%)	-48	3123	0.36	0.00	-23	-75
3	DWDA	281,200	101	529.29	0.05	8.83	20
	DGA	≤ 146,880	1715	524.84	0.05	2.01	5
	Diff. (%)	-48	1599	0.84	0.00	-77	-75

5.7.2. The Performance of the Methodology

Figure 5-6 presents the disagreement on the value of shared decision variables for three sub-problems with their neighbors for three demand profiles over iterations. In demand profile 1, when each sub-problem has a disagreement of less than 0.5, the algorithm is terminated. The algorithm reached a 5.5% optimality gap for demand profile 1 at iteration 1706, see Figure 5-6.a. The algorithm reached a disagreement value of 0.5 for the second demand pattern at iteration 1708 with the 5.5% optimality gap, see Figure 5-6.b. The approach also reached a disagreement value of 0.5 at iteration 1715 iterations for the third demand pattern with a similar gap of 5.0%. As expected, the disagreement increased with the demand level. However, the gap is the same for all scenarios once the approach is terminated.



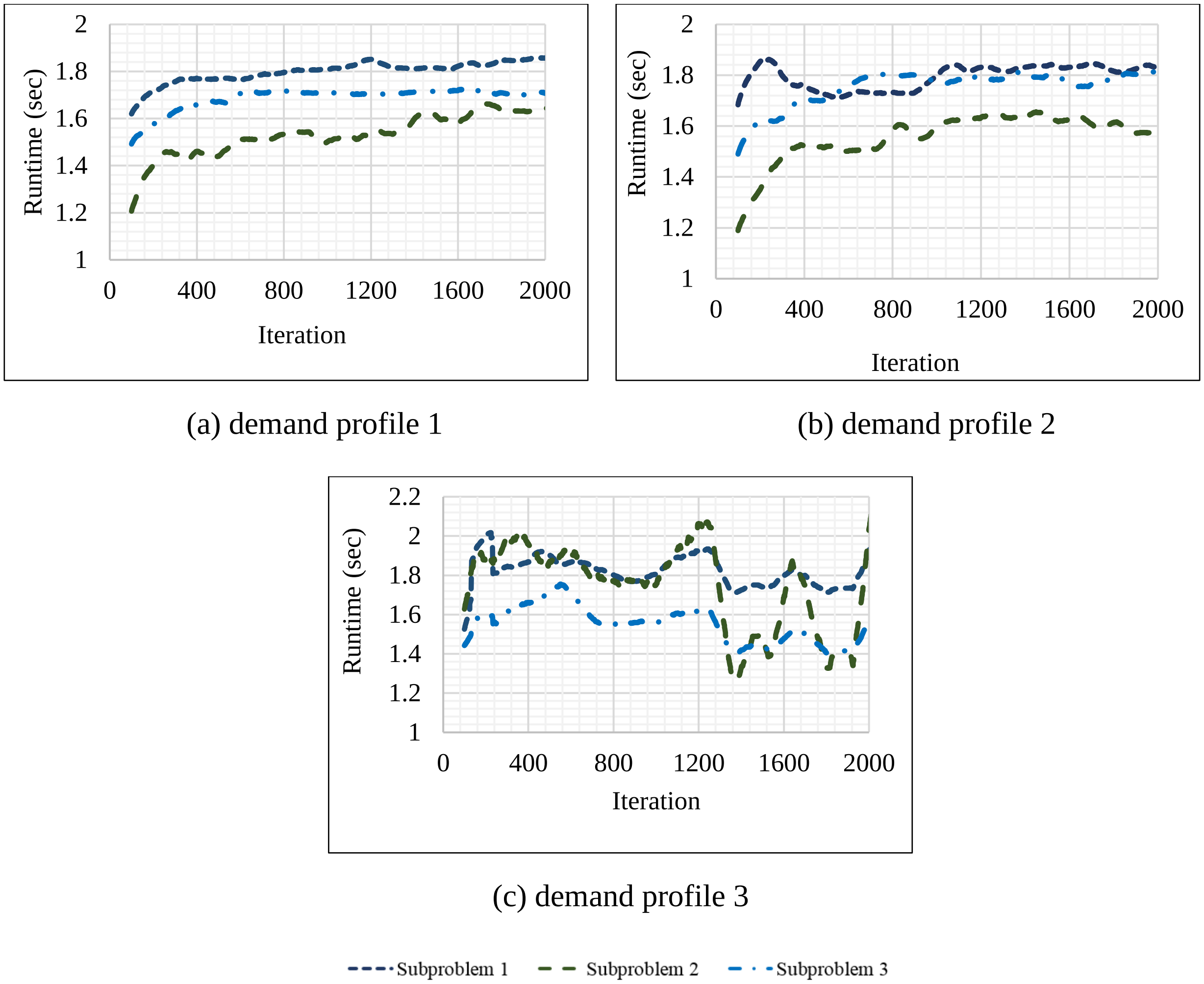
(a) demand profile 1

(b) demand profile 2

(c) demand profile 3

Figure 5-6 The disagreement value ε over iterations for the network of 20 intersections © 2022 IEEE

Figure 5-7. a-c shows the run-time of three sub-problems over iterations for the three demand profiles. We assigned each sub-problem to a different computational node using a multi-thread platform. The sub-problems are independent and optimized synchronously. Each sub-problem represents one intersection with an almost equal number of variables and constraints with other sub-problems. Therefore, the run-time is approximately the same among different nodes, which reduces overhead delays. The run-time of each sub-problem is relatively similar for all three demand profiles, and having a more congested sub-problem did not significantly affect the computation time; however, the total run-time differs by at most 20% between demand profiles due to the additional number of required iterations.



(a) demand profile 1

(b) demand profile 2

(c) demand profile 3

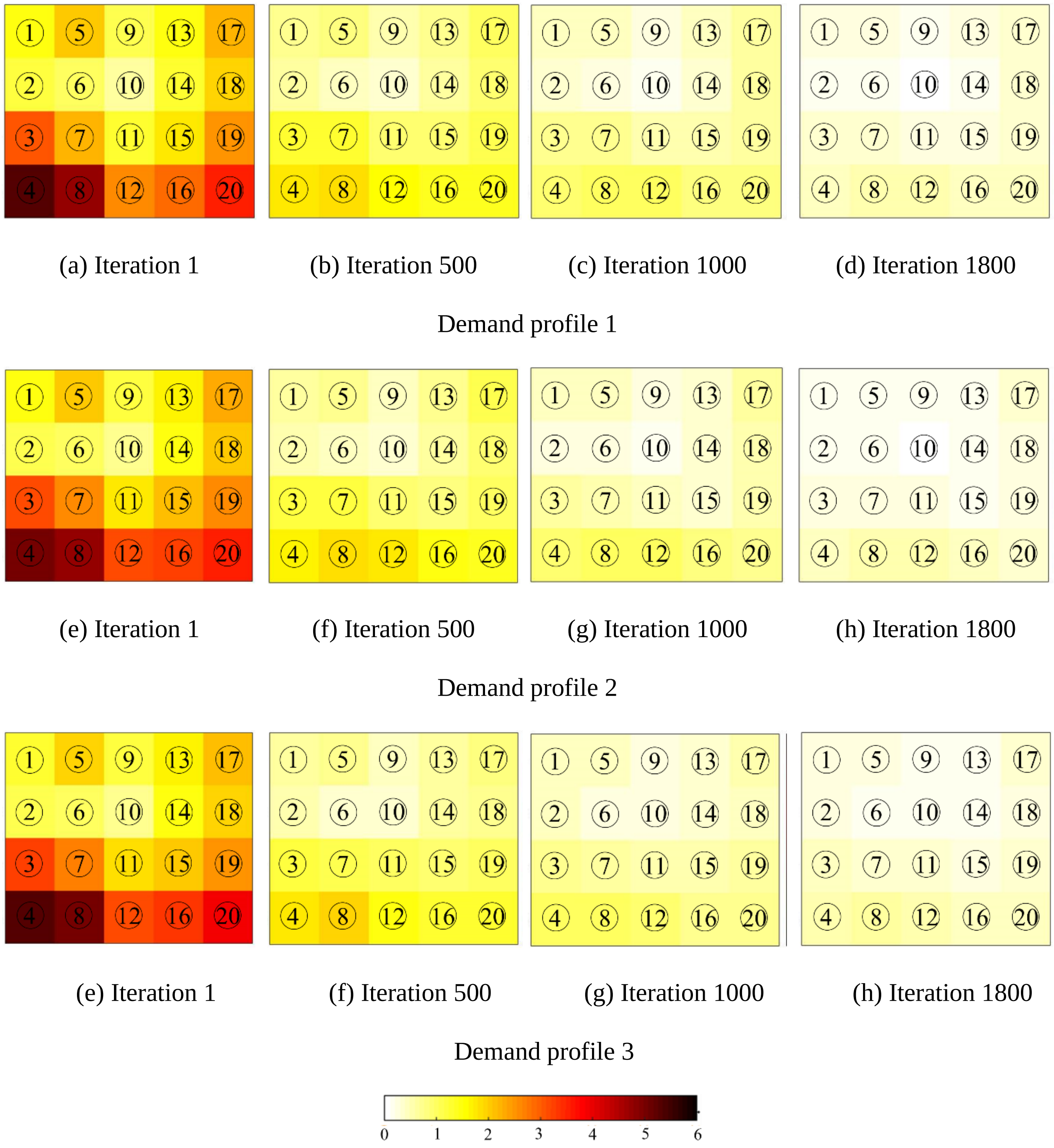
Figure 5-7 The run-time of three sub-problems with three demand profiles in the network of 20 intersections © 2022 IEEE

Table 5-5 shows the computation time for each step of the approach under three demand patterns. The distributed gradient update consists of step 1: computing weighted averages, step 2: moving values towards the negative direction of the gradient, and step 3: projecting values on the feasible region of sub-problems. The runtime for the initialization step is negligible for all demand profiles. Finding the weighted average needs the least CPU time, and projection has the highest computation time. The projection of the value of variables is implemented for all sub-problems at the same time, and its CPU time has increased by increasing the demand. This can happen because more decision variables have non-zero values as the demand increases, and more computational effort is required to find variables' optimal values. No specific trend has appeared for other steps including termination criterion calculation.

Table 5-5 Breakdown of runtimes for the network of 20 intersections © 2022 IEEE

Steps of approach/ demand		Profile 1	Profile 2	Profile 3
Run-time for initialization (hr)		0.0000	0.0000	0.0000
Run-time for distributed gradient update (hr)	Compute weighted averages	0.0023	0.0024	0.0022
	Move values towards negative direction of gradient	0.0035	0.0036	0.0033
	Project values on feasible region	1.5525	1.7643	1.9689
Run-time for checking termination criterion (hr)		0.0351	0.0346	0.0338

Figure 5-8.a-d shows the disagreement on the value of decision variables for each sub-problem with its neighbors at iterations 1, 500, 1000, and 1800 for three demand profiles. Each sub-problem is shown with a number following the same layout shown in Figure 5-4. The spectrum shows a range of colors depending on the value of the disagreement. Darker color represents a higher disagreement value. Figure 5-8. a-d present the disagreement at each sub-problem for the under-saturated demand profile. The colors become lighter as the number of iterations increases, which shows that the conflict on the proposed value by each sub-problem with its neighbors decreases. Figure 5-8. e-h and Figure 5-8. j-m display the same pattern for the second and third demand profiles, respectively.



(a) Iteration 1 (b) Iteration 500 (c) Iteration 1000 (d) Iteration 1800

Demand profile 1

(e) Iteration 1 (f) Iteration 500 (g) Iteration 1000 (h) Iteration 1800

Demand profile 2

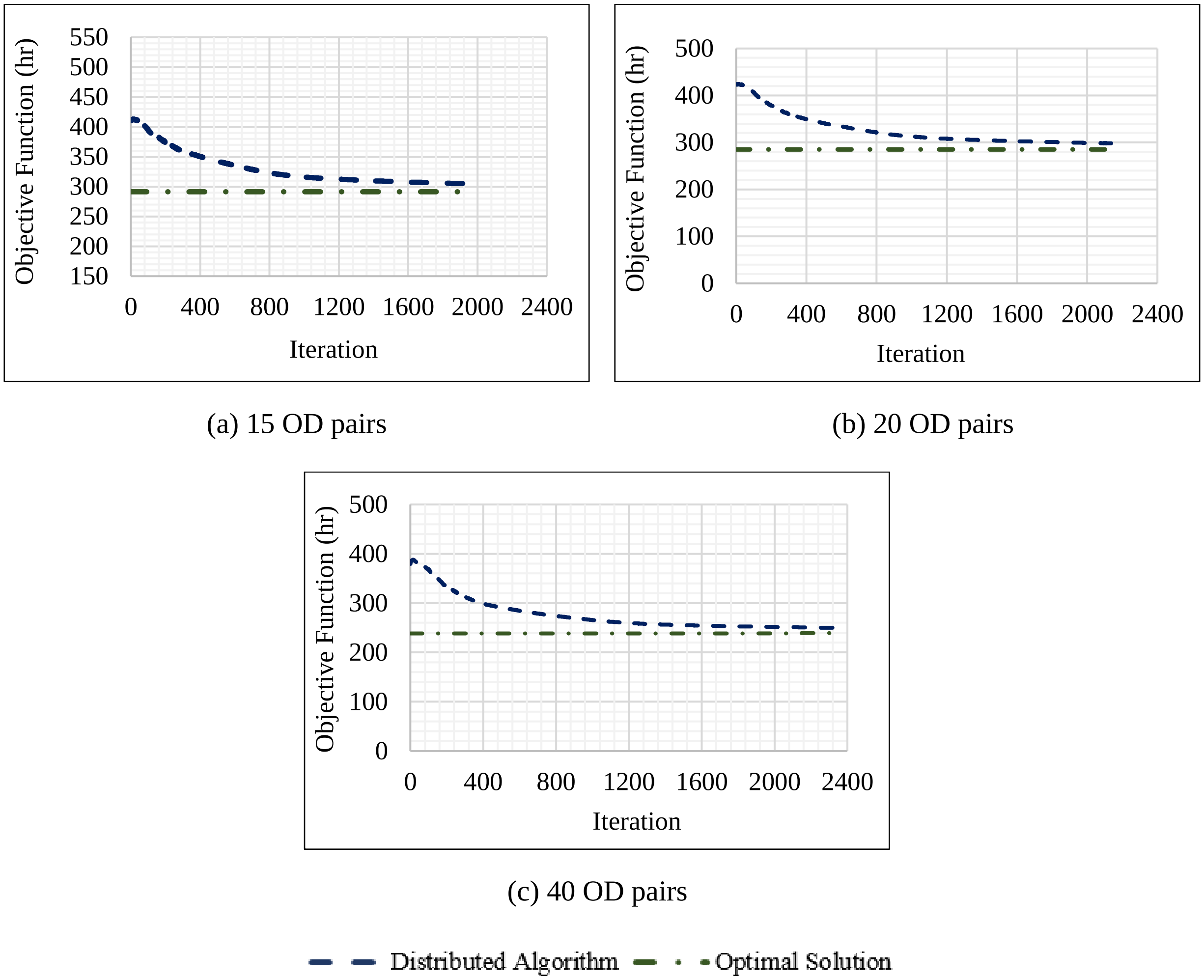
(e) Iteration 1 (f) Iteration 500 (g) Iteration 1000 (h) Iteration 1800

Demand profile 3

Figure 5-8 The value of disagreement at each sub-problem for two demand patterns at iterations 100, 1000, 2000, and 3000 for the network of 20 intersections © 2022 IEEE

Note: the circled numbers represent sub-problem numbers.

Figure 5-9 shows the impact of the number of OD pairs (15, 20, and 40) on the convergence of our approach in the test network of 20 intersections and semi-saturated demand. The algorithm reached the termination criteria in 1708, 2157, and 2319 iterations with 5%, 4%, and 5% optimality gaps for 15, 20, and 40 ODs, respectively. Increasing the number of OD pairs led to more iterations for convergence; however, the number of iterations does not increase as fast as the number of decision variables. Specifically, increasing the number of ODs from 20 to 40 doubles the number of decision variables but only increases the number of iterations by 7%.



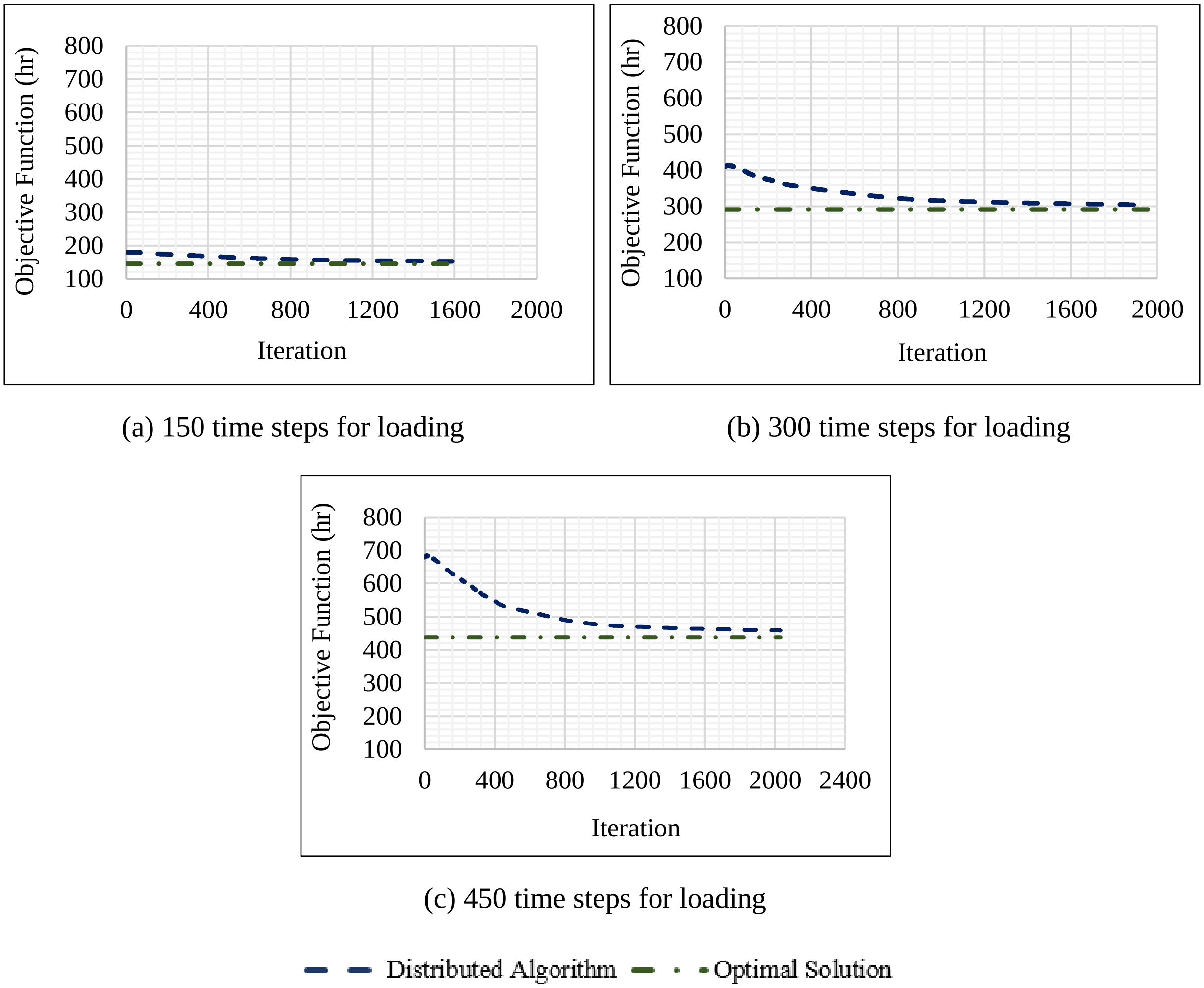
(a) 15 OD pairs

(b) 20 OD pairs

(c) 40 OD pairs

Figure 5-9 The effects of the number of OD pairs on DGA convergence © 2022 IEEE

In this section, we evaluated the effect of increasing the duration of the study period on convergence. We tested three loading periods of 150, 300, and 450 time steps, as shown in Figure 5-10. The three cases required 1420, 1708, and 2081 iterations with 5%, 4%, and 4% optimality gaps to reach the termination criterion. As we increased the loading time, the number of required iterations to meet the termination was increased. Increasing loading time created more congestion. Therefore, the number of iterations was increased by increasing the loading period. However, the rate of increase in the required iterations was much less than the increase in the number of decision variables.



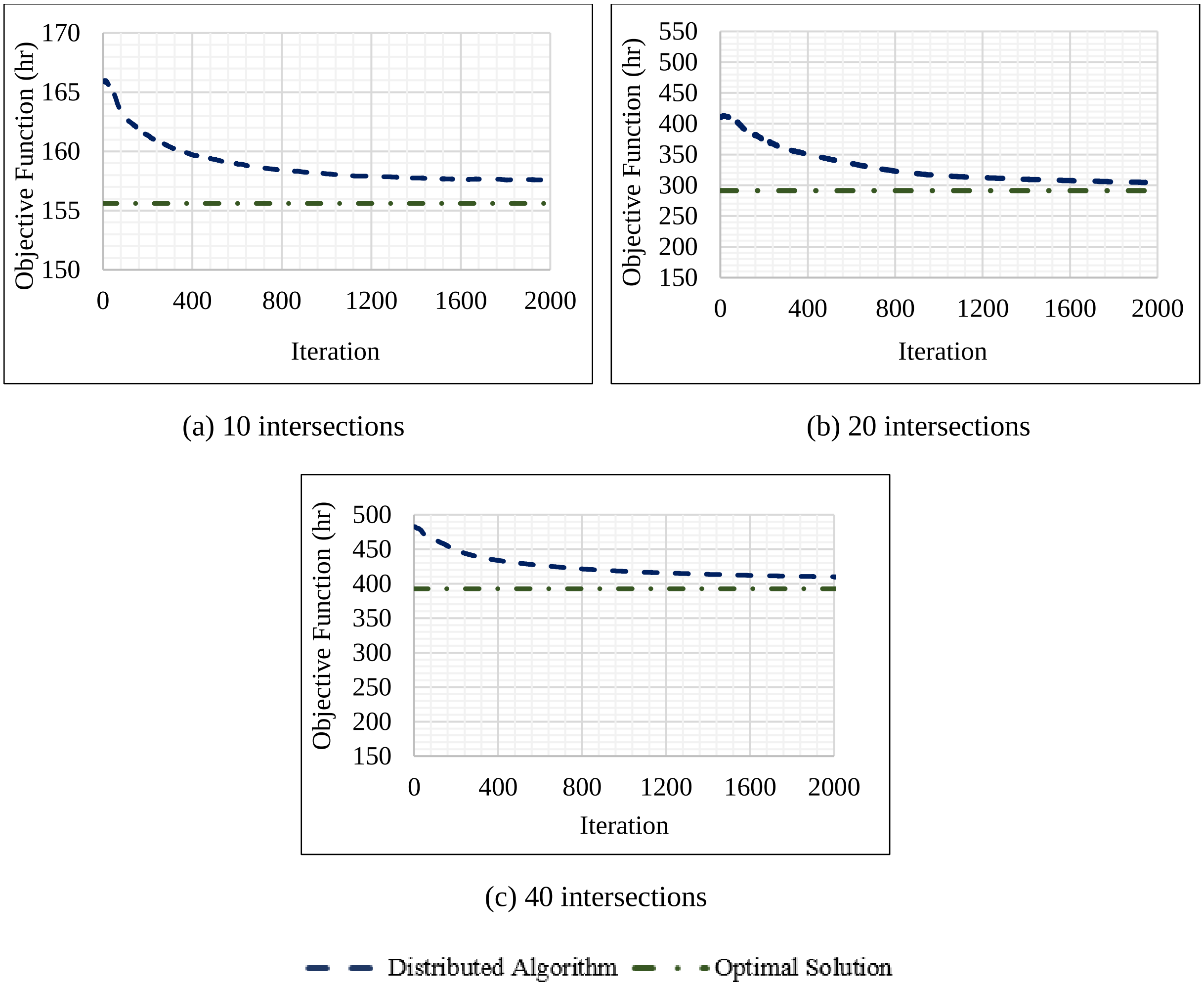
(a) 150 time steps for loading

(b) 300 time steps for loading

(c) 450 time steps for loading

Figure 5-10 The effect of study period duration on DGA convergence © 2022 IEEE

We also studied the effects of network size on convergence by looking at networks of 10, 20, and 40 intersections with similar characteristics, as shown in Figure 5-11. An increase in network size increased the number of decision variables from 1,674,000 to 4,218,000, and 14,120,00. DGA was converged in 807, 1708, and 1710 iterations with 2%, 4%, and 5% optimality gaps for networks with 10, 20, and 40 intersections, respectively. The loading period, demand, and the number of OD pairs were the same in all cases. The number of iterations for convergence was increased by increasing the network size; however, at a rate much slower than the increase in the network size. Increasing the size of the network from 20 intersections to 40 increases the number of required iterations by two, which shows the scalability of the proposed methodology.



(a) 10 intersections

(b) 20 intersections

(c) 40 intersections

Figure 5-11 The effect of network size on DGA convergence © 2022 IEEE

We also tested the approach on a network with 40 intersections with 632 cells, 780 links, 400 time steps, and 25 OD pairs that bring the total number of decision variables to 14,120,000. We used the same termination criterion that was applied to other cases. By increasing the intersections from 20 to 40, the number of decision variables increased from ~4 million to ~14 million (more than a factor of 3). We need to mention that the literature that includes models of networks with thousands of intersections and OD pairs uses either exit flow functions, point queue models, or link-performance functions. These approaches are aggregated and have significantly fewer decision variables and do not provide the accuracy that is required for traffic operation purposes. In this study, we use the CTM model, which is more accurate but at the expense of additional complexity. The network of 40 intersections is significantly larger than comparable studies that have used the cell transmission model (Aziz and Ukkusuri, 2012; Chiu and Zheng, 2007; Doan and Ukkusuri, 2015). The studies solve the problem for 5805 to 489,700 decision variables (Li et al., 2003; Zheng and Chiu, 2011).

Table 5-6 presents the number of cells, links, time steps, and decision variables for cases with 20 and 40 intersections. Increasing the number of links from 387 to 780 and nodes from 316 to 632 does not change the complexity of each sub-problem due to the intersection-based distribution of the formulation. However, increasing the study period from 400 to 500 time steps and OD pairs from 15 to 25 lead to having more variables. Even though the variables are increased by 70%, we can find the solutions with at most a 5.70% optimality gap in at most 5.34 hours.

Table 5-6 The effect of increasing the network size from 20 to 40 intersections on DGA © 2022 IEEE

Demand Pattern	Intersections	Cells	Links	Time steps	Variables	Optimality gap (%)	Runtime (hr)	Iterations to satisfy termination
Under-saturated	20	316	387	400	4,218,000	5.90	1.60	1704
	40	632	780	500	14,120,00	5.68	5.19	1710
Semi-saturated	20	316	387	400	4,218,000	5.28	1.81	1708
	40	632	780	500	14,120,00	5.68	5.26	1711
Over-saturated	20	316	387	400	4,218,000	5.93	2.01	1715
	40	632	780	500	14,120,00	5.70	5.34	1716

Figure 5-12 shows the total run-time for sub-problems for the network of 20 intersections with 15 and 30 ODs and the network of 40 intersections with 25 ODs. Increasing the number of OD pairs from 15 to 30 in the network of 20 intersections has increased the number of decision variables by 50% and run-time at each iteration on average by 71%. When the size increases, the number of variables in each sub-problem increases only by the number of OD pairs in this case. The number of cells, links, and time steps of the horizon remain constant. By increasing the number of intersections from 20 with 15 ODs to 40 intersections with 25 ODs, the number of decision variables in each sub-problem has increased by 70%. The run-time to generate solutions at each time step has increased by 52% on average.

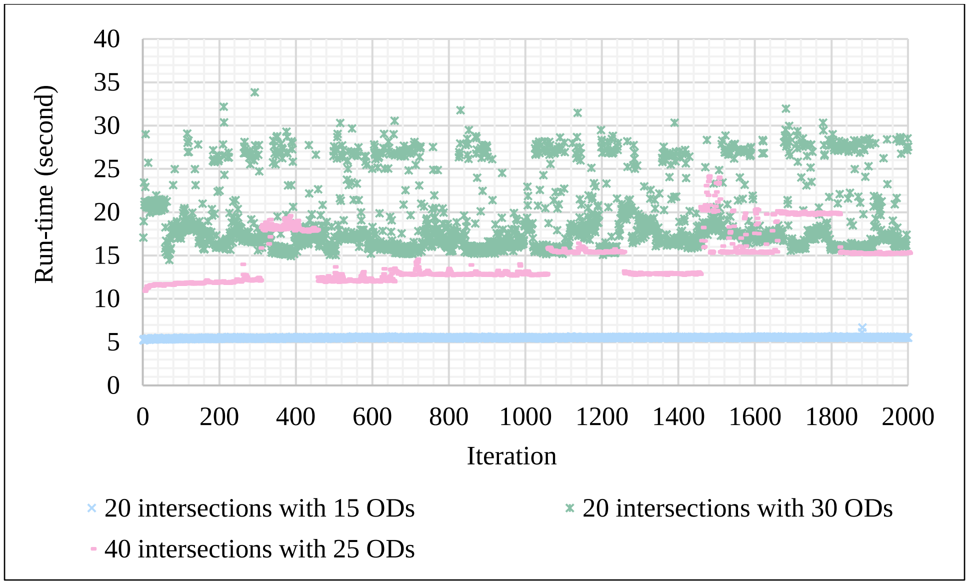


Figure 5-12 The run-time of approach using different cases of 20 intersections with 15 and 30 ODs and 40 intersections with 25 ODs © 2022 IEEE

CHAPTER 6. DISTRIBUTED COORDINATED METHODOLOGY TO SOLVE SODTA

The proposed methodology in this chapter consists of two main components: distributed optimization and distributed coordination. The distributed optimization decomposes a network-level SODTA problem into several intersection-level sub-problems by relaxing the constraints that represent interrelationships between sub-problems. Dummy source and sink cells are added to the sub-problems (when needed) to make them stand-alone systems capable of making their own decisions. These sub-problems are optimized in parallel and simultaneously. The distributed coordination exchanges data among the sub-problems and implements it in their objective functions and constraints to reduce the possibility of finding locally optimal solutions. The exchange of data is key to maintaining an appropriate balance between computational efficiency and solution quality.

Each sub-problem receives information on the (a) number of incoming vehicles, (b) available capacity of receiving cells at the adjacent sub-problems, and (c) and travel time on network cells. The information is estimated by a CTM simulation run (will be detailed later in this section) and is incorporated into each sub-problem by either re-introducing and reinforcing the relaxed constraints or modifying the objective function. The information on travel times is used to help assign traffic to appropriate routes. The algorithm estimates the shortest travel times from the dummy sink cells of each sub-problem to the network sinks to estimate the remaining travel time and select how traffic should be assigned within each sub-problem. The estimated travel times are incorporated in the objective function of each sub-problem as a penalty term to give a higher priority to the sink cells of a sub-problem that are estimated to have a shorter travel time to the actual network destination. The details of the proposed algorithm will follow. Table 6-1 presents the notations used in this chapter.

Table 6-1 Definition of sets, decision variables, and parameters used in chapter 6

Sets:	
T	The set of all time steps
C	The set of all network cells
C_O	The set of all source cells
C_S	The set of all sink cells
C_I	The set of all intersection cells
C_{OD}	The set of all OD pairs
$P(i)$	The set of all predecessors to cell $i \in C$
$S(i)$	The set of all successors to cell $i \in C$
C_E	The set of all ordinary cells
C_D	The set of all diverge cells
N	The set of all sub-problems
C^k	The set of all cells that belong to sub-problem $k \in N$
${C_O}^k$	The set of all source cells that belong to sub-problem $k \in N$
${C_S}^k$	The set of all sink cells that belong to sub-problem $k \in N$
${C_I}^k$	The set of all intersection cells that belong to sub-problem $k \in N$
${C_{OD}}^k$	The set of all OD pairs that belong to sub-problem $k \in N$
$P(i)^k$	The set of all predecessors to cell $i \in C$ that belong to sub-problem $k \in N$
$S(i)^k$	The set of all successors to cell $i \in C$ that belong to sub-problem $k \in N$
${C_D}^k$	The set of all diverge cells that belong to sub-problem $k \in N$
Decision variables:	
$x_i^{t,od}$	The number of vehicles in cell $i \in C$ at time step $t \in T$ with OD pair $(o,d) \in C_{OD}$
$y_{ij}^{t,od}$	The number of vehicles flowing from cell $i \in C$ to downstream cell $j \in S(i)$ at time step $t \in T$ with OD pair $(o,d) \in C_{OD}$
Parameters:	
τ	The duration of each time step
$d_i^{t,od}$	The entry demand level at source cell $i \in C_O$ at time step $t \in T$ from origin o to destination d in OD pair $(o,d) \in C_{OD}$

Table 6-1 (continued)

F_i	The saturation flow rate at cell $i \in C$
M_i	The maximum number of vehicles that cell $i \in C$ can accommodate
g_i^t	A binary parameter to define signal status at intersection cell $i \in C_I$ at time step $t \in T$. Zero and one values indicate red and green signals, respectively.
μ	An arbitrary small and positive number
$R_{ij}^{t,od}$	The turning ratio of diverge cell $i \in C_D$ to successor cell $j \in S(i)$ at time step $t \in T$ for OD pair $(o,d) \in C_{OD}$
$f_i^t = g_i^t F_i$	The variable saturation flow rate of intersection cell $i \in C_I$ at time step $t \in T$
$\hat{x}_i^{t,od}$	The number of vehicles in cell $i \in C$ at time step $t \in T$ with OD pair $(o,d) \in C_{OD}$ obtained from a CTM simulation
$\hat{y}_{ij}^{t,od}$	The number of vehicles flowing from cell $i \in C$ to downstream cell $j \in S(i)$ at time step $t \in T$ with OD pair $(o,d) \in C_{OD}$ obtained from a CTM simulation
$\overline{x}_i^t$	The total number of vehicles in cell $i \in C$ at time step $t \in T$ (over all OD pairs $(o,d) \in C_{OD}$)
$\overline{y}_{ij}^t$	The total number of vehicles flowing from cell $i \in C$ to downstream cell $j \in S(i)$ at time step $t \in T$ (over all OD pairs $(o,d) \in C_{OD}$)
$\mathcal{X}_{ij}^t$	The total number of vehicles in diverge cell $i \in C_D$ at time step $t \in T$ heading to successor cell $j \in S(i)$
$\mathfrak{X}_{ij}^{t,od}$	The number of vehicles in diverge cell $i \in C_D$ at time step $t \in T$ heading to cell $j \in S(i)$ with OD pair $(o,d) \in C_{OD}$
$\wp_{i,od}^{t'}$	Vehicles that entered cell i at time step t' with origin o and destination d
$\mathcal{T}_{ij}$	The time that not all vehicles can exit a cell $i \in C$ outgoing to $j \in S(i)$
$\mathscr{f}_{ij}$	The fraction of vehicles that can leave cell $i \in C$ in time $\mathcal{T}_{ij}$
z_{ij}	An axillary variable for links between cells $i \in C_D, j \in S(i)$
ΔT	The prediction horizon

6.1. Distributed Optimization

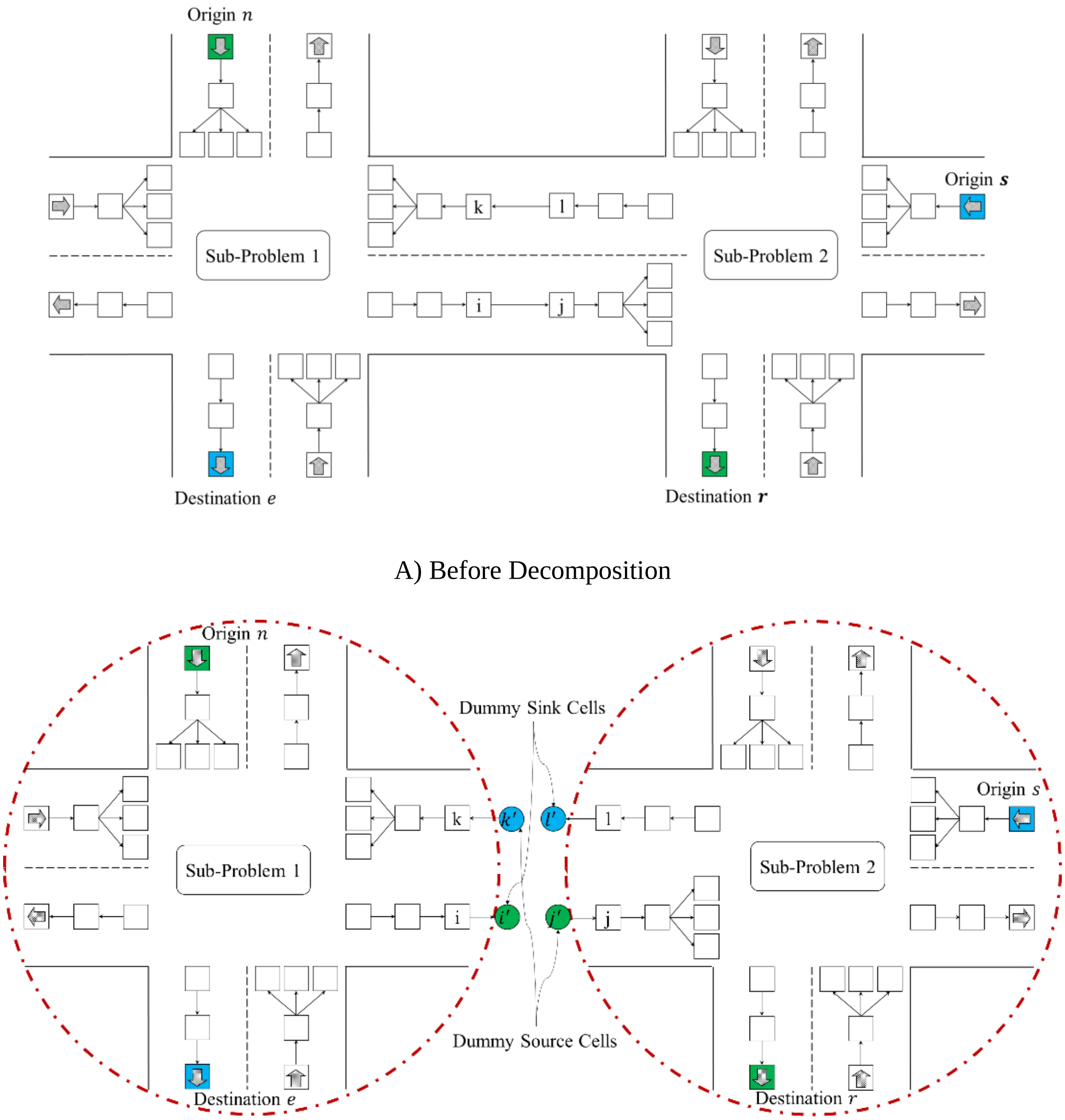
Figure 6-1 shows a network of two intersections with a CTM cell representation and two OD pairs (s,e) in blue and (n,r) in green colors. The network is shown before and after distribution in parts A and B, respectively. The formulation is distributed by relaxing the constraints representing the interrelationship between the two intersections and adding a penalty function to the objective function (similar to the Lagrangian relaxation approach). The relaxation is equivalent to disconnecting the link between cells $i \in C^1$ and $j \in C^2$ and cells $k \in C^1$ and $l \in C^2$ or relaxing Constraints (3-3)-(3-9) for links between cells $i \in C^1$ to $j \in C^2$ and cells $l \in C^2$ to $k \in C^1$. Dummy source cells $j' \in {C_O}^2$, $k' \in {C_O}^1$ and sink cells $l' \in {C_S}^2$ and $i' \in {C_S}^1$ are added after omitting the links to make each sub-problem a stand-alone system. Constraint (3-4) is added for new source cells $j' \in {C_O}^2$ and $k' \in {C_O}^1$ in their corresponding sub-problems' formulations. Constraint (3-5) is also added for new sink cells $l' \in {C_S}^2$ and $i' \in {C_S}^1$.

The distribution affects the set of OD pairs $C_{OD} = \{(n, r), (s, e)\}$ when either an origin or a destination for an OD belongs to different sub-problems. This is the case for both OD pairs shown in Figure 6-1. Therefore, the set of OD pairs will be ${C_{OD}}^1 = \{(n, i'), (k', e)\}$ for Sub-problem 1, and ${C_{OD}}^2 = \{(s, l'), (j', r)\}$ for Sub-problem 2. Example of intersection level distribution for signal timing optimization and DTA can be found in (Mehrabipour, 2018; Mehrabipour and Hajbabaie, 2020). Hajbabaie et al., (2020), Mohebifard and Hajbabaie (2019b) and Mohebifard and others (2021) also studied the distribution of a CTM-based formulation for traffic metering.

Note that each sub-problem $k \in N$ optimizes the outgoing flow $y_{ij}^{t,od}$ from diverge cell $i \in {C_D}^k$ to successor cell $j \in S(i)^k$ at time step $t \in T$ for $(o, d) \in {C_{OD}}^k$ exclusively. Note that the maximization of outflows in a CTM-based formulation is equivalent to the total travel time minimization. The decision variables corresponding to the diverge cells and their outflow are controlled only by one sub-problem. Therefore, the turning ratios can be found for the entire network without any conflict and infeasibility. After optimizing all sub-problems, the turning ratios $R_{ij}^{t,od}$for the link between diverge cell $i \in C_D$ and successor cell $j \in S(i)$ at time step $t \in T$ with $(o, d) \in C_{OD}$ are computed as shown in Equation (6-1). These turning ratios are given as inputs to the distributed coordination.

$$R_{ij}^{t,od} = y_{ij}^{t,od} \Big/ \sum_{j \in S(i)} y_{ij}^{t,od} \qquad \forall t \in T, i \in C_D, j \in S(i), (o,d) \in C_{OD} \tag{6-1}$$

Note that besides network decomposition, the study period is also decomposed into smaller horizons using the rolling horizon technique for faster performance.



A) Before Decomposition

B) After Decomposition

Figure 6-1 An example of a network of two intersections before and after decomposition

6.2. Distributed Coordination

The independent optimization of the localized sub-problems does not lead to finding a system-level optimal solution. Furthermore, the solutions of these sub-problems may not be feasible to the original formulation. Therefore, an effective coordination scheme among the sub-problems is a critical step towards reducing the possibility of finding locally optimal or infeasible solutions. As mentioned before, each sub-problem needs to receive information on the (a) incoming vehicles, (b) available capacity of receiving cells at neighboring sub-problems, and (c) travel time on the network cells. The information is estimated by a CTM simulation run and is incorporated into each sub-problem by either re-introducing and reinforcing the relaxed constraints and/or modifying the objective function. We will first show how each sub-problem's constraints and objective function are modified and how the CTM estimates the required information.

The entry demand at the dummy source cells of each sub-problem needs to be communicated. Constraint (6-2) updates the entry demand at the boundary cells according to the number of vehicles coming from the neighboring intersections. Following the example presented in Figure 6-1, traffic demand $d_{k'}^{t,od}$ in dummy source cell $k' \in {C_O}^1$ (see Figure 6-1.B) is equal to the number of vehicles $\hat{y}_{lk}^{t,od}$ flowing from cell $l \in C^2$ to cell k $\in C^1$ at time step $t \in T$ for $(o,d) \in {C_{OD}}^1$ (see Figure 6-1.A). The flow of vehicles between the two cells is estimated by a CTM simulation based on turning ratios given from all sub-problems.

$$d_{k'}^{t,od} = \hat{y}_{lk}^{t,od} \qquad \forall t \in T, (o,d) \in {C_{OD}}^1 \tag{6-2}$$

Constraint (6-3) ensures that Sub-problem 1 does not send more vehicles than the available capacity of receiving cell $j \in C^2$ in Sub-problem 2. Hence, the outgoing flow $y_{ii'}^{t,od}$ from cell $i \in C^1$ to cell $i' \in {C_S}^1$ at time $t \in T$ for $(o,d) \in {C_{OD}}^1$ in Sub-problem 1 is limited to the available capacity $\left(M_j - \sum_{\forall (o,d) \in {C_{OD}}^2} \hat{x}_j^{t,od}\right)$ of receiving cell $j \in C^2$ at time step $t \in T$. The available capacity is found by estimating the total number of vehicles $\sum_{\forall(o,d)\in {C_{OD}}^2} \hat{x}_j^{t,od}$ in cell $j \in C^2$ at time step $t \in T$ by a CTM simulation. Cell $j \in C^2$ is shown in Figure 6-1.A, and cells $i \in C^1$ and $i' \in {C_S}^1$ are shown in Figure 6-1.B.

$$y_{ii'}^{t,od} \leq M_j - \sum_{(o,d)\in {C_{OD}}^2} \hat{x}_j^{t,od} \qquad \forall t \in T, (o,d) \in {C_{OD}}^1 \tag{6-3}$$

Information on cell travel times is used to help assign traffic to appropriate routes. The approach estimates the shortest travel times using Dijkstra's algorithm (Ahuja et al., 1993) from the dummy sink cells of each sub-problem to the sink cells of the network to estimate the remaining travel time and select how the traffic should be assigned within each sub-problem. The estimated travel times will be incorporated into the objective function of each sub-problem $k \in N$ as a penalty term $-1/\alpha_j^{t,od}$(for sink cell $i \in {C_S}^k$ in the sub-problem $k \in N$ at time step $t \in T$ with $(o,d) \in {C_{OD}}^k$) to give a higher priority to the sink cells that have a shorter travel time to the actual destinations of the network. The term $\alpha_j^{t,od}$ is defined as the total travel time on the shortest path. The objective function of each sub-problem is redefined as shown in Expression (6-4).

$$\min \sum_{t\in T} \sum_{(o,d)\in C_{OD}^k} \sum_{i\in C^k \setminus C_S^k} x_i^{t,od} + \sum_{t\in T} \sum_{(o,d)\in C_{OD}^k} \sum_{j\in C_S^k} -x_j^{t,od}/\alpha_j^{t,od} \tag{6-4}$$

The total travel time on the shortest path (denoted by $\alpha_j^{t,od}$) is a positive value that is assigned to sink cells. To absorb vehicles into sink cells in a minimization problem, term $-1/\alpha_j^{t,od}$ as a penalty function needs to be used not to interfere with the first term in Expression (6-4).

A link-based CTM simulation is run for the entire network to find the required information to be exchanged among adjacent sup-problems. We modified the path-based CTM simulation developed by Ukkusuri et al. (2012) and created a link-based simulation by changing the definition of decision variables, incorporating turning ratios, and updating the equations. The main modification is changing the definition of variables so that the index of variables showing path represents OD pairs. The other modification is computing the value of decision variables for vehicles moving out from a diverge cell based on the turning ratio instead of checking the cells that appeared in a path. The OD component was added to the turning ratios to ensure that the demand between each OD pair was met.

The distributed optimization of each sub-problem provides turning ratios $R_{ij}^{t,od}$ as inputs to the CTM. The turning ratios are defined for links between diverge cell $i \in C_D$ to successor cell $j \in S(i)$ at time step $t \in T$ with $(o,d) \in C_{OD}$. The CTM estimates the number of vehicles in each cell and the flow between adjacent cells over the duration of a prediction period. The CTM simulation equations are described below. The algorithm estimates the initial value of decision variables by generating the shortest path between each origin and destination. It then uses a path-based simulation to compute the value of variables. However, a link-based simulation is used in each iteration to be compatible with link-based sub-problems.

The total number of vehicles $\overline{x}_i^t$ in cell $i \in C$ at time step $t \in T$ is the summation of the number of vehicles in that cell over all OD pairs as shown in Equation (6-5). The same concept is used to find the total flow between two adjacent cells as shown in Equation (6-6).

$$\overline{x}_i^t = \sum_{(o,d)\in C_{OD}} \hat{x}_i^{t,od} \qquad \forall t \in T, i \in C \qquad (6\text{-}5)$$

$$\overline{y}_{ij}^t = \sum_{(o,d)\in C_{OD}} \hat{y}_{ij}^{t,od} \qquad \forall t \in T, i \in C, j \in S(i) \qquad (6\text{-}6)$$

Equation (6-7) determines the total flow between adjacent cells, similar to flow feasibility constraints explained in the problem formulation. The OD-level flow of vehicles between two adjacent cells (from cell $i \in C$ to cell $j \in S(i)$) is determined based on Equation (6-8) by distributing the flow proportional to the ratio of OD-level to the total cell occupancy $(x_i^{t,p}/\overline{x}_i^t)$.

This equation is converted to Equation (6-9) to formulate an if-condition and ensure that it works with empty cells as well. Note that for intersection cells $i \in C_I$, we substitute F_i by $g_i^t F_i$.

$$\overline{y}_{ij}^{t} = min\,\{\overline{x}_i^t\,, F_i\,, F_j\,, M_j - \overline{x}_j^t\} \qquad \forall t \in T, (i,j) \in E_O \tag{6-7}$$

$$\hat{y}_{ij}^{t,od} = \begin{cases} min\,\{\overline{x}_i^t\,, F_i\,, F_j\,, M_j - \overline{x}_j^t\} \times \hat{x}_i^{t,od} / \overline{x}_i^t & if\ \overline{x}_i^t > 0 \\ 0 & Otherwise \end{cases} \qquad \forall t \in T, (i,j) \in E_O, (o,d) \in C_{OD} \tag{6-8}$$

$$\hat{y}_{ij}^{t,od} = min\,\{\overline{x}_i^t\,, F_i\,, F_j\,, M_j - \overline{x}_j^t\} \times \hat{x}_i^{t,od} / \overline{x}_i^t + \mu \qquad \forall t \in T, (i,j) \in E_O, (o,d) \in C_{OD} \tag{6-9}$$

We define two new variables for diverge cells to be able to track the paths of vehicles through these cells. Let $\mathcal{X}_{ij}^t$ denote the number of vehicles in diverge cell $i \in C_D$ that are headed to successor cell $j \in S(i)$ at time step $t \in T$, see Equation (6-10). Note that $\mathfrak{x}_{ij}^{t,od}$ represents the OD-level definition of $\mathcal{X}_{ij}^t$, see Equation (6-11).

$$\mathcal{X}_{ij}^t = \sum_{(o,d) \in C_{OD}} \hat{x}_i^{t,od} R_{ij}^{t,od} \qquad \forall t \in T, i \in C_D, j \in S(i) \tag{6-10}$$

$$\mathfrak{x}_{ij}^{t,od} = \hat{x}_i^{t,od} R_{ij}^{t,od} \qquad \forall t \in T, i \in C_D, j \in S(i), (o,d) \in C_{OD} \tag{6-11}$$

The flow from diverge cells is found using Equations (6-12)-(6-15). There are two possible conditions for the maximum total outflow $\sum_{j \in S(i)} min\,\{\mathcal{X}_{ij}^t, F_j\,, M_j - \overline{x}_j^t\}$ for diverge cell $i \in C_D$ at time step $t \in T$. If the first condition in (6-12) is satisfied, the total flow $\overline{y}_{ij}^t$ will be the minimum of $\mathcal{X}_{ij}^t, F_j$ and $M_j - \overline{x}_j^t$. Otherwise, Equation (6-13) is applied, where $\overline{y}_{ij}^t$ is limited to saturation flow rate F_i, and F_i will be distributed among all outgoing links from diverge cell $i \in C_D$. Equation (6-14) is a compact form of (6-12) and (6-13). Equation (6-15) finds the OD-level flow $y_{ij}^{t,od}$ by distributing the total flow $\overline{y}_{ij}^{t}$ among all links of diverge cell $i \in C_D$ at time step $t \in T$ according to $\mathfrak{X}_{ij}^{t,od} / \mathcal{X}_{ij}^{t} + \mu$ ratio.

$$if \sum_{j \in S(i)} min\,\{\mathcal{X}_{ij}^{t}, F_j\,, M_j - \overline{x}_j^{t}\} \leq F_i \qquad , \overline{y}_{ij}^{t} = min\,\{\mathcal{X}_{ij}^{t}, F_j\,, M_j - \overline{x}_j^{t}\} \qquad \forall t \in T, i \in C_D, j \in S(i) \tag{6-12}$$

$$if \sum_{j \in S(i)} min\,\{\mathcal{X}_{ij}^{t}, F_j\,, M_j - \overline{x}_j^{t}\} > F_i \qquad , \overline{y}_{ij}^{t} = F_i \frac{min\,\{\mathcal{X}_{ij}^{t}, F_j\,, M_j - \overline{x}_j^{t}\}}{\sum_{j \in S(i)} min\,\{\mathcal{X}_{ij}^{t}, F_j\,, M_j - \overline{x}_j^{t}\}} \qquad \forall t \in T, i \in C_D, j \in S(i) \tag{6-13}$$

$$\overline{y}_{ij}^{t} = min\,\{\mathcal{X}_{ij}^{t}, F_j\,, M_j - \overline{x}_j^{t}\} min\,\{1, \frac{F_i}{\sum_{j \in S(i)} min\{\mathcal{X}_{ij}^{t}, F_j\,, M_j - \overline{x}_j^{t}\} +} \qquad \forall t \in T, i \in C_D, j \in S(i) \tag{6-14}$$

$$\hat{y}_{ij}^{t,od} = \overline{y}_{ij}^{t}\, \mathfrak{X}_{ij}^{t,od} / \mathcal{X}_{ij}^{t} + \mu \qquad \forall t \in T, i \in C_D, j \in S(i), (o,d) \in C_{OD} \tag{6-15}$$

Equations (6-16), (6-17), and (6-18) are the conservation flow constraints to update $x_i^{t+1,od}$ for ordinary cell $i \in\ C \setminus \{C_S\,, C_O\}$, source cell $i \in C_O$ and sink cell $i \in C_S$, respectively.

$$\hat{y}_{ki}^{t,od} - \hat{y}_{ij}^{t,od} = \hat{x}_i^{t+1,od} - \hat{x}_i^{t,od} \qquad \forall t \in T, i \in\ C \setminus \{C_S\,, C_O\}, k \in P(i), j \in S(i), (o,d) \in C_{OD} \tag{6-16}$$

$$\varphi^{t,od} - \hat{y}_{ij}^{t,od} = \hat{x}_i^{t+1,od} - \hat{x}_i^{t,od} \qquad \forall t \in T, i \in C_O\,, j \in S(i), (o,d) \in C_{OD} \tag{6-17}$$

$$\hat{y}_{ki}^{t,od} = \hat{x}_i^{t+1,od} - \hat{x}_i^{t,od} \qquad \forall t \in T, i \in C_S, k \in P(i), (o,d) \in C_{OD} \tag{6-18}$$

6.3. FIFO implementation

In this section, we present a link-based simulation to approximate the First–In–First–Out

(FIFO) rule as described by Carey et al. (2014). The flow is computed similar to CTM simulation and disaggregated based on the entry time to the current link. The traffic is labeled by its current cell, entry time to link, and path and exit the cell following the same order that it enters the link. We use $\mathscr{p}_{i,od}^{t'}$ to denote vehicles that entered cell i at time step t' with origin o and destination d. In equation (6-19), the value of $\mathscr{p}_{i,od}^{t'}$ are initialized and set equal to 0.

$$\mathscr{p}_{i,od}^{0} = 0 \qquad \forall\, i \in C, (o,d) \in C_{OD} \tag{6-19}$$

In equation (6-20), we set the value of occupancy $x_i^{0,od}$ at time zero to $\mathscr{p}_{i,od}^{0}$. Let $\overline{x}_i^t$ denote total number of vehicles $\overline{x}_i^t$ in cell $i \in C$ at time step t for all OD pairs. Equation (6-21) shows the value of $\overline{x}_i^0$ that is equal to the total number of vehicles $\mathscr{p}_{i,od}^{0}$ that entered cell i at time step 0.

$$x_i^{0,od} = \mathscr{p}_{i,od}^{0} \qquad \forall i \in C, (o,d) \in C_{OD} \tag{6-20}$$

$$\overline{x}_i^0 = \sum_{(o,d) \in C_{OD}} \mathscr{p}_{i,od}^{0} \qquad \forall i \in C \tag{6-21}$$

Next, we update the value of flow variables. Let us define new variables $\mathcal{T}_{ij}$ and $\mathscr{f}_{ij}$ for all network links $i \in C, j \in S(i)$ to track the time that not all vehicles can exit a cell and the fraction of flow that can exit the cell in this time in (6-22) and (6-23), respectively.

$$\mathcal{T}_{ij} = 0 \qquad \forall i \in C, j \in S(i) \tag{6-22}$$

$$\mathscr{f}_{ij} = 0 \qquad \forall i \in C, j \in S(i) \tag{6-23}$$

The value of $\overline{y}_{ij}^t$ shows the total flow moving from cell $i \in C$ to cell $j \in S(i)$ at time step t, and its value is updated using equation (6-24) for all cells except for diverge and intersection cells. This equation is defined by Daganzo (1995).

$$\overline{y}_{ij}^t = \min \{\overline{x}_i^t, F_i, F_j, M_j - \overline{x}_j^t\} \qquad \forall\, t \in T, (i,j) \in E_O \tag{6-24}$$

If $\overline{y}_{ij}^t > 0$, we use equations (6-25), (6-26), and (6-27) to determine flow for each OD pair.

Equation (6-25) finds the first time step $\mathcal{T}_{ij}$ that not all vehicles can exit. In (6-26), the fraction of vehicles $\mathcal{f}_{ij}$ that can leave the cell in $\mathcal{T}_{ij}$ is found. Using this fraction value, the outflow for each OD pair is found in (6-27).

$$\mathcal{T}_{ij} = \max_{\tau}\{\sum_{t'=0}^{\tau}\sum_{(o,d)\in C_{OD}} \mathcal{p}_{i,od}^{t'} > \overline{y}_{ij}^{t}\} \qquad \forall t \in T, i \in C, j \in S(i), \tau = 0, \dots, t-1 \tag{6-25}$$

$$\mathcal{f}_{ij} = \overline{y}_{ij}^{t} - \sum_{t'=0}^{\mathcal{T}_{ij}-1} \sum_{(o,d)\in C_{OD}} \mathcal{p}_{i,od}^{t'} \Big/ \sum_{(o,d)\in C_{OD}} \mathcal{p}_{i,od}^{\mathcal{T}_{ij}} \qquad \forall t \in T, i \in C, j \in S(i) \tag{6-26}$$

$$y_{ij}^{t,od} = \mathcal{f}_{ij} \sum_{(o,d)\in C_{OD}} \mathcal{p}_{i,od}^{\mathcal{T}_{ij}} + \sum_{(o,d)\in C_{OD}} \sum_{t'=0}^{\mathcal{T}_{ij}-1} \mathcal{p}_{i,od}^{t'} \qquad \forall t \in T, i \in C_O, (o,d) \in C_{OD} \tag{6-27}$$

For intersection cells, the value of $\overline{y}_{ij}^{t}$ is found using equation (6-28). The difference with other cells is the signal can influence the value of flow to become zero. The flow for each OD pair is found using equations (6-25), (6-26), and (6-27).

$$\overline{y}_{ij}^{t} = \min\{\overline{x}_i^t, g_i^t F_i, F_j, M_j - \overline{x}_j^t\} \qquad \forall\, t \in T, (i,j) \in E_O \tag{6-28}$$

The update of outflow of diverge cells is also different. An axillary variable z_{ij} for diverge links $i \in C_D, j \in S(i)$ is defined and set equal to $x_i^{t,od} R_{ij}^{t,od}$ as shown in equation (6-29). The value of $\overline{y}_{ij}^{t}$ is determined using equation (6-30) explained by Ukkusuri et al. (2012).

$$z_{ij} = \sum_{(o,d)\in C_{OD}} R_{ij}^{t,od}\, x_i^{t,od} \qquad \forall\, i \in C_D, j \in S(i) \tag{6-29}$$

$$\overline{y}_{ij}^{t} = \min\{z_{ij}, F_i, F_j\,, M_j - \overline{x}_j^{t}\}\min\{1, \frac{F_i}{\sum_{j\in S(i)} \min\{z_{ij}, F_i, F_j\,, M_j - \overline{x}_j^{t}\} + \mu}\} \qquad \forall t \in T, i \in C_D, j \in S(i) \qquad (6\text{-}30)$$

$$\mathcal{T}_{ij} = \max_{\tau}\{\textstyle\sum_{t'=0}^{\tau} \sum_{(o,d)\in C_{OD}} R_{ij}^{t,od} \mathscr{p}_{i,od}^{t'} > \overline{y}_{ij}^{t}\} \qquad \forall\, t \in T, i \in C, j \in S(i), \tau = 0, \dots, t-1 \qquad (6\text{-}31)$$

$$\mathscr{f}_{ij} = {\overline{y}_{ij}^{t} - \textstyle\sum_{t'=0}^{\mathcal{T}_{ij}-1} \sum_{(o,d)\in C_{OD}} R_{ij}^{t,od} \mathscr{p}_{i,od}^{t'}} \Big/ {\textstyle\sum_{(o,d)\in C_{OD}} R_{ij}^{t,od} \mathscr{p}_{i,od}^{\mathcal{T}_{ij}}} \qquad \forall\, t \in T, i \in C, j \in S(i) \qquad (6\text{-}32)$$

$$y_{ij}^{t,od} = \mathscr{f}_{ij} R_{ij}^{t,od} \mathscr{p}_{i,od}^{\mathcal{T}_{ij}} + \sum_{(o,d)\in C_{OD}} \sum_{t'=0}^{\mathcal{T}_{ij}-1} R_{ij}^{t,od} \mathscr{p}_{i,od}^{t'} \qquad \forall\, t \in T, i \in C_O, (o,d) \in C_{OD} \qquad (6\text{-}33)$$

For source cell $i \in C_O$ and OD pair $(o,d) \in C_{OD}$, we update the value of $\mathscr{p}_{i,od}^{t'}$ using equations (6-34)-(6-37).

$$\mathscr{p}_{i,od}^{t'} = 0 \qquad \forall\, t' = 0, \dots, \mathcal{T}_{ij} - 1, i \in C_O, j \in S(i), (o,d) \in C_{OD} \qquad (6\text{-}34)$$

$$\mathscr{p}_{i,od}^{\mathcal{T}_{ij}} = \mathscr{p}_{i,od}^{\mathcal{T}_{ij}}(1 - \mathscr{f}_{ij}) \qquad \forall i \in C_O, j \in S(i), (o,d) \in C_{OD} \qquad (6\text{-}35)$$

$$\mathscr{p}_{j,od}^{t} = \mathscr{p}_{j,od}^{t} + D_i^{t,od} \qquad \forall\, t \in T, i \in C_O, i = o, (o,d) \in C_{OD} \qquad (6\text{-}36)$$

$$\mathscr{p}_{j,od}^{t} = \mathscr{p}_{j,od}^{t} + y_{ij}^{t,od} \qquad \forall\, t \in T, i \in C_O, i \neq o, (o,d) \in C_{OD} \qquad (6\text{-}37)$$

For each cell $i \in C$ and OD pair $(o,d) \in C_{OD}$ except for the source and diverge cells, we update the value of $\mathscr{p}_{i,od}^{t'}$ using equations (6-38)-(6-40).

$$\mathscr{p}_{i,od}^{t'} = 0 \qquad \forall\, t' = 0, \dots, \mathcal{T}_{ij} - 1, i \in C, j \in S(i), (o,d) \in C_{OD} \tag{6-38}$$

$$\mathscr{p}_{i,od}^{\mathcal{T}_{ij}} = \mathscr{p}_{i,od}^{\mathcal{T}_{ij}}(1 - \mathscr{f}_{ij}) \qquad \forall i \in C, j \in S(i), (o,d) \in C_{OD} \tag{6-39}$$

$$\mathscr{p}_{j,od}^{t} = \mathscr{p}_{j,od}^{t} + y_{ij}^{t,od} \qquad \forall\, t \in T, i \in C, j \in S(i), (o,d) \in C_{OD} \tag{6-40}$$

Equations (6-41)-(6-43) show the update procedure for diverge cells.

$$\mathscr{p}_{i,od}^{t'} = \mathscr{p}_{i,od}^{t'} - R_{ij}^{t,od}\mathscr{p}_{i,od}^{t'} \qquad \forall\, t' = 0, \dots, \mathcal{T}_{ij} - 1, i \in C, j \in S(i), (o,d) \in C_{OD} \tag{6-41}$$

$$\mathscr{p}_{i,od}^{\mathcal{T}_{ij}} = \mathscr{p}_{i,od}^{\mathcal{T}_{ij}}(1 - R_{ij}^{t,od}\mathscr{f}_{ij}) \qquad \forall i \in C, j \in S(i), (o,d) \in C_{OD} \tag{6-42}$$

$$\mathscr{p}_{j,od}^{t} = \mathscr{p}_{j,od}^{t} + y_{ij}^{t,od} \qquad \forall\, t \in T, i \in C, j \in S(i), (o,d) \in C_{OD} \tag{6-43}$$

Finally, we update the value of $x_i^{t+1,od}$ by summing $\mathscr{p}_{i,od}^{t',}$ over all entry times in (6-44) and find $\overline{x}_i^{t+1}$using the updated values of $\mathscr{p}_{i,od}^{t'}$ in (6-45).

$$x_i^{t+1,od} = \sum_{t'=0}^{t} \mathscr{p}_{i,od}^{t'} \qquad \forall t \in T, i \in C, j \in S(i), (o,d) \in C_{OD} \tag{6-44}$$

$$\overline{x}_i^{t+1} = \sum_{(o,d)\in C_{OD}} \sum_{t'=0}^{t} \mathscr{p}_{i,od}^{t'} \qquad \forall t \in T, i \in C, j \in S(i) \tag{6-45}$$

6.4. Overall DOCA-SODTA framework

The DOCA-SODTA approach is summarized in the following steps and shown in Figure 6-2.

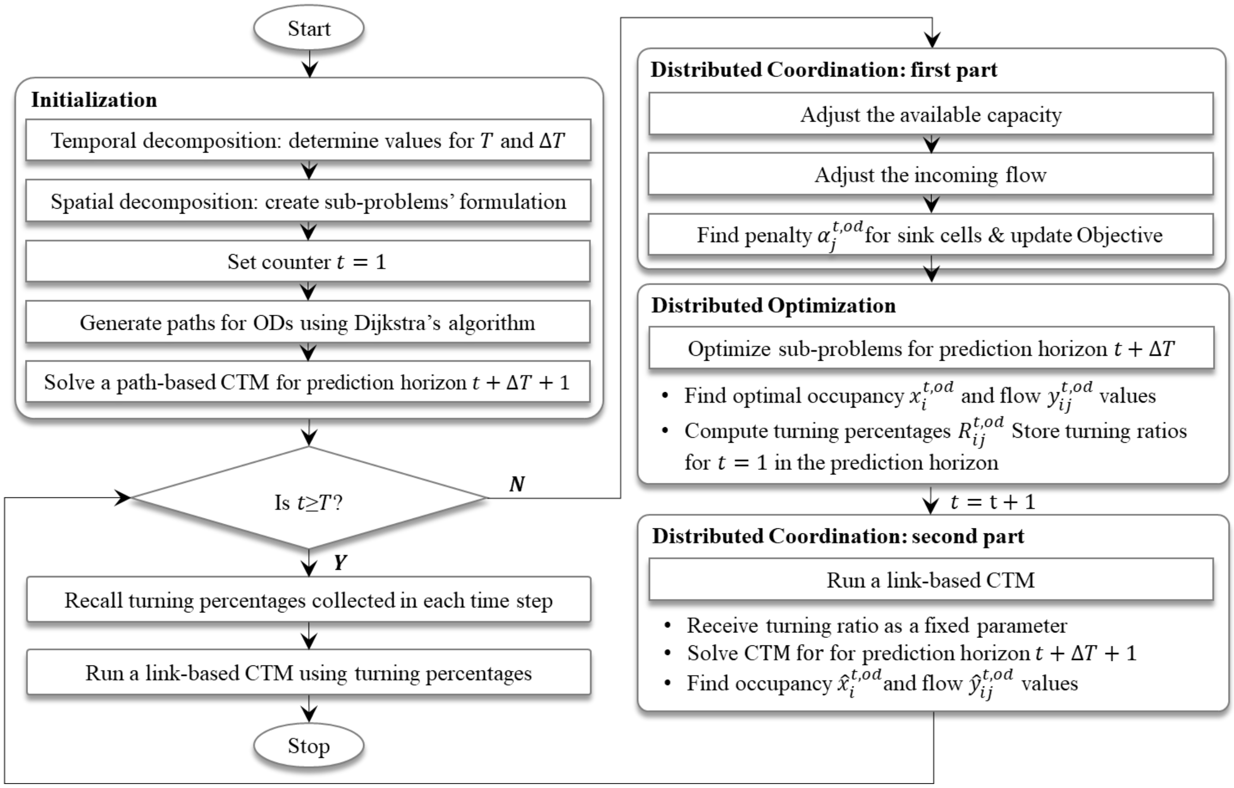

Start
Initialization
Temporal decomposition: determine values for T and ΔT
Spatial decomposition: create sub-problems' formulation
Set counter $t = 1$
Generate paths for ODs using Dijkstra's algorithm
Solve a path-based CTM for prediction horizon $t + \Delta T + 1$
Is $t \geq T$?
N
Y
Recall turning percentages collected in each time step
Run a link-based CTM using turning percentages
Stop
Distributed Coordination: first part
Adjust the available capacity
Adjust the incoming flow
Find penalty $\alpha_j^{t,od}$ for sink cells & update Objective
Distributed Optimization
Optimize sub-problems for prediction horizon $t + \Delta T$
• Find optimal occupancy $x_i^{t,od}$ and flow $y_{ij}^{t,od}$ values
• Compute turning percentages $R_{ij}^{t,od}$ Store turning ratios for $t = 1$ in the prediction horizon
$t = t + 1$
Distributed Coordination: second part
Run a link-based CTM
• Receive turning ratio as a fixed parameter
• Solve CTM for for prediction horizon $t + \Delta T + 1$
• Find occupancy $\hat{x}_i^{t,od}$ and flow $\hat{y}_{ij}^{t,od}$ values

Figure 6-2 The DOCA-SODTA framework

1. Initialization › the study period T and prediction horizon ΔT are initialized. Then, the approach adjusts sets and parameters for each sub-problem and sets counter $t = 1$. It next generates paths using the Dijkstra algorithm and initializes occupancy and flow variables by running a path-based simulation (Ukkusuri et al., 2012) for prediction horizon $t + \Delta T + 1$. Next, the algorithm goes to Step 2.

2. Termination Criteria › if $t \geq T$, the algorithm runs a link-based CTM using stored turning percentages and is terminated. Otherwise, the algorithm goes to Step 3.

3. Distributed Coordination: first part › Constraints (6-2) and (6-3) and objective function (6-4) in the formulation of each sub-problem are updated using occupancy and flow values given from either the path-based CTM (when $t = 1$) or the link-based CTM (when $t \neq 1$). Next, the algorithm goes to Step 4.

4. Distributed Optimization › The sub-problems are optimized simultaneously for prediction horizon $t + \Delta T$, and turning percentages are computed using Equation (6-1). The turning percentages of the first time step t of the current horizon are stored, and the algorithm goes to Step 5.

5. Distributed Coordination: second part › The algorithm sets $t = t + 1$. The link-based CTM simulation is run for prediction horizon $t + \Delta T + 1$ using turning percentages from the distributed optimization step. The algorithm next goes to Step 2.

a. Benchmark Solutions

To evaluate the solutions found by DOCA-SODTA, the SODTA optimization problem is solved centrally using CPLEX. CPLEX was run on a Linux-based cluster with 150.0 GB of memory.

6.5. Test Network

The approach was first tested on a portion of the downtown Springfield network in Illinois, as shown in Figure 6-3. This network consists of 20 (4×5) intersections with one-way and two-way streets. All intersections are signalized with pre-timed signal timing parameters. We considered 15 OD pairs, and the network was loaded with three fixed and three time-variant demand profiles, see Table 6-3. The total study period was 60 minutes which included 30 minutes of loading the network. Six test scenarios correspond to six demand profiles in Table 6-3, respectively. Another test network with 40 (4×10) intersections and 25 OD pairs is created by duplicating the network of 20 intersections to evaluate the performance of the approach when the number of decision variables and constraints increase. Table 6-2 shows three demand profiles tested for this network.

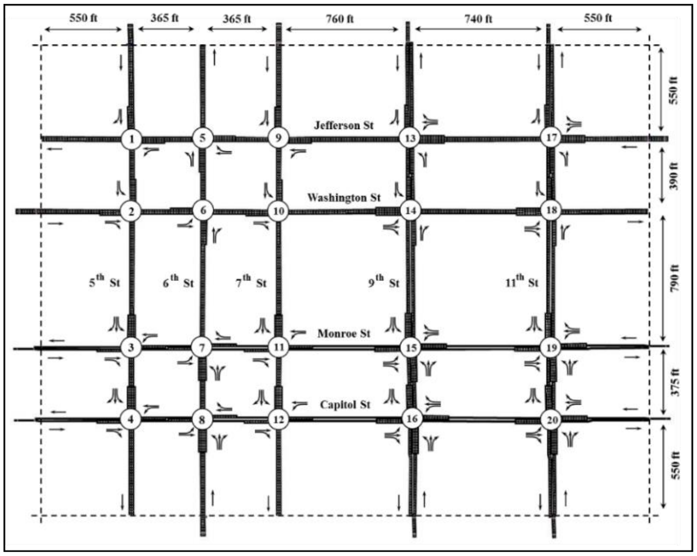
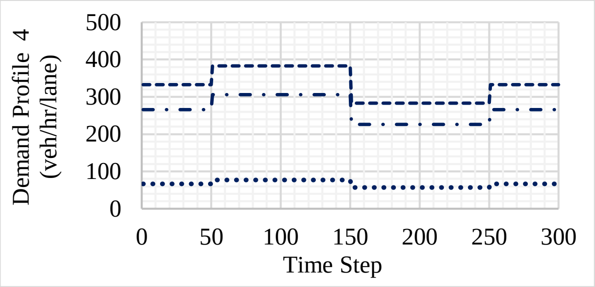
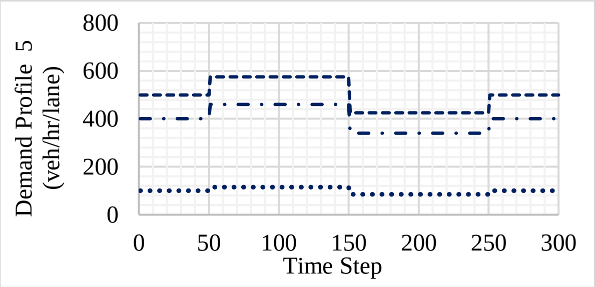
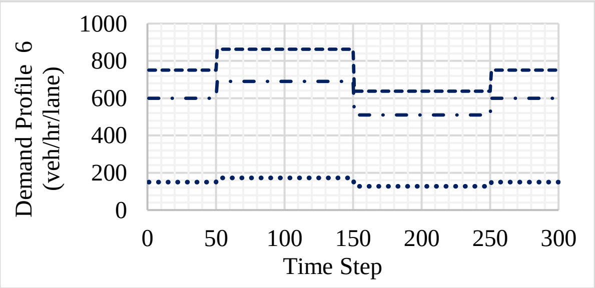
The case study of 20 intersections – Springfield, IL	**CTM characteristics**	
	The duration of each time step (sec)	6
	The number of cells in each link	2,3, 4
	The total number of cells	316
	The total number of links	387
	Free-flow speed (mph)	25
	Cell length (ft)	220
	The capacity of cells except for source and sink cells (veh)	9, 24, 36
	Saturation flow rate except for source and sink cells (veh/ts/ln)	3,6, 9
	Capacity and Saturation flow rate for source and sink cells	1000
	sec: second, mph: mile per hour, ft: feet, veh: vehicle, ts: time step	

Figure 6-3 The case study of 20 intersections and its CTM characteristics

Table 6-2 Demand profiles for the test network of 40 intersections

Network of 40 intersections with 25 ODs							
OD/Demand (veh/hr/ln)	Under-saturated	Semi-saturated	Over-saturated	OD/Demand (veh/hr/ln)	Under-saturated	Semi-saturated	Over-saturated
1	333	500	750	14	333	500	750
2	267	400	600	15	267	400	600
3	27	40	60	16	27	40	60
4	40	60	90	17	40	60	90
5	267	400	600	18	267	400	600
6	67	100	150	19	67	100	150
7	320	480	720	20	320	480	720
8	200	300	450	21	200	300	450
9	120	180	270	22	120	180	270
10	200	300	450	23	200	300	450
11	120	180	270	24	120	180	270
12	200	300	450	25	320	480	720
13	120	180	270				

Table 6-3 Demand profiles for network with 20 intersections

Number	OD	Constant Demand (veh/hr/lane)			Variable Demand (veh/hr/lane)
		Profile 1	Profile 2	Profile 3	------ O-D pairs 1,2,3,6,7,8, and 9 •••••• O-D pairs 4 and 11 – · – · O-D pairs 5 and 10
1	B-L	333	500	750	
2	C-K	333	500	750	
3	D-J	333	500	750	
4	F-O	67	100	150	
5	F-Q	266	400	600	
6	H-O	333	500	750	
7	I-N	333	500	750	
8	J-D	333	500	750	
9	K-C	333	500	750	
10	M-A	266	400	600	
11	M-H	67	100	150	
12	N-I	333	500	750	
13	O-H	333	500	750	
14	P-G	333	500	750	
15	R-E	333	500	750	

6.6. Results

6.6.1. General Results and FIFO Implementation

Table 6-4 shows the network performance measures achieved by applying DOCA and the benchmark algorithm to solve SODTA on the test network of 20 intersections using six demand patterns. The proposed approach is a heuristic, so it is not guaranteed that the approach provides a tight optimality gap in general. However, our numerical results show that a gap of at most 2.39% was achieved in the evaluated scenarios. The difference between the number of completed trips achieved by the two solutions was less than 0.05% because vehicles had 30 minutes to exit the network. DOCA's solutions had shorter congestion durations than those of the optimal solutions. The reason is that the optimal solutions of the benchmark algorithm suffer from the flow holding back issue while DOCA's solutions avoid it. In other words, the flow may be held as long as it does not increase the network-level travel time (i.e., the objective function of the problem). As such, the flow holding back phenomena may create unwanted congestions that do not change the total travel time. Note that DOCA found its near-optimal solutions in real-time and reduced the computation times in different scenarios between 99.63% and 99.71%. These reductions are significant and highlight the capability of the proposed distributed solution technique to obtain high-quality solutions very efficiently. The results of implementing FIFO in CTM are also included in this table. Considering FIFO has led to more travel times, fewer completed trips, higher congestion, and higher CPU time compared to the original DOCA.

Table 6-4 Network performance measures for scenarios 1-6 in the Springfield network

Performance Measures	Approach/Gap	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5	Scenario 6
Objective function: Total Travel Time (hr)	Optimal Solution	187.1	291.30	489.6	187.5	291.6	499.6
	DOCA	187.5	292.7	499.9	188.1	292.8	511.8
	Gap (%)	0.26	0.47	2.05	0.30	0.43	2.39
	DOCA-FIFO	190.9	294.6	539.6	192.2	294.4	607.1
Total Completed Trips (veh)	Optimal Solution	4590.0	6750.0	9990.0	4590.0	6750.0	9990.0
	DOCA	4590.0	6750.0	9987.1	4590.0	6750.0	9984.9
	Gap (%)	0.00	0.00	0.03	0.00	0.00	0.05
	DOCA-FIFO	4590.0	6749.9	9950.6	4590.0	6740.0	9935.4
Total Congestion Duration (hr)	Optimal Solution	19.7	25.4	30.1	19.6	24.2	33.6
	DOCA	19.2	23.1	27.9	20.0	21.9	34.4
	Gap (%)	-2.82	-9.83	-7.71	2.08	-10.41	2.30
	DOCA-FIFO	118.5	108.9	178.9	120.6	102.3	210.6
Total CPU-Time (hr)	Optimal Solution	70.5	73.6	90.6	75.2	79.0	75.1
	DOCA	0.24	0.24	0.26	0.25	0.26	0.28
	Gap (%)	-99.7	-99.7	-99.7	-99.7	-99.7	-99.6
	DOCA-FIFO	0.48	0.60	0.73	0.58	0.76	0.81

Figure 6-4. A to F show the total travel times found by DOCA and the benchmark algorithm for scenarios 1 to 6, respectively. The maximum observed difference was 1.00% across all time steps and scenarios. This small difference indicates that the solutions of DOCA are very close to the optimal solutions generated by the benchmark algorithm at different time steps of the study period.

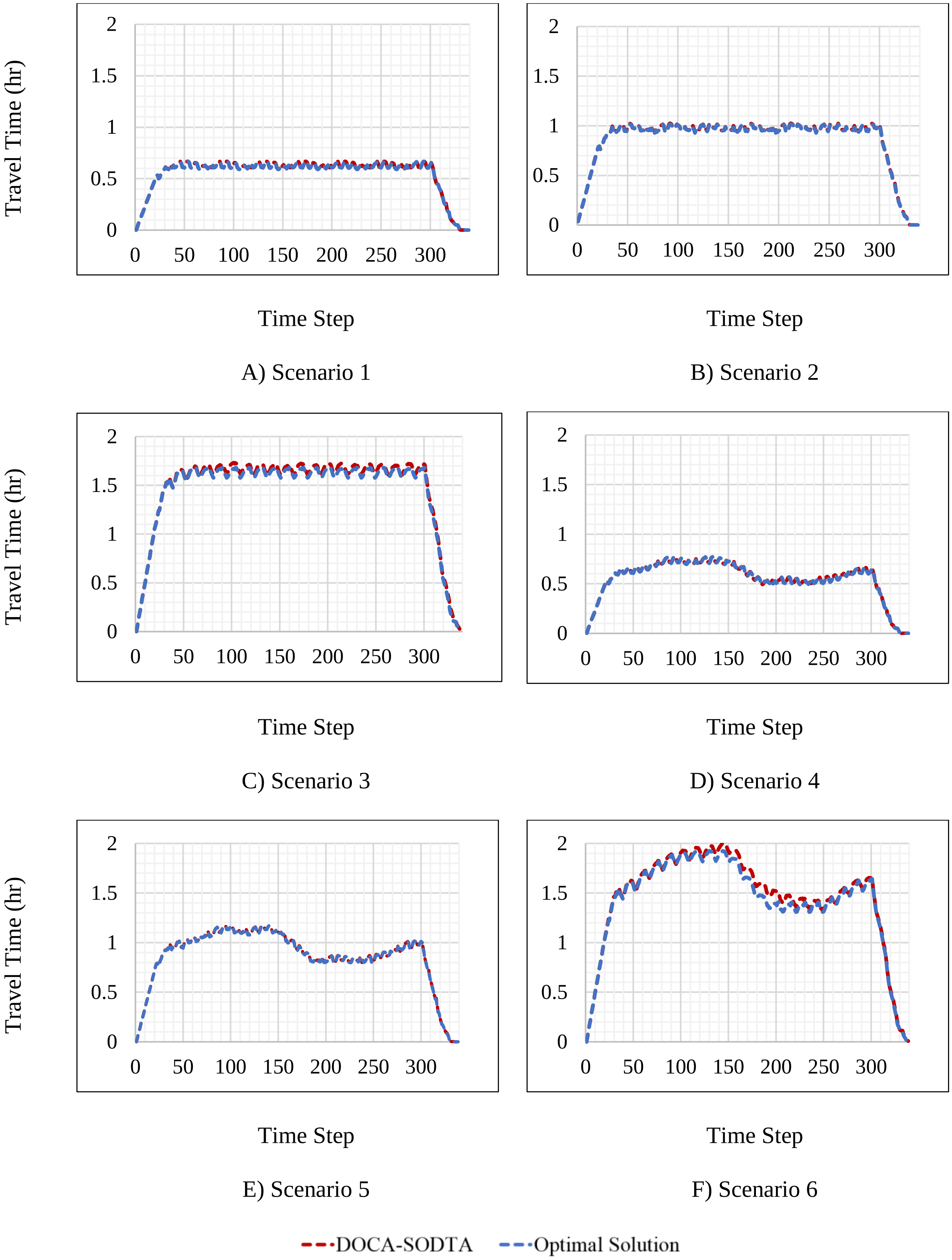
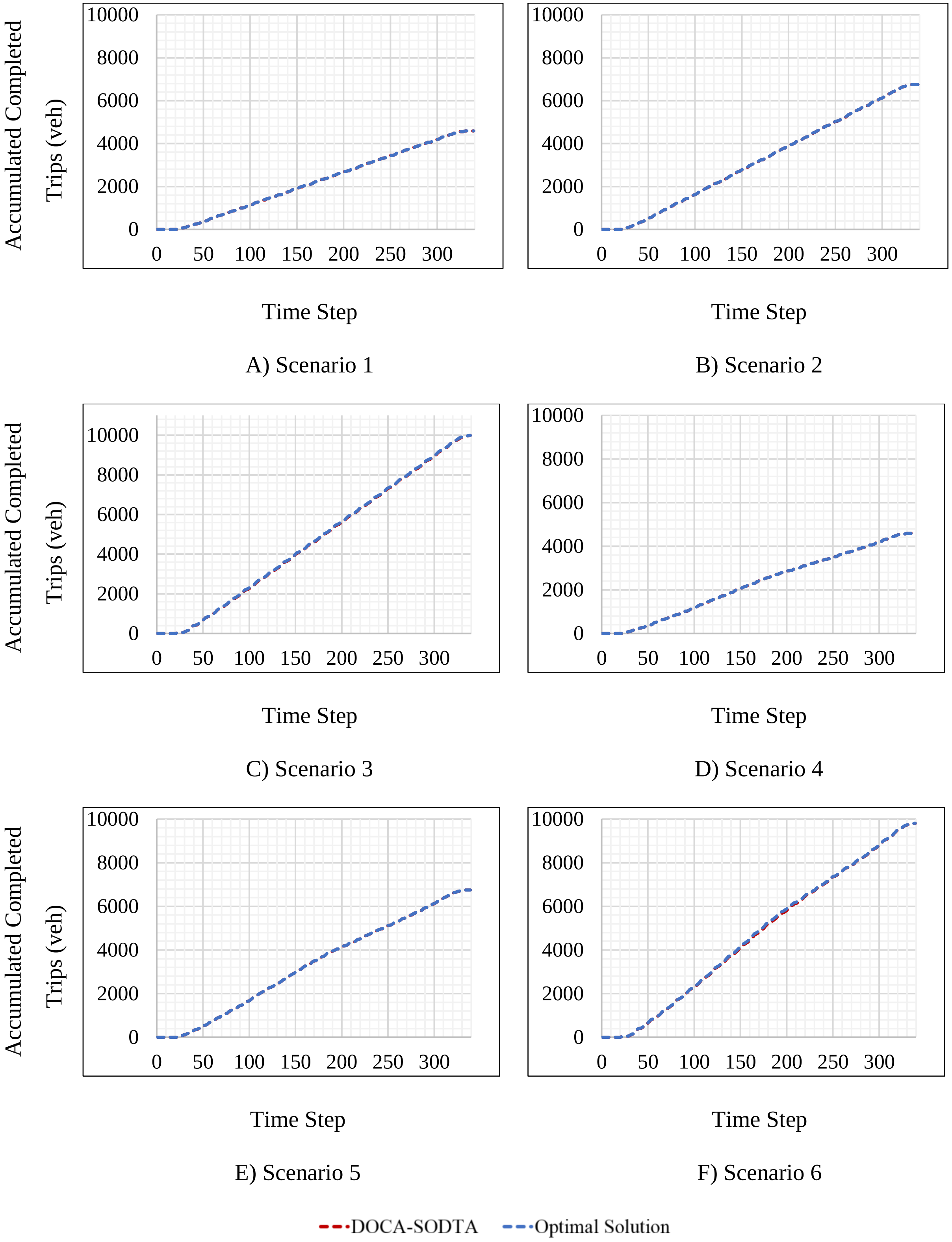


Figure 6-4 Travel time over time steps for scenarios 1-6 in the Springfield network

Figure 6-5. A to F show the accumulative number of completed trips in the network over the duration of the study period for scenarios 1 to 6, respectively. Both DOCA and the benchmark algorithm found solutions with relatively the same accumulative number of completed trips (the maximum difference was 0.44%). This small difference shows that not only the total number of completed trips were the same, but DOCA also processed the same number of vehicles compared to the benchmark algorithm over the duration of the study period.



A) Scenario 1

B) Scenario 2

C) Scenario 3

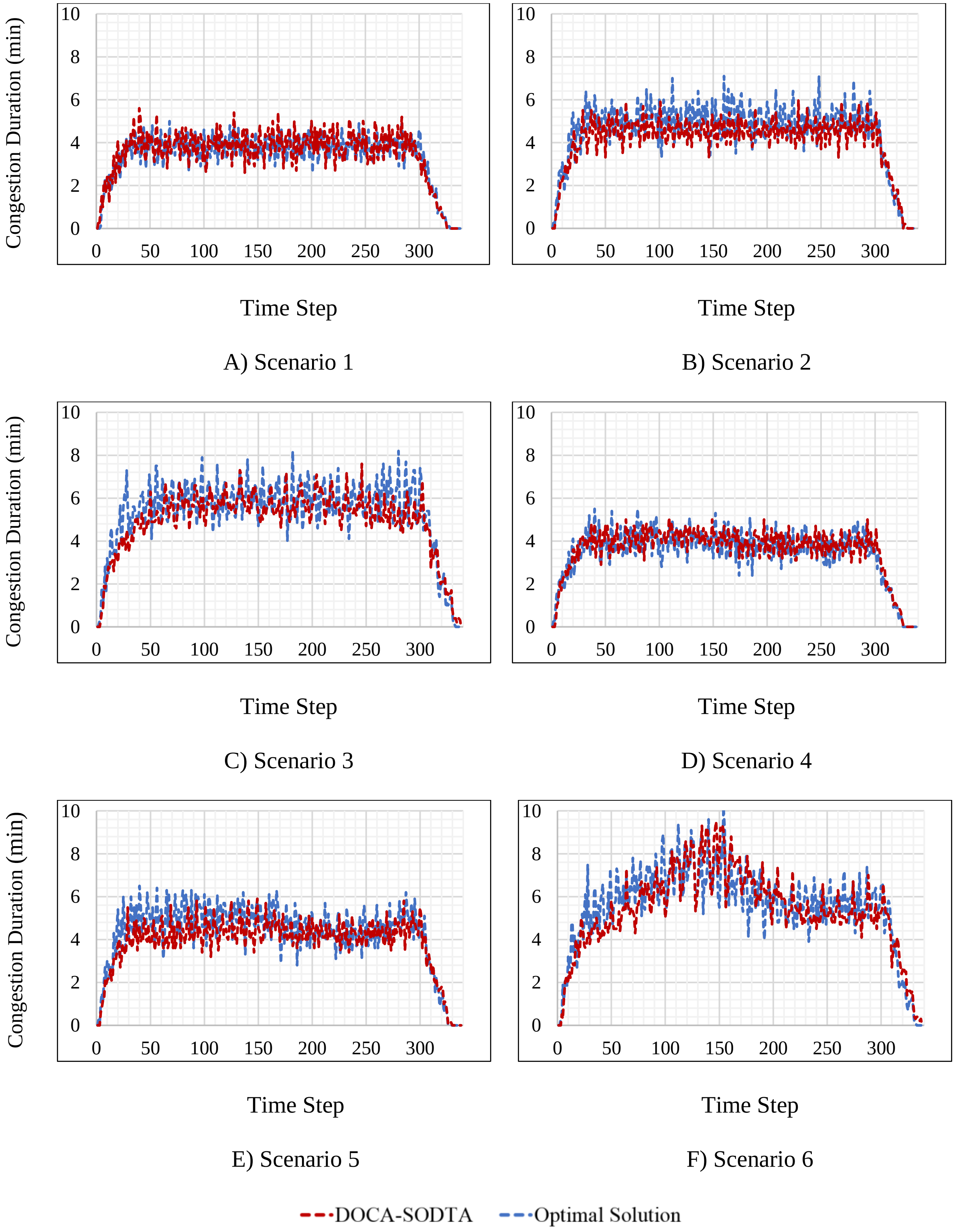
D) Scenario 4

E) Scenario 5

F) Scenario 6

Figure 6-5 Accumulated completed trips over time steps for scenarios 1-6 in Springfield Network

Figure 6-6. A to F show the congestion duration for all time steps for scenarios 1 to 6, respectively. The congestion duration found by the solutions of DOCA was always within a 1% gap from that of the benchmark solutions at each time step. The portion of time with queued vehicles varies over time steps. This congestion duration is less in the initial time steps because the network has not got congested yet. Also, the congestion time does not decrease substantially until time step 300, when the network starts to be unloaded. DOCA has the same congestion duration patterns as the optimal solution in scenarios 1 through 6. As it appears in Figure 6-6. A, B, C, and E, the optimal values are mostly higher than DOCA, leading to less total congestion duration in DOCA. The reason can be the holding back issue in the optimal solutions that keeps vehicles at some cells while there is enough capacity in the downstream cells with no signal interruption.



A) Scenario 1

B) Scenario 2

C) Scenario 3

D) Scenario 4

E) Scenario 5

F) Scenario 6

Figure 6-6 Congestion duration over time steps for scenarios 1-6 in Springfield network

6.6.2. Non-holding Back Solution

The solutions generated by DOCA are feasible. These solutions are found by giving the sub-problem level optimal solutions (turning ratios) to the CTM simulation. Therefore, they do not have the flow holding-back issue since the simulation does not have a way to hold the flow. Figure 6-7. A to D show the accumulative departure rates in scenario 1 for 4 source cells. The vehicles were sent into the network at a constant rate, which is less than the saturation flow rate at each time step. Scenario 1 is selected as it has an under-saturated demand level so that the incoming flow to the network is not blocked by other traffic. Therefore, if the incoming flow does not get into the network, it is held back by the optimization process. As the figures show, the benchmark algorithm has the flow holding back issue, while the solution of DOCA does not have this problem. Note that this methodology guarantees non-holding back solutions due to the utilization of simulation. However, this approach may not be applicable to other formulations and methodologies.

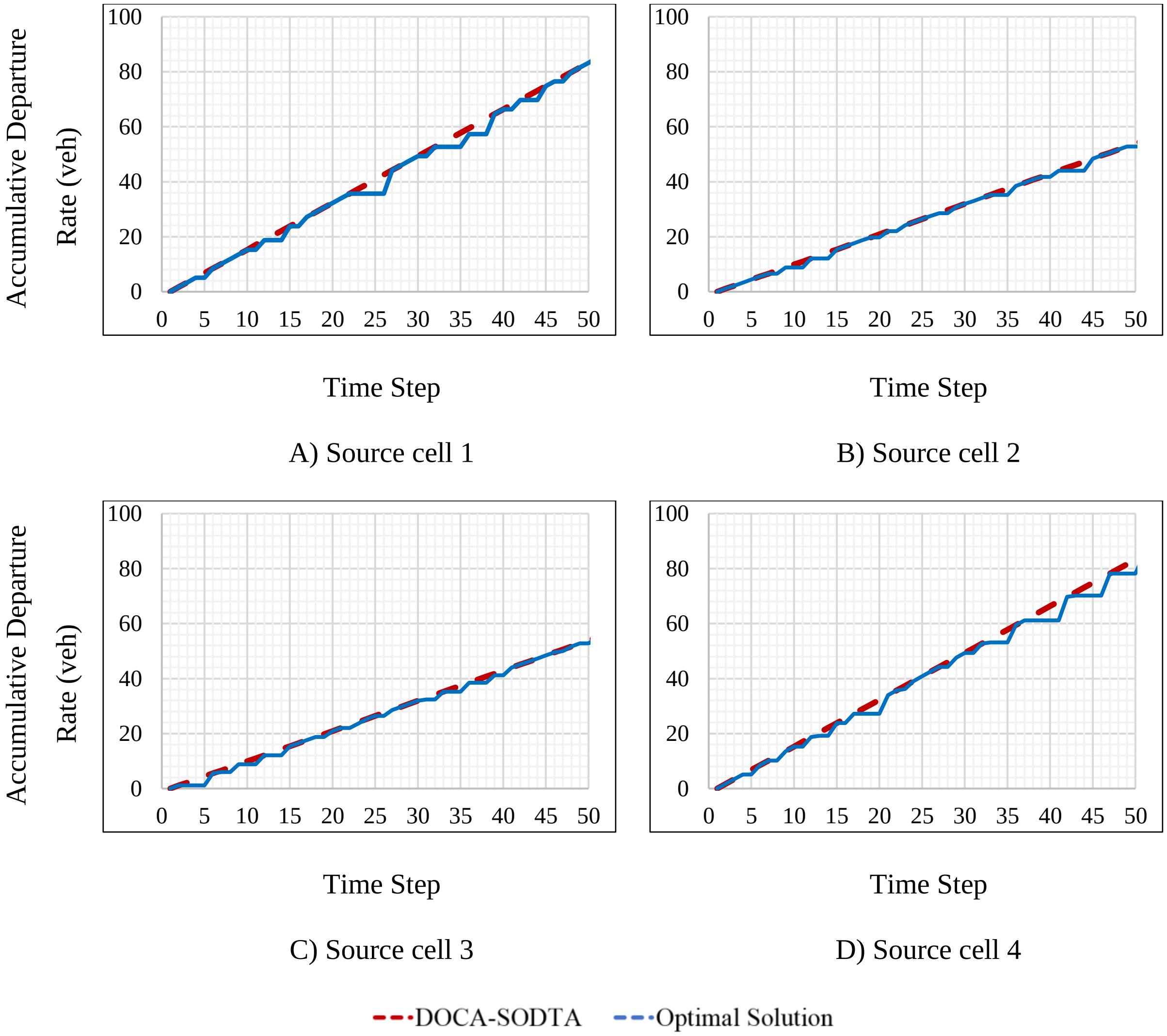


Figure 6-7 Accumulative departure rates in scenario 1 for source Cells 1, 67, 89 and 182

6.6.3. Scalability of DOCA

Figure 6-8 shows the required CPU time to find the optimal solution for each time step in all test scenarios in the Springfield network. DOCA took at most 3.6 seconds to find a near-optimal solution, which is considerably shorter than the duration of each time step (six seconds). As a result, by assuming a significant safety margin for communications between the sub-problems, the solutions are generated in real-time. Even though the solutions for all tested scenarios are found in real-time, the approach does not guarantee real-time solutions in general. Note that the proposed methodology was run on a Linux-based cluster with 16.0 GB of memory. The multi-thread platform was used in the distributed optimization.

We used 21 processors and one node for parallelization. Increasing the number of intersections will not increase the computational time for optimizing subproblems, but more processors will be required for parallelization. Moreover, the computation time of simulation will also increase by adding more partitions to the network since its complexity is dependent on network size. One of the main reasons for lower runtimes in our approach compared to CPLEX and other heuristics is that sub-problems are far less computationally complex and can be optimized in parallel.

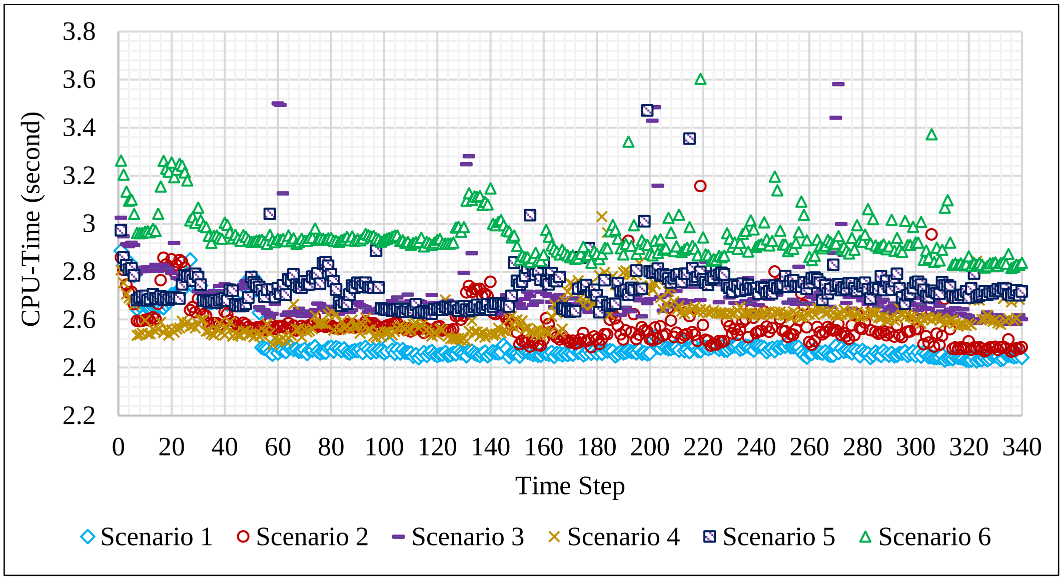


Figure 6-8 CPU-time of DOCA for scenarios 1-6 in the Springfield network

Table 6-5 shows the run-time of each component of DOCA in the Springfield network when the study period was 400 time steps. Adding up the run times of all steps except for the initialization step and dividing that by the number of time steps gives the average run-time of each time step. The initialization step needed at most 0.10 seconds in scenarios 2,3 and 5. This step is a one-time process and does not repeat in each time step. The distributed optimization step took at most 325.98 seconds in scenario 6. In the distributed coordination step, link-based CTM simulation took at most 8.93 seconds in scenario 6. The time duration of updating Constraints (6-2) and (6-3) and the objective function (6-4) in the formulations of sub-problems were at most 60.10 and 612.00 seconds in scenario 5 for the variable semi-saturated demand profile and scenario 6 for the variable oversaturated demand profile, respectively. Finally, the CPU-time for checking the termination criterion was negligible. Therefore, the maximum run-time per time step was 3.6 seconds, which is less than six seconds, hence the approach is real-time.

Table 6-5 The computational time for different components of DOCA in the Springfield network

Run-time for 400 Time Steps (sec)		Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5	Scenario 6
Initialization		0.09	0.10	0.10	0.08	0.10	0.08
Distributed Optimization		220.66	232.57	276.66	230.06	246.87	325.98
Distributed Coordination	Link-based CTM Simulation	8.07	8.62	8.50	8.39	8.90	8.93
	Update Constraints (6-2) and (6-3) in Sub-problems	56.17	58.01	58.21	57.75	60.10	60.07
	Update Objective Function (6-4)	569.28	577.73	578.03	590.77	612.00	605.97
Termination Criteria		0.00	0.00	0.00	0.00	0.00	0.00

We applied DOCA to a network with 40 intersections with 632 cells, 780 links, 400 time steps, and 25 OD pairs to test its scalability. The number of decision variables in this network was 14,120,000. The number of decision variables in the Springfield network was ~4 million. The existing studies that deal with thousands of intersections and OD pairs either use exit flow functions, point queue models, or link-performance functions. These approaches are aggregated and do not provide the accuracy that is required for traffic operation purposes. In this paper, we use the CTM model, which is more accurate but at the expense of additional complexity. The network of 40 intersections is significantly larger than comparable studies that have used the cell transmission model (Aziz and Ukkusuri, 2012; Chiu and Zheng, 2007; Doan and Ukkusuri, 2015). The studies solve the problem for 5805 to 489,700 decision variables (Li et al., 2003; Zheng and

Chiu, 2011). The algorithm can find solutions very efficiently with at most a 3.13% difference from the optimal solution in real-time in the new network. Table 6-6 shows the number of cells, links, time steps, and decision variables for case studies with 20 and 40 intersections. By increasing the number of intersections, the optimality gap increases by at most 0.51%, and the solutions are generated in real-time.

Table 6-6 DOCA performance by increasing the network size from 20 to 40 intersections

Demand Pattern	Number of intersections	Number of cells	Number of links	Number of time steps	Number of variables	Optimality gap (%)	Runtime (hr)
1. Under-saturated	20	316	387	400	4,218,000	0.26	0.24
	40	632	780	500	14,120,00	0.36	0.28
2. Semi-saturated	20	316	387	400	4,218,000	0.47	0.26
	40	632	780	500	14,120,00	0.60	0.29
3. Over-saturated	20	316	387	400	4,218,000	2.05	0.27
	40	632	780	500	14,120,00	3.13	0.30

Figure 6-9 shows the required time for optimizing sub-problems over all time steps for both networks with 20 and 40 intersections. When the size increases, the number of variables in each sub-problem increases only by the number of OD pairs. The number of cells, links, and time steps of the horizon remain constant. By increasing the number of intersections from 20 to 40, the number of decision variables in each sub-problem has increased by 60%. However, as it is shown in the figure, the run-time to generate solutions at each time step has not increased on average. This happens because the majority of decision variables will take on the value of zero as the flow will be on a small subset of the links.

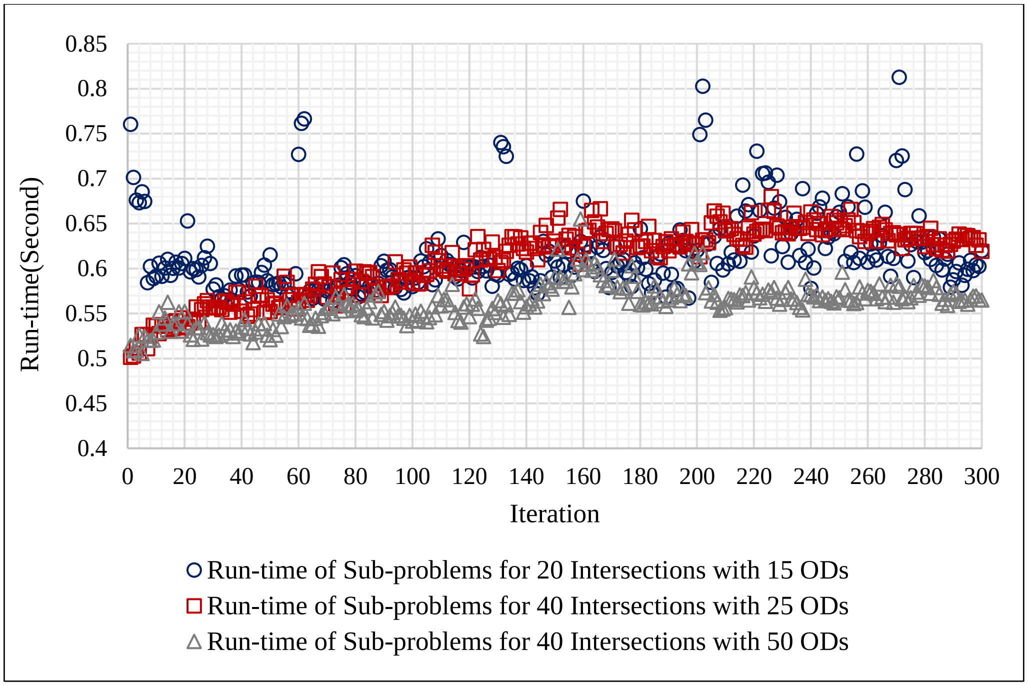


Figure 6-9 Run-time of sub-problems over time steps for networks with 20 and 40 intersections

6.6.4. Comparison with other heuristics

This section presents the results of comparing DOCA to the Method of Successive Averages (MSA) and the Projection Algorithm (PA) used by Doan and Ukkusuri (2015). The largest test network in Doan and Ukkusuri (2015) is the Nguyen-Dupuis network. This network has 57 cells, 63 links, 4 OD pairs, and 100 time steps for the study period, resulting in 48,000 decision variables. The loading horizon contains 80 time steps, and each time step is 1 minute. It is assumed that each OD pair has only three possible paths, but we relax this assumption in DOCA. The Nguyen-Dupuis' network is decomposed into two regions for creating sub-problems in DOCA. The two regions have 30 and 27 cells. They also have 36 and 31 links, respectively, from which four links are shared.

Table 6-7 shows the required number of iterations, computation time, and optimality gap for MSA, PA, and DOCA for ten different scenarios. The scenarios differ in the demand level. MSA and PA are iterative approaches and required 200 and 25 iterations at most to meet the termination criterion, respectively. However, DOCA is not an iterative approach, and it generates solutions for each time step in real-time. The computational time for MSA and PA is reported for the total number of iterations, and for DOCA, it is for the total number of time steps. The computational time of DOCA is significantly faster than MSA and PA. The optimality gap is 0.00% for DOCA, but the relative gap changes from 0.17 to 4.29% in other heuristics.

Table 6-7 Performance comparison of MSA, PA, and DOCA for Nguyen-Dupuis Network

Scenarios	OD demand	Number of iterations			Computational time (minute)			Relative Gap/Optimality Gap		
		MSA	PA	DOCA	MSA	PA	DOCA	MSA	PA	DOCA
1	400	200	9	NA*	80	3.5	0.19	4.29	0.17	0.00
2	500	200	8	NA	80	3.5	0.19	3.9	0.2	0.00
3	600	200	10	NA	82	4	0.19	3.35	0.18	0.00
4	700	200	9	NA	81	4	0.19	2.67	0.2	0.00
5	800	200	13	NA	82	5.5	0.19	1.77	0.19	0.00
6	900	200	20	NA	85	9	0.19	1.68	0.17	0.00
7	1000	200	19	NA	85	9	0.19	1.97	0.19	0.00
8	1100	200	22	NA	85	10	0.20	1.97	0.19	0.00
9	1200	200	25	NA	86	11	0.20	1.53	0.19	0.00
10	1300	200	25	NA	86	11	0.21	1.48	0.2	0.00

*NA: Not Applicable: DOCA is not an iterative approach.

We also plotted the required time to generate solutions for each time step by DOCA in Figure 6-10 for all scenarios tested in the Nguyen-Dupuis network. The run-time varies from 0.08 to 0.19 among different scenarios and time steps.

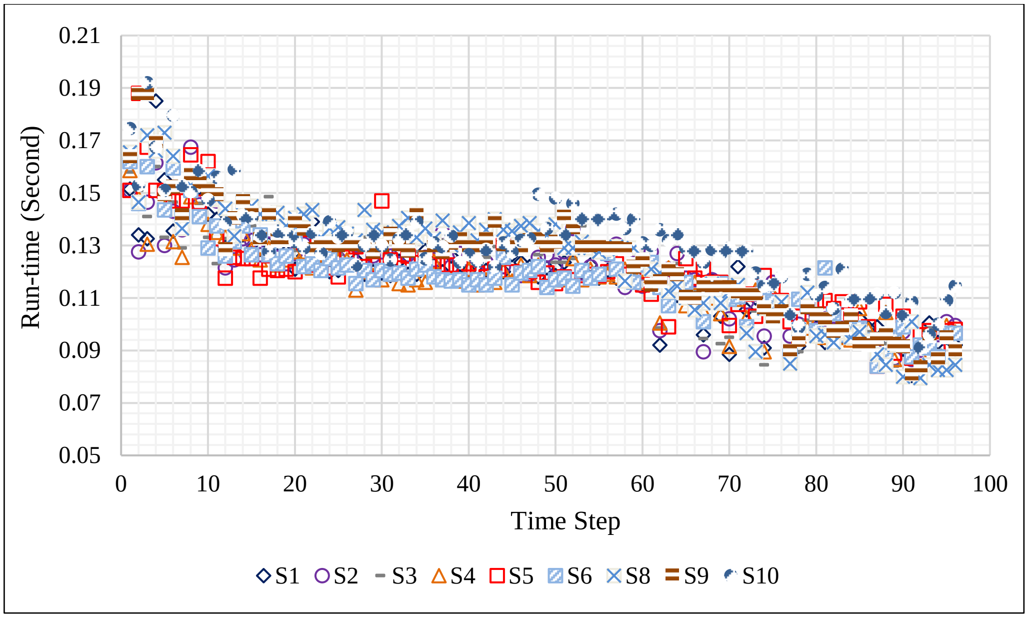


Figure 6-10 Run-time of DOCA over time steps for the Nguyen-Dupuis network and scenarios 1-10

CHAPTER 7. TRAFFIC CONGESTION MANAGEMENT IN URBAN STREET NETWORK BOTTLENECKS

Congestion pricing has been studied widely as an effective approach to decrease traffic congestion and generate revenue for maintaining and building infrastructure. Congestion pricing techniques also encourage people to use public transportation during peak hours. The congestion pricing concept has been introduced by Pigou (1912) and Vickrey (1969) more than 10 decades ago. Congestion pricing has been implemented mainly by tolling through currency or tokens. Tolling through currency or dollars is a form of price regulation that enforces a fixed price for using a network area or specific links. Tolling through tokens or permits is a form of quantity regulation that provides variable distribution and quantity control opportunities. Despite tolling through currency, the maximum number of travelers using tokens is fixed (de Palma et al., 2018).

In tolling through currency, toll levels or prices should be determined constantly to stay adaptive and react to daily changes in demand and supply. Constant changes in prices can impose control costs and public opposition (de Palma et al., 2018). Moreover, there exists asymmetric information between system managers and travelers. The system manager needs detailed information on travelers' demand, the value of time, arrival time, etc. to determine toll levels. However, this information is private, not accessible, or very hard to obtain. The lack of accurate or complete information may lead to non-optimal toll levels that result in the failure of tolling systems and economic loss (Lin et al., 2011a; Sharon et al., 2017).

Tolling through tokens or credits is proposed as an alternative to tolling through dollars. The new alternative has shown to be more equitable with more public acceptance than the price regulation (de Palma et al., 2018), but it has some major drawbacks. In finding the supply of tokens, permit distribution, or their price, tolling through tokens can face a similar problem in tolling through currency in price determination. How to distribute tokens among travelers can have a significant effect on the system and is not straightforward to determine. Travelers can consume allocated credits gradually at any time during a cycle which depends on travelers' budgeting behaviors and mobility needs. Therefore, the behavior of travelers in consuming credits at different times in a cycle can affect traffic congestion adversely. In addition, strategies should be designed for cases when travelers have used their cycle credits but are in need of emergency trips (Kockelman and Kalmanje, 2005; Wu et al., 2012; Yang and Wang, 2011).

In the third group of tolling approaches, travelers are involved in a competitive process for using certain parts of a transportation system with limited roadway capacity. This approach can create market competition by letting drivers determine the toll price or simply personalized pricing. Research has shown economic shocks can be handled faster and more efficiently in market competition compared to markets that are controlled centrally (Basar and Cetin 2017). Researches in this area are in the primary stages of examining public perception and the current studies are simulation-based that lack a decision support system. In this chapter, we design a mathematical formulation and a solution technique for congestion management at bottlenecks in urban street networks with similar concepts to this group of approaches.

7.1. Problem Description

Consider a portion of an urban street network and a study period in which traffic congestion occurs. One example of this area is downtown during rush hours. High traffic congestion in an urban street network can occur due to bottlenecks caused by limited capacity. Our objective is to alleviate traffic congestion and generate revenue using a congestion management approach. Considering the road segments with bottlenecks, we create a set of paths that passes each of these segments. Then, these paths are offered to travelers for submitting their Willingness To Pay

(WTP).

This approach has two main sets of players: a system manager and travelers. The goal of the system manager is to minimize the total travel time of the system and maximize revenue. The system manager informs travelers about options for WTP submission. Travelers present their WTP values to use paths passing network bottlenecks. The system manager selects some travelers with WTP submissions and determines their assignment to paths with bottlenecks. If a traveler is not selected in the selection and allocation process, she/he should use an alternative path for her/his trip.

There is a tradeoff between minimizing the total travel time and maximizing revenue because assigning more travelers, who submit their WTPs, to roads will increase the revenue, but it will also increase the total system travel time. The goal of travelers is to minimize their individual travel times considering the enforced assignment of travelers with WTP submission. The following sections provide the formulation and solution technique for the proposed approach. We used the cell transmission model to model network flows in the proposed formulation. Genetic and projection algorithms are developed to solve the formulation.

7.2. Model Description and Mathematical Formulation

The problem includes two agents: a system manager and travelers. The first part of formulation is an allocation problem in which the system manager selects the best travelers with WTP submission and allocates them to paths with bottlenecks to minimize total travel time and maximize revenue. This part is an integer program with binary decision variables which take the value of one if a traveler is assigned to a path with bottleneck, and zero otherwise. The system manager wants to find the best decisions for the selection and allocation of travelers with WTP submission, providing path options considering network bottlenecks, and receiving WTPs. This decision-making process requires the manager to consider the response of other travelers to the selection and allocation decisions and the decision impact on the total travel time of the system. This consideration leads to a second part of this problem.

The second part of formulation is a dynamic user equilibrium problem that is formulated using variational inequalities. At this part, travelers choose a path to minimize their travel time, and travelers with WTP submissions are routed through paths with bottlenecks considering the results of the assignments from the first part of the formulation. We now present a mathematical formulation for the described approach. Table 7-1 presents the definition of sets, decision variables, and parameters used in this chapter.

Table 7-1 Definition of sets, decision variables, and parameters used in Chapter 7

Sets	
T	The set of all time steps
T'	The set of time steps for departure
C	The set of all network cells
C_O	The set of all source cells
C_S	The set of all sink cells
C_I	The set of all intersection cells
C_D	The set of diverge cells
C_{OD}	The set of Origin-Destination pairs
$P(i)$	The set of all predecessors to cell $i \in C$
$S(i)$	The set of all successors to cell $i \in C$
P	The set of all paths
R	The set of road segments
B	The set of travelers with WTP submission
$L(i)$	The set of paths for traveler $i \in B$ with WTP submission
Q	The set of all paths for traveler $i \in B$ with WTP submission

Table 7-1 (continued)

K	The set of all paths for travelers who do not submit any WTP and travelers who do not win in the selection process
S	The set of feasible solutions
Parameters	
τ	The duration of each time step
F_i	The saturation flow rate at cell $i \in C$
v	The value of time for travelers
M_i	The maximum number of vehicles that cell $i \in C$ can accommodate
g_i^t	A binary parameter to define signal status at intersection cell $i \in C_I$ at time step $t \in T$. Zero and one values indicate red and green signals, respectively.
f_i^t	The variable saturation flow rate of intersection cell $i \in C_I$ at time step $t \in T$ that is equal to $g_i^t F_i$
ω_i	The capacity of road segment $i \in R$
π_i^p	The price that traveler $i \in B$ will pay for using path $p \in L(i)$
$\theta(\boldsymbol{x})$	Total travel time function that is equal to $\sum_{p \in P} \sum_{t \in T} \sum_{i \in C \backslash C_S} v \tau x_i^{t,\mathrm{p}}$
μ	An arbitrary small and positive number
$o^{t,t',p}$	An auxiliary variable for average travel time estimation with time step $t \in T$, departure time $t \in T'$, and path $p \in P$
$O(p)$	Origin of path $p \in P$
$D(p)$	Destination of path $p \in P$
$r^{t,p}$	Departure rate at time step $t \in T$ for path $p \in P$
$\boldsymbol{r}$	A vector of departure rates
$\boldsymbol{r}^*$	A vector of optimal departure rates
$\vartheta^{t,p}$	Average travel time estimation at time step $t \in T$ for path $p \in P$
$\boldsymbol{\vartheta}$	The vector of average travel times
λ	Step size used in projection algorithm
$\boldsymbol{d}$	The vector of demand for travelers who do not participate or win in WTP submission
$\boldsymbol{\eta}$	The vector of path flows that represents any feasible dynamic traffic assignment
$\boldsymbol{\eta}^*$	The vector of path flows that represents user optimal dynamic traffic assignment

Table 7-1 (continued)

$\boldsymbol{\psi}(\boldsymbol{\eta}^*,\boldsymbol{\vartheta})$	A vector that represents the path costs
$d_i^{t,p}(\boldsymbol{b},\boldsymbol{d})$	Demand function defined for source cell $i \in C_O$ time step $t \in T$ in path $p \in P$
$\chi_i^{0,p}$	Initial values for $x_i^{t,p}$
$\mathbb{P}$	The feasible region of the second part of the formulation
$\mathbb{D}$	Feasible region to optimize departure rates
$d^{t,od}$	The demand of travelers who do not submit any WTP and travelers who do not win in the selection process at time step $t \in T$ for origin-destination pair $od \in OD$
$\rho^{t,p}$	The demand of travelers with WTP submission at time step $t \in T$ for path $p \in P$
φ_i	The selection probability of solution i
ϖ_i	The fitness value or objective function of the first part of formulation for solution i
p_m	Mutation percentage
p_c	Crossover percentage
σ_i^p	A binary parameter which is one if path $p \in P$ passes road segment $i \in R$ and zero otherwise
Decision Variables	
$x_i^{t,p}$	The number of vehicles in cell $i \in C$ at time step $t \in T$ in path $p \in P$
$y_{ij}^{t,p}$	The number of vehicles flowing from cell $i \in C$ to downstream cell $j \in S(i)$ at time step $t \in T$ in path $p \in P$
b_i^p	A binary variable which is 1 if traveler $i \in B$ with WTP submission is assigned to path $p \in L(i)$, and zero otherwise
Variables	
$\overline{x}_i^t$	The total number of vehicles in cell $i \in C$ at time step $t \in T$
$\overline{y}_{ij}^t$	The total number of vehicles flowing from cell $i \in C$ to downstream cell $j \in S(i)$ at time step $t \in T$
ς_{ij}^t	The total number of vehicles in diverge cell $i \in C_D$ at time step $t \in T$ heading to successor cell $j \in S(i)$
$\varrho_{ij}^{t,p}$	The number of vehicles in diverge cell $i \in C_D$ at time step $t \in T$ heading to cell $j \in S(i)$ for path $p \in P$
$\boldsymbol{b}$	The vector of solutions for b_i^p for traveler $i \in B$ and path $p \in L(i)$
$\boldsymbol{z}$	A vector of solutions
$\boldsymbol{z}^*$	A vector of optimal solutions

7.2.1. First Part of Formulation

The formulation for finding an optimized assignment of travelers to the network paths with bottlenecks is presented in this section. The objective function (7-1) has two parts: minimization of total travel time and maximization of revenue.

The total travel time is estimated using the solution from the second part of the formulation based on the behavior of other travelers on route selection. Using the second part of the formulation, the number of vehicles $x_i^{t,p}$ in cell $i \in C$ at time step $t \in T$ in path $p \in P$ is obtained. Then, the total travel time is computed by multiplying $x_i^{t,p}$ by the duration of each time step τ and summing the result over all paths $p \in P$, time steps $t \in T$, and cells $i \in C$ expect for sink cells C_s, $\sum_{p \in P} \sum_{t \in T} \sum_{i \in C \setminus C_S} \tau x_i^{t,p}$.

Then, the total travel time is multiplied by the value of time υ and converted to an equivalent monetary value for having the same unit as the revenue expression which is the second part of the objective function. The resultant expression is $\theta(\boldsymbol{x}) = \sum_{p \in P} \sum_{t \in T} \sum_{i \in C \setminus C_S} \upsilon \tau x_i^{t,\mathrm{p}}$. We use $\theta(\boldsymbol{x})$ to denote the total travel time function. Notation $\boldsymbol{x}$ presents a vector of solutions for $x_i^{t,od}$. Note that the value of time can be estimated using methodologies in literature, and the value is mainly determined based on household incomes and compensations (Federal Highway Administration, 2005; Oregon Department of Transportation, 2004).

In the second expression, the revenue is found by multiplying the binary variable b_i^p with the WTP value π_i^p and summing the result over all travelers $i \in B$ with WTS submission and paths $p \in L(i)$, $\sum_{p \in L(i)} \sum_{i \in B} b_i^p \, \pi_i^p$.

$$min\, \theta(\boldsymbol{x}) - \sum_{p \in L(i)} \sum_{i \in B} b_i^p \, \pi_i^p \qquad (7\text{-}1)$$

Constraint (7-2) ensures that each traveler with WTP submission is assigned to no more than one path. The summation of binary variables b_i^p over all paths $L(i)$ is set to be less than or equal to one. This constraint is written for each traveler $i \in B$.

$$\sum_{p \in L(i)} b_i^p \leq 1 \qquad \forall i \in B \qquad (7\text{-}2)$$

Constraint (7-3) limits the number of travelers with WTP submissions assigned to road segments that are open for WTP submissions. The total number of travelers assigned to road segment $i \in R$ is computed by $\sum_{p \in L(i)} \sum_{i \in B} b_i^p \sigma_r^p$. We use σ_r^p to denote a binary parameter which is one if the path passes road segment $r \in R$, and it is zero, otherwise. Hence, if a traveler is assigned to the path that passes a road segment, this traveler is considered in the capacity constraint of the corresponding road segment. Let ω_r denotes the capacity of road segment $r \in R$. The computed expression $\sum_{p \in L(i)} \sum_{i \in B} b_i^p \sigma_r^p$ is set to be less than or equal to ω_i.

$$\sum_{p \in L(i)} \sum_{i \in B} b_i^p \sigma_r^p \leq \omega_r \qquad \forall r \in R \qquad (7\text{-}3)$$

7.2.2. Second Part of Formulation

The second part of the formulation is modeled as a variational inequality in (7-4) since it cannot be modeled as a CTM-based closed-form formulation for capturing UEDTA (Lin et al., 2011a). We use $\boldsymbol{\eta}$ and $\boldsymbol{\eta}^*$to show a vector of feasible path flows for DTA and a vector of optimal path flows for UEDTA, respectively. Notation $\boldsymbol{\psi}$ represents a vector for path costs, and $\boldsymbol{\vartheta}$ shows a vector of average travel times. Inequality (7-4) shows that the cost of an optimal path flow set for UEDTA is less than or equal to the cost of any feasible DTA path flows, i.e., $\boldsymbol{\psi}(\boldsymbol{\eta}^*, \boldsymbol{\vartheta})'\boldsymbol{\eta} \geq \boldsymbol{\psi}(\boldsymbol{\eta}^*, \boldsymbol{\vartheta})'\boldsymbol{\eta}^*$**.**

$$\boldsymbol{\psi}(\boldsymbol{\eta}^*, \boldsymbol{\vartheta})'(\boldsymbol{\eta} - \boldsymbol{\eta}^*) \geq 0 \qquad \boldsymbol{\eta} \in \mathbb{P} \qquad (7\text{-}4)$$

Set $\mathbb{P}$ is the feasible region of the second part of the formulation. The following constraints (7-

5)-(7-20) shape the region $\mathbb{P}$ using variables $x_i^{t,p}$ and $y_{ij}^{t,p}$. Let $x_i^{t,p}$ denote the number of vehicles in cell $i \in C$ at time step $t \in T$ with path $p \in P$ and $y_{ij}^{t,p}$ denote the number of vehicles flowing from cell $i \in C$ to downstream cell $j \in S(i)$ at time step $t \in T$ with path $p \in P$. Constraint (7-5) set the initial state of the network by putting occupancy $x_i^{0,p}$at time step 0 for path $p \in P$ equal to initial values $\chi_i^{0,p}$.

$$x_i^{0,p} = \chi_i^{0,p} \qquad \forall i \in C, p \in P \tag{7-5}$$

Constraints (7-6), (7-7), and (7-8) are conservation flow constraints for all cells except sink and resource cells, resource cells, and sink cells, respectively. The demand parameter $d_i^{t,p}$ in constraint (7-7) is a fuction of travelers with WTP submission and the rest of travellers demand. Let $\boldsymbol{b}$ and $\boldsymbol{d}$ denote vectors of solutions for travelers with WTP submission assignment and other travelers demand, respectively.

$$y_{ki}^{t,p} - y_{ij}^{t,p} = x_i^{t+1,p} - x_i^{t,p} \qquad \forall t \in T, i \in \ C \setminus \{C_S\ ,\ C_O\}, k \in P(i), j \in S(i), p \in P \tag{7-6}$$

$$d_i^{t,p}(\boldsymbol{b}, \boldsymbol{d}) - y_{ij}^{t,p} = x_i^{t+1,p} - x_i^{t,p} \qquad \forall t \in T, i \in C_O, j \in S(i), p \in P \tag{7-7}$$

$$y_{ki}^{t,p} = x_i^{t+1,p} - x_i^{t,p} \qquad \forall t \in T, i \in C_S, k \in P(i), p \in P \tag{7-8}$$

Equation (7-9) determines the aggregate flow for all cells $i \in C$ except diverge cells $i \in C_D$ and intersection cells $i \in C_I$ at time step $t \in T$. The minimum function in this equation follows the CTM rules for determining flow. The aggregated occupancy and flow are noted by $\overline{x}_j^t$ and $\overline{y}_{ij}^t$, respectively. To find the disaggregated flow for each path $p \in P$, the aggregated flow $\overline{y}_{ij}^t$ is distributed among the paths using $\frac{x_i^{t,p}}{\overline{x}_i^t}$ ratio, as shown in equation (7-10). The if-condition in (7-10) can be written as equation (7-11) by adding an arbitrary small and positive number μ to the denominator of $\frac{x_i^{t,p}}{\overline{x}_i^t}$.

$$\overline{y}_{ij}^t = \min\{\overline{x}_i^t, F_i, F_j, M_j - \overline{x}_j^t\} \qquad \forall t \in T, i \in C\backslash\{C_D, C_I\}, j \in S(i) \tag{7-9}$$

$$y_{ij}^{t,p} = \begin{cases} \min\{\overline{x}_i^t, F_i, F_j, M_j - \overline{x}_j^t\} \dfrac{x_i^{t,p}}{\overline{x}_i^t} & \overline{x}_i^t > 0 \\ 0 & \overline{x}_i^t \leq 0 \end{cases} \qquad \forall t \in T, i \in C\backslash\{C_D, C_I\}, j \in S(i) \tag{7-10}$$

$$y_{ij}^{t,p} = \min\{\overline{x}_i^t, F_i, F_j, M_j - \overline{x}_j^t\} \times \frac{x_i^{t,p}}{\overline{x}_i^t + \mu} \qquad \forall\, t \in T, i \in C\backslash\{C_D, C_I\}, j \in S(i), p \in P \tag{7-11}$$

The constraints in (7-12), (7-13), and (7-14) find flow for intersection cell $i \in C_I$. We have used $g_i F_i$ instead of F_i in (7-9), (7-10), and (7-11) to model variable saturation flow rates for these cells.

$$\overline{y}_{ij}^t = \min\{\overline{x}_i^t, g_i F_i, F_j, M_j - \overline{x}_j^t\} \qquad \forall t \in T, i \in C_I, j \in S(i) \tag{7-12}$$

$$y_{ij}^{t,p} = \begin{cases} \min\{\overline{x}_i^t, g_i F_i, F_j, M_j - \overline{x}_j^t\} \dfrac{x_i^{t,p}}{\overline{x}_i^t} & \overline{x}_i^t > 0 \\ 0 & \overline{x}_i^t \leq 0 \end{cases} \qquad \forall t \in T, i \in C_I, j \in S(i) \tag{7-13}$$

$$y_{ij}^{t,p} = \min\{\overline{x}_i^t, g_i F_i, F_j, M_j - \overline{x}_j^t\} \times \frac{x_i^{t,p}}{\overline{x}_i^t + \mu} \qquad \forall t \in T, i \in C_I, j \in S(i), p \in P \tag{7-14}$$

Let ς_{ij}^t and $\varrho_{ij}^{t,p}$denote the total number of vehicles in diverge cell $i \in C_D$ at time step $t \in T$ heading to successor cell $j \in S(i)$ and the number of vehicles in diverge cell $i \in C_D$ at time step $t \in T$ heading to cell $j \in S(i)$ for path $p \in P$. Equations (7-15) and (7-16) determine the outflow of diverging cells using two if-conditions. In (7-15), the maximum outflow $\overline{y}_{ij}^{t}$ is the minimum of ς_{ij}^{t}, F_j, and $M_j - \overline{x}_j^{t}$, when $\sum_{j\in S(i)} \min\,(\varsigma_{ij}^{t}, F_j\,, M_j - \overline{x}_j^{t}) \le F_i$. In (7-16), since $\sum_{j\in S(i)} \min\big(\varsigma_{ij}^{t}, F_j\,, M_j - \overline{x}_j^{t}\big) > F_i$, the aggregated flow $\overline{y}_{ij}^{t}$ is limited to saturation flow rate F_i, and saturation flow rate F_i should be shared among all outgoing links from a diverge cell. Equation (7-17) is a compact form of (7-15) and (7-16). Equation (7-18) finds the disaggregated flow $y_{ij}^{t,p}$ by distributing the aggregated flow $y_{ij}^{t,p}$using $\frac{\varrho_{ij}^{t,p}}{\varsigma_{ij}^{t}+\mu}$ ratio.

$$if \sum_{j\in S(i)} min\,(\varsigma_{ij}^{t}, F_j\,, M_j - \overline{x}_j^{t}) \le F_i \qquad \forall t \in T, i \in C_D, j \in S(i)$$
$$\overline{y}_{ij}^{t} = min\,(\varsigma_{ij}^{t}, F_j, M_j - \overline{x}_j^{t}) \tag{7-15}$$

$$if \sum_{j\in S(i)} min\,(\varsigma_{ij}^{t}, F_j\,, M_j - \overline{x}_j^{t}) > F_i \qquad \forall t \in T, i \in C_D, j \in S(i)$$
$$\overline{y}_{ij}^{t} = \frac{min\,(\varsigma_{ij}^{t}, F_j\,, M_j - \overline{x}_j^{t})}{\sum_{j\in S(i)} min\,(\varsigma_{ij}^{t}, F_j\,, M_j - \overline{x}_j^{t})} \times F_i \tag{7-16}$$

$$\overline{y}_{ij}^{t} = min\,(\varsigma_{ij}^{t}, F_j, M_j - \overline{x}_j^{t}) min\{1, \frac{F_i}{\sum_{j\in S(i)} min(\varsigma_{ij}^{t}, F_j, M_j - \overline{x}_j^{t}) + \mu}\} \qquad \forall t \in T, i \in C_D, j \in S(i) \tag{7-17}$$

$$y_{ij}^{t,p} = \overline{y}_{ij}^{t} \times \frac{\varrho_{ij}^{t,p}}{\varsigma_{ij}^{t} + \mu} \qquad \forall t \in T, i \in C_D, j \in S(i), p \in P \tag{7-18}$$

Constraints (7-19) and (7-20) are non-negativity constraints for occupancy $x_i^{t,p}$ and flow $y_{ij}^{t,p}$ variables, respectively.

$$x_i^{t,p} \geq 0 \qquad \forall t \in T, i \in C, p \in P \tag{7-19}$$

$$y_{ij}^{t,p} \geq 0 \qquad \forall t \in T, i \in C \setminus C_S, j \in S(i), p \in P \tag{7-20}$$

7.3. Solution Technique

In this section, we present a heuristic solution technique that integrates Projection Algorithm and Genetic Algorithm (GA) to solve the described formulation. The projection algorithm is widely employed to solve a system of variational inequality. It is also used to find an approximation for the solution of UEDTA. Even though the method of successive averages can also be used to find user equilibrium flows, the projection algorithm has shown better performances with higher quality solutions compared to the method of successive averages. The projection algorithm switches the demand among paths such that a UE solution is found. This algorithm involves solving a quadratic program, simulating the network, and finding travel times iteratively (Doan and Ukkusuri, 2015; Facchinei and Pang, 2003; Nie and Zhang, 2010; Ukkusuri et al., 2012). Moreover, GA explores the program feasible region to find good quality solutions. GA is inspired by the concepts of natural evolution and genetics (Goldberg, 1989; Holland, 1975) and is shown to be practical in solving computational optimization problems for transportation systems (Hajbabaie, 2012; Hajbabaie and Benekohal, 2011b; Liu and Song, 2019; Shepherd and Sumalee, 2004). There are also several research records of promising results to solve congestion pricing problems using GA (Fan and Gurmu, 2014; Fan, 2016; Liu et al., 2013; Shepherd and Sumalee, 2004; Zhang and Yang, 2004).

The proposed solution technique includes four main parts: initialization, evaluation, variation, and selection. Figure 7-1 presents a flow chart of the solution technique.

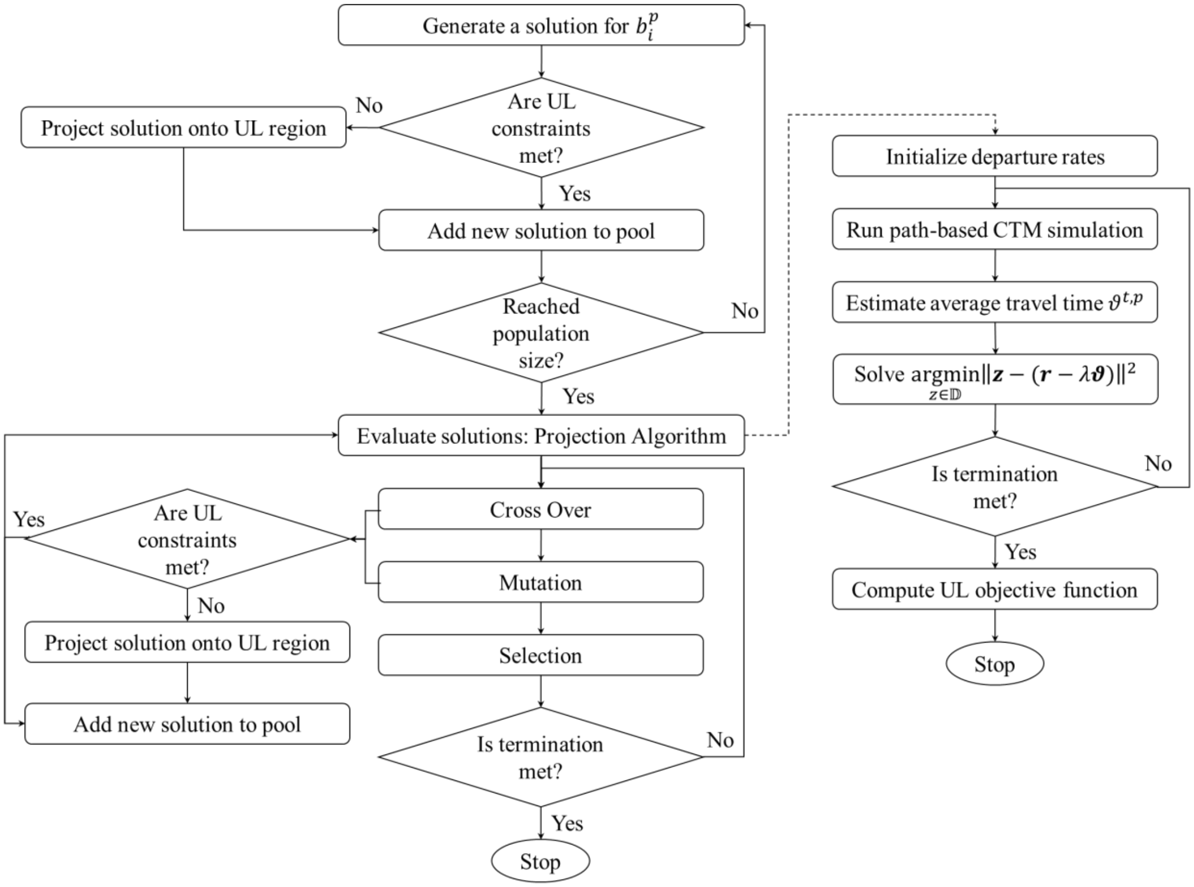

Generate a solution for b_i^p
Are UL constraints met?
No
Project solution onto UL region
Yes
Add new solution to pool
Reached population size?
No
Yes
Evaluate solutions: Projection Algorithm
Yes
Are UL constraints met?
No
Project solution onto UL region
Add new solution to pool
Cross Over
Mutation
Selection
Is termination met?
No
Yes
Stop
Initialize departure rates
Run path-based CTM simulation
Estimate average travel time $\vartheta^{t,p}$
Solve $\underset{z\in\mathbb{D}}{\mathrm{argmin}}\|\boldsymbol{z}-(\boldsymbol{r}-\lambda\boldsymbol{\vartheta})\|^2$
Is termination met?
No
Yes
Compute UL objective function
Stop

Figure 7-1 The flowchart of solution technique for the bottleneck congestion management approach

7.3.1. Initialization

The initialization generates a feasible solution pool. Since the decision variables b_i^p are binary, binary strings are used to show potential solutions. The strings have similar lengths, and the length of each binary string or candidate solution is equal to the summation of the number of paths for all travelers with WTP submission (i.e., size of binary decision variable b_i^p).

After creating a solution by randomly generating a vector of 0 and 1 for variable b_i^p for traveler $i \in B$ and paths $p \in L(i)$, the feasibility of the solution is checked, and if the solution is feasible, it is added to the solution pool. For checking the feasibility, we check if constraints (7-2) and (7-3) are met. If these constraints are not satisfied, we project the current solution onto a feasible region that is formed by constraints (7-2) and (7-3). In other words, we solve the following quadratic problem with a minimization objective function (7-21) and constraints (7-2) and (7-3) to find the closest feasible solution to the current solution such that the new solution is feasible. Let $\boldsymbol{z}$ and $\boldsymbol{b}$ denote a vector of decision variables and a vector of assignment of travelers with WTP submission , respectively. Note that $\|.\|$ is the second norm operator and hence makes the objective function quadratic.

$$min\ \|\boldsymbol{z}-\boldsymbol{b}\| \tag{7-21}$$

Constraints (7-2) and (7-3)

7.3.2. Evaluation: Solving the UEDTA problem

In this section, we present an algorithm to determine the quality of each solution with a real number, i.e., fitness. We should find the quality of the solutions by computing the objective function values of the first part of formulation for all solutions in the pool. This objective function has two parts: total travel time and revenue. The total travel time should be estimated by solving a UEDTA problem. Since UEDTA does not have a closed form, we use the projection algorithm to solve UEDTA. The revenue part of the objective function can be computed by multiplying the travelers assignment b_i^p by WTP value π_i^p and summing the result over all travelers and paths involved in the WTP submission process, i.e., $\sum_{p\in L(i)}\sum_{i\in B} b_i^p\ \pi_i^p$.

The projection algorithm proposed by Doan and Ukkusuri (2015) is adopted to solve the UEDTA problem. This algorithm starts with some initial demand for paths and departure route choice rates. The network is then simulated to find occupancy and flow across the network, and the average travel time is computed using simulation outputs. The number of travelers on each path is updated using the negative direction of estimated average travel times. The updated values are then projected to the feasible area to find feasible departure rates. The new departure rates are used as initial solutions for the next iteration. This algorithm iterates until the termination criterion, which is a specific number of iterations in our case, is met. The departure route choice rates are changed over iterations such that a UE solution is found. The projection algorithm is implemented to evaluate all solutions simultaneously and in parallel. More details on the steps of the projection algorithm are provided below.

Considering each solution in the pool, the initial demand profile should be adjusted knowing how travelers with WTP submission are assigned to paths with bottlenecks. The demand on these paths can be determined considering these travelers' assignments in the current solution. The demand for paths that do not have bottlenecks can be updated by adding the travelers who do not win in the selection process to their alternative paths.

A path-based simulation is then executed to find occupancy $x_i^{t,p}$ and flow $y_{ij}^{t,p}$ across the network. The simulation is described in detail in 4.1.1. Paths and their demands are given as input to this step. The average path travel times $\vartheta^{t,p}$, is computed as shown in (7-22), (7-23), and (7-24) similar to a study by Ukkusuri et al., (2012), using the value of occupancy $x_i^{t,p}$ obtained from the CTM path-based simulation. The travel time is estimated by comparing the cumulative arrival and departure rates in (7-22). We use $o^{t,t',p}$ and $r^{t,p}$ to denote an auxiliary variable for average travel time estimation and departure rate at time step $t \in T$ with departure time $t' \in T'$ for path $p \in P$, respectively.

$$o^{t,t',p} = max\ (0, \sum_{h=0}^{t} r^{h,p} - x_{D(p)}^{t',p}) \qquad \forall t' \in T', t \in T\ , p \in P \tag{7-22}$$

$$\vartheta^{0,p} = \frac{\sum_{h=0}^{T'-1} (o^{0,h,p} - o^{0,h+1,p})h}{r^{0,p} + \mu} \qquad \forall p \in P \tag{7-23}$$

$$\vartheta^{t,p} = \frac{\sum_{h=t}^{T'-1}(o^{t,h,p} - o^{t,h+1,p} + o^{t-1,h+1,p} - o^{t-1,h,p})(h-t)}{r^{t,p} + \mu} \quad \forall t \in T, p \in P \tag{7-24}$$

In the next step, the departure rates are updated, as shown in equation (7-25). Let λ denote a step size. The departure rate of iteration $k+1$ at time step $t \in T$ for path $p \in P$ is found by moving the departure rate $(r^{t,p})^k$ of previous iteration k towards the negative direction of average travel time $\vartheta^{t,p}$ considering step size λ.

$$(r^{t,p})^{k+1} = (r^{t,p})^k - \lambda\vartheta^{t,p} \quad \forall t \in T, p \in P \tag{7-25}$$

The updated departure rates are now projected onto constraints (7-27) using the following problem formulation. The objective function (7-26) minimizes the distance between the previous departure rate vector $\boldsymbol{r}$ and the new rates obtained by solving this formulation. Let us use $\boldsymbol{z}$ and $\boldsymbol{r}$ to denote a vector of variables for departure rates and original departure rates. Constraint (7-27) puts the summation of departure rates $r^{t,p}$ over all paths $p \in K$ with no bottlenckes that conntces origin-destination pair $od \in OD$ equal to demand for all times $t \in T'$.

$$min\, \|\boldsymbol{z} - \boldsymbol{r}\| \tag{7-26}$$

$$\sum_{p \in K: O(p)=o, D(p)=d} r^{t,p} = d^{t,od} \quad \forall t \in T', od \in C_{od} \tag{7-27}$$

Then, we check the termination criterion. If the termination criterion is met, we stop the algorithm. We use the latest result from the CTM simulation to compute the objective function of the first part of the formulation. If the termination is not met, a new CTM path-based simulation is executed, and we go back to the travel time estimation step.

This algorithm is summarized in the following steps:

1. Set $k = 0$. Initialize a feasible departure rate r^0

2. Run the simulation CTM and compute $\boldsymbol{\vartheta}$
3. Update departure rates $\boldsymbol{r}^{k+1} = \boldsymbol{r}^{k} - \lambda\boldsymbol{\vartheta}$
4. Project solutions of step 3 using $\underset{\boldsymbol{z}\in\mathbb{D}}{argmin}\|\boldsymbol{z} - \boldsymbol{r}^{k+1}\|^2$
5. If the termination is met, stop the algorithm and put $\boldsymbol{r}^* = \boldsymbol{z}^*$. Otherwise, set $\boldsymbol{r}^{k+1} = \boldsymbol{z}^*$, $k = k + 1$ and go to step 2.

7.3.3. Variation

In this section, we generate new solutions by making changes to the current solutions to explore the feasible region and find better solutions. New solutions are generated using two variation operations: crossover and mutation. The crossover operation generates new solutions by exchanging several strings between a pair of solutions. The mutation operator changes a few strings in a solution to generate solutions that are slightly different and find solutions close to current solutions.

7.3.3.1. Crossover

We provide a higher probability of selecting the solutions with higher fitness values or objective function values of the first part of the formulation. Equation (7-28) shows the calculation of selection probability for a solution. Let S presents a set of feasible solutions, φ_i denotes the selection probability of solution $i \in S$, and ϖ_i shows the fitness value or objective function value of the first part of formulation for the current solution $i \in S$.

$$\varphi_i = \frac{e^{-\frac{\varpi_i}{\max_i \varpi_i}}}{\sum_i \varphi_i} \qquad \forall i \in S \qquad (7\text{-}28)$$

We then use selection probability values φ_i to run a roulette wheel selection or fitness proportionate selection algorithm. The roulette wheel selection algorithm selects two solutions for pairing. These solutions are input to the crossover operation, and we implemented one point crossover with full replacement to generate two new solutions. The crossing position is selected randomly to exchange binary strings. The number of generated solutions from crossover operation in each iteration of the proposed solution technique is equal to cross over percentage p_c multiplied by population size.

The feasibility of two generated solutions from crossover operation is evaluated by checking constraints (7-2) and (7-3). If the new solutions are feasible, they are added to the solution pool. Otherwise, the solutions are projected on the constraint set of the first part of the formulation. The problem with a minimization objective function (7-21) and constraints (7-2) and (7-3) is solved to find the closest feasible solutions to the current solution, and the feasible solutions are added to the solution pool.

Then, the feasible solutions are evaluated for quality by running a projection algorithm described in section 7.3.2. This algorithm is parallelized using a multi-thread platform in which each solution is evaluated on one thread. All evaluations are implemented simultaneously, and the objective function values of the first part of formulation as the output of the evaluation operation are stored for each solution.

7.3.3.2. Mutation

In the mutation operation, a random solution is selected, and a random string of this solution is changed from zero to one or vice versa. The number of selected random strings is equal to the multiplication of mutation percentage p_m and the string length. The number of iterations in the mutation and added new solutions are also equal to mutation percentage p_m multiplied by population size.

In case that the generated solution is feasible, it is added to the solution pool. Otherwise, the problem with objective function (7-21) and constraints (7-2) and (7-3) is solved to project the solution onto the constraint set of the first part of formulation and find the closest feasible solution to the current solution. The feasible solution is then added to the solution pool. All generated solutions by mutation operation are evaluated by the projection algorithm, described in section 7.3.2, in parallel, and the objective function values of solutions are stored.

7.3.4. Selection

Once new solutions are generated and added to the solution pool using the variation operation, we discard the solutions with the lowest fitness values. The number of removals is equal to the number of added solutions in the crossover and mutation operations which is the multiplication of the initial population size by $p_m + p_c$. The selection operation helps in keeping the same number of solutions in the pool over iterations of the proposed solution technique.

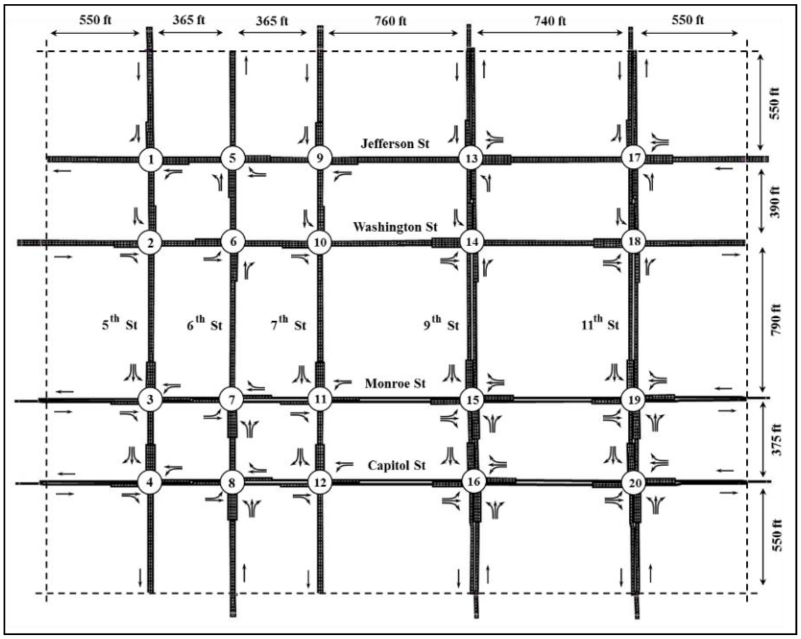
7.4. Test Networks

In this section, the presented formulation is solved using the proposed solution technique for two case studies. The first case study is a portion of the downtown Springfield network in Illinois. This network consists of 20 intersections with one-way and two-way streets. All intersections are signalized with predefined signal timing parameters. We considered 15 OD pairs, 400 time steps, and three demand profiles for this test network. The loading period has 300 time steps, and each time step is 6 seconds. Detailed information on demand profiles and CTM network characteristics are presented in Figure 7-2. The second test network has 40 (4×10) intersections, 25 ODs, and 500 time steps. This network is a hypothetical network and is created by duplicating the network of 20 intersections. Table 7-2 shows three demand profiles for this network. In all cases, demand profile 1, demand profile 2, and demand profile 3 represent under-saturated, semi-saturated, and over-saturated demands, respectively. Note that the OD matrix is dynamic.

The parameters for the genetic algorithm include population size, crossover probability, and mutation probability which are set to 250, 0.5, and 0.2, respectively. The parameter λ for the projection algorithm is set to 0.01, considering the recommendations by Ukkusuri et al., (2012). For each OD pair, three paths are considered resulting in a total of 45 paths in the network with 20 intersections and 75 paths in the network of 40 intersections. Four road segments are considered as bottlenecks in both case studies. The number of paths for WTP submission is 18 and 20 for case studies of 20 and 40 intersections, respectively. These are the paths that pass the road segments which are considered bottlenecks. The value of time is set to $15.31 as suggested by (Oregon Department of Transportation, 2004). In a real-world implementation of this approach, WTPs are determined by travelers, but for our analysis, it is assumed the WTPs are distributed uniformly between zero and four dollars based on the studies on drivers willingness to pay (Brownstone et al., 2003; Jou et al., 2012; Li et al., 2010). The WTPs are not demand responsive. The termination criterion for the solution technique is 200 iterations. We assume travelers are homogenous.

Demand (veh/hr/ln)/OD	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Profile 1	333	133	333	333	67	333	333	333	133	133	333	333	333	333	67
Profile 2	500	200	500	500	100	500	500	500	200	200	500	500	500	500	100
Profile 3	750	300	750	750	150	750	750	750	300	300	750	750	750	750	150

The case study of 20 intersections –Springfield, IL



CTM characteristics	
The duration of each time step (sec)	6
The number of cells in each link	2,3 and 4
The total number of cells	316
The total number of links	387
Free-flow speed (mph)	25
Cell length (ft)	220
The capacity of cells except for source and sink cells (veh/cell)	9, 24, and 36
Saturation flow rate except for source and sink cells (veh/ts/cell)	3,6, and 9
Capacity and Saturation flow rate for source and sink cells	1000

sec: second, mph: mile per hour, ft: feet, veh: vehicle, ts: time step

Figure 7-2 The case study of 20 intersections, its CTM characteristics, and demand patterns

Table 7-2 Demand profiles for the test network of 40 intersections

Network of 40 intersections with 25 ODs							
OD/Demand (veh/hr/ln)	Demand Profile 1	Demand Profile 2	Demand Profile 3	OD/Demand (veh/hr/ln)	Demand Profile 1	Demand Profile 2	Demand Profile 3
1	333	500	750	14	333	500	750
2	267	400	600	15	267	400	600
3	27	40	60	16	27	40	60
4	40	60	90	17	40	60	90
5	267	400	600	18	267	400	600
6	67	100	150	19	67	100	150
7	320	480	720	20	320	480	720
8	200	300	450	21	200	300	450
9	120	180	270	22	120	180	270
10	200	300	450	23	200	300	450
11	120	180	270	24	120	180	270
12	200	300	450	25	320	480	720
13	120	180	270				

7.5. Results

7.5.1. Solution Technique Performance

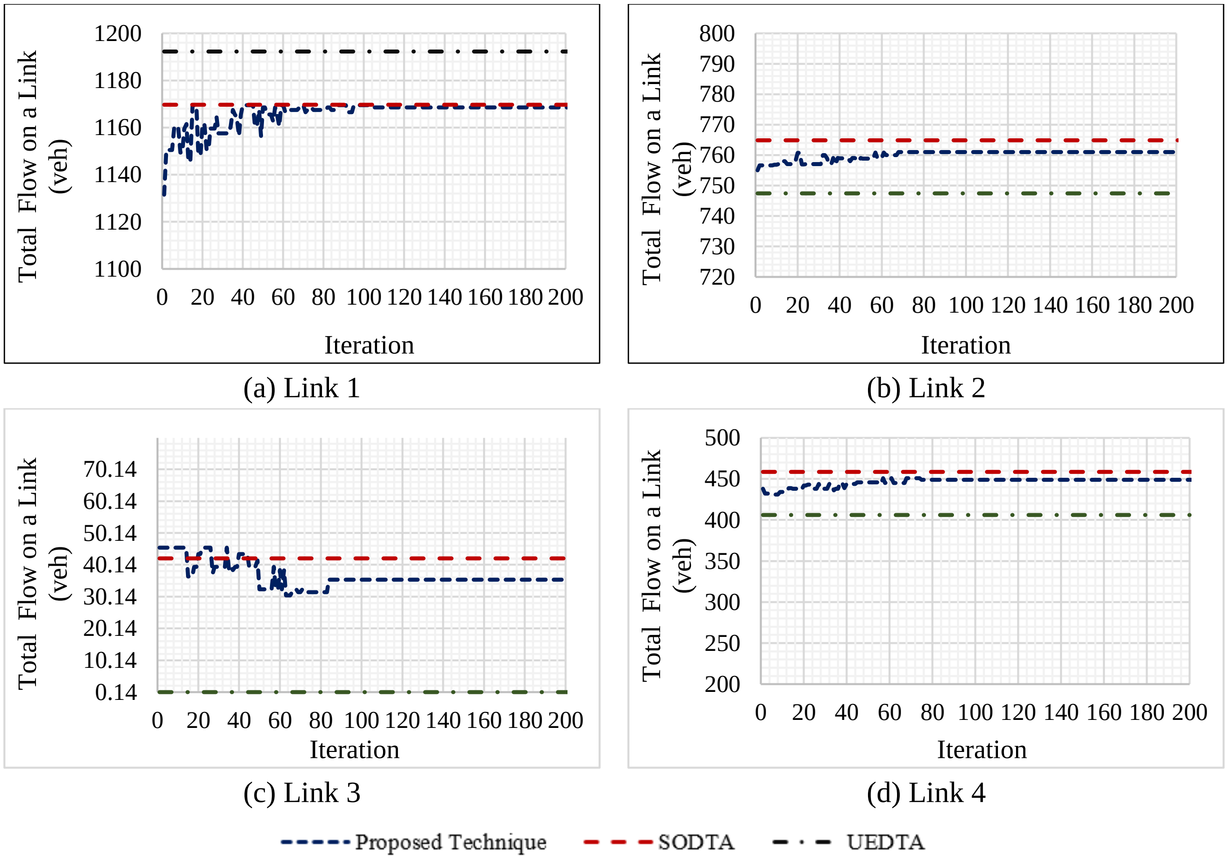
For benchmarking, 3 problems are solved, UEDTA, SODTA, and the proposed approach for bottleneck congestion management. The user equilibrium traffic assignment is solved with the projection algorithm, described in section 7.3.2. SODTA is solved centrally using CPLEX. The presented problem is solved with the proposed solution techniques with projection and genetic algorithms.

In Table 7-3, we compared the total travel time of the proposed solution technique with the total travel times obtained from UEDTA and SODTA problems for three demand profiles. According to the table, the total travel time of the proposed solution technique is within the bounds created by UEDTA and SODTA. We have the minimum total travel time in SODTA since the objective of this problem is to minimize the total travel time. UEDTA's objective is to minimize individual travel time. We have reported the total travel time based on the solution obtained from UEDTA. In the proposed solution technique, the objective has two parts: minimization of a travel time function and maximization of revenue, and, based on its solutions, total travel time is reported. The total travel time of our approach has a gap of 0.03%, 0.05%, and 13.08% with UEDTA, and a gap of 2.83%, 2.72%, and 4.61% with SODTA for demand profiles 1,2, and 3, respectively. When the network has the most congestion in demand profile 3, we can observe the effect of our approach better. In this case, the solutions have a 13.08% gap with UEDTA and a 4.61% gap with SODTA. The goal of our approach is also to keep the solutions away from a UEDTA behavior and as close as possible to SODTA.

Table 7-3 Total travel time of UEDTA, SODTA, and our approach in the case study with 20 intersections with three demand profiles

	Approach/Demand	Demand Profile 1	Demand Profile 2	Demand Profile 3
Total Travel Time (hr)	UEDTA	187.19	297.79	597.95
	Proposed Solution Technique	187.13	297.65	519.72
	SODTA	181.98	289.78	496.84

Figure 7-3. a-d presents the total link flow over iterations for our proposed technique, SODTA, and UEDTA for the case study of 20 intersections with demand profile 3 (oversaturated demand). Four links are selected as link representatives, and each figure shows the total link flow for one link. The total link flow is the summation of flow over all time steps and paths on a link. Our approach objective is to minimize the total travel time of the system and maximize the revenue. Hence, our approach pushes the solutions towards SODTA and away from UEDTA while ensuring revenue is maximized. We observe that the total link flow in all figures has stayed close to the SODTA solution. Since the objective of our problem is both minimizing total travel time and maximizing revenue, the flow may have a gap with the SODTA solution to satisfy the second part of the objective function in the first part of the formulation. We also have a considerable gap with a UEDTA to avoid the selfish behavior of travelers.



(a) Link 1

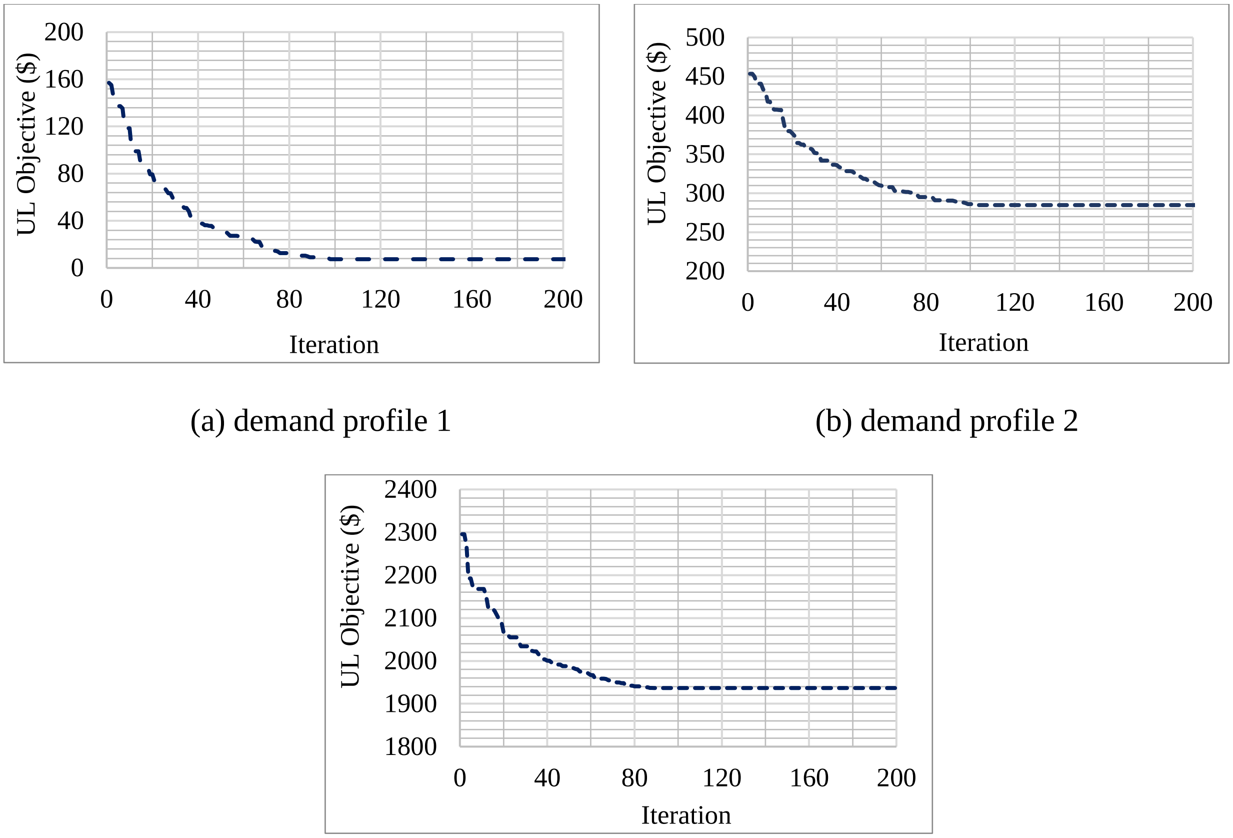
(b) Link 2

(c) Link 3

(d) Link 4

Figure 7-3 Link flows from the proposed technique, SODTA, and UEDTA for the case study of 20 intersections with demand profile 3

Figure 7-4.a, b, and c present the objective function value of the first part of formulation over iterations for the 20-intersection network with Demand Profile 1 (under-saturated), Demand Profile 2 (semi-saturated), and Demand Profile 3 (oversaturated), respectively. The objective of the first part of the formulation is the total travel time function minus revenue, so we expect to see a decreasing trend over iteration. The objective is improved by 95.28%, 37.17%, and 15.63% over iterations in demand profiles 1,2, and 3, respectively. The changes in the objective are zero after iterations 98, 103, and 104, where the solution technique is converged, for under-saturated, semi-saturated, and oversaturated demand profiles, respectively. In all cases, the convergence happened before the termination criterion of 200 iterations.



(a) demand profile 1

(b) demand profile 2

(c) demand profile 3

Figure 7-4 Objective function value of the first part of formulation over iterations for 20 intersections and 3 demand profiles

Figure 7-5 demonstrates a graphical distribution of WTP values for three paths in 2 stages: (1) all WTP values for the path and (2) selected WTP values by the proposed approach. The results are shown for the case study of 20 intersections with the demand profile 3 (oversaturated). The first two box plots present the distribution of WTP values for Path a. The third and fourth plots are for Path b, and the last two plots are for Path c. By comparing the box plots for one path, we observe that the highest WTP values are selected by our approach. Hence, the average WTP values have increased from $2.02 to $2.76 after selecting travelers with WTP submission for Path a. For Path b, the average has increased from $1.87 to $2.59, and, for Path c, the average is increased from $1.91 to $2.54. Other measures including median, first quartile, third quartile, minimum, and maximum for WTP values indicate the selection of the highest WTP values by applying the approach.

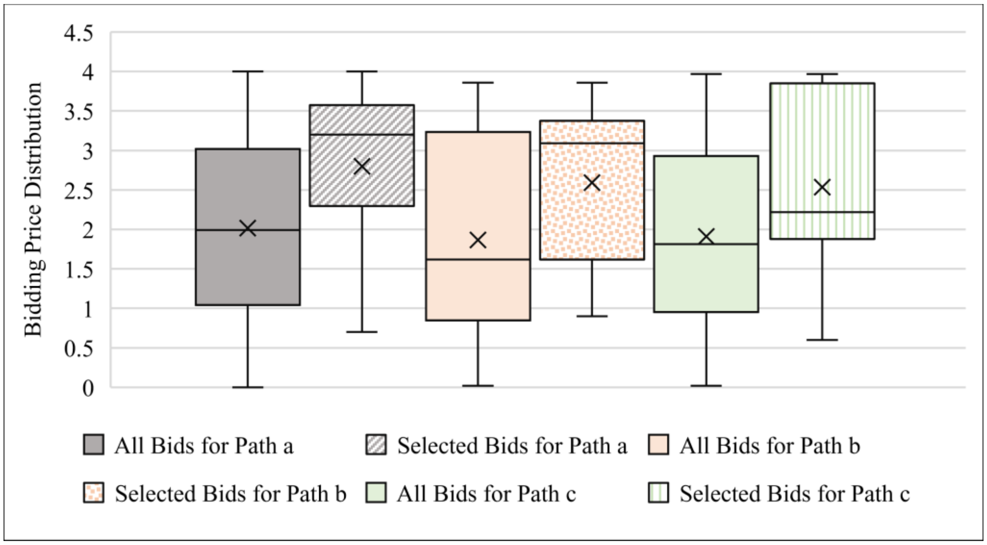


Figure 7-5 The distribution of willingness to pay values in 20 intersections network with demand profile 3

Table 7-4 shows the computation time for each stage and operation in the solution technique for three demand patterns, profile 1 (undersaturated), profile 2 (semi-saturated), and profile 3 (oversaturated) for 20 intersections. The run times are reported for 200 iterations which is the required number of iterations to satisfy the termination criterion. The population size is 250. The most time-consuming operations are the evaluations of solutions that run the projection algorithm. The evaluations in crossover and mutation range from 48.44 to 53.93 minutes. The evaluation operation in the initialization has the least times (0.25, 0.26, and 0.27 minutes) since the initialization is only implemented in the first iteration. Nothing that the evaluation step in all stages is parallelized using a multi-thread approach. The total runtimes are 107.22, 117.35, and 122.34 minutes for demand profiles 1, 2, and 3, respectively. The increase in demand has increased the runtime.

Table 7-4 Breakdown of runtimes for the network of 20 intersections for 3 demand profiles

Stage	Operations	Demand Profile 1	Demand Profile 2	Demand Profile 3
Initialization (min)	Solution pool	0.00	0.00	0.00
	Feasibility	0.14	0.22	0.38
	Evaluation	0.25	0.26	0.27
Variation through Cross Over (min)	Crossover	0.13	0.10	0.08
	Feasibility	3.21	7.96	4.62
	Evaluation	50.43	51.31	53.93
Variation through Mutation (min)	Mutation	0.57	1.03	2.27
	Feasibility	4.05	5.72	8.33
	Evaluation	48.44	50.74	52.45
Selection (min)	Removal	0.00	0.01	0.00

7.5.2. Sensitivity Analysis

Figure 7-6 shows the values for total travel time and revenue with different associated weights in the objective function for the case study of 20 intersections with a semi-saturated demand profile (demand profile 2). We observe that the lowest values for revenue and total travel time happen when their weight is zero and one, respectively, and the highest values occur when their weight is one and zero, respectively, as expected. Increasing the weight of revenue has led to an increase in revenue, and increasing the weight of total travel time has led to the travel time decrease. Decreasing the weight of revenue decreases the revenue and decreasing the weight of total travel time increases the total travel time. The model is more sensitive to weights of 1, 0 and 0.8, 0.2 for revenue and travel time, respectively, compared to other weight combinations.

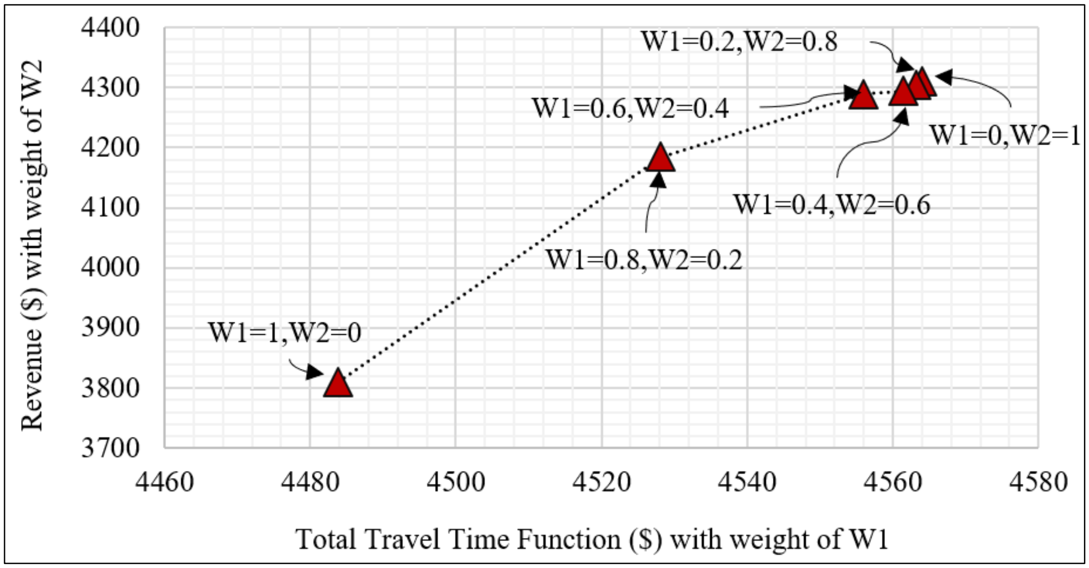


Figure 7-6 Total travel time and revenue with different associated weights for the case study of 20 intersections with demand profile 2

Figure 7-7 shows the effect of population size on the value of the first part of the formulation objective function for the semi-saturated demand profile. For sensitivity analysis, we have selected the semi-saturated demand profile as a good representative of both undersaturated and oversaturated demand. By increasing the population size from 50 to 250, the objective function has improved due to the possibility of finding better solutions in a bigger population. Note that, a population of 250 was the maximum population size that we could test on the computer used for this analysis, considering the cores and memory resources.

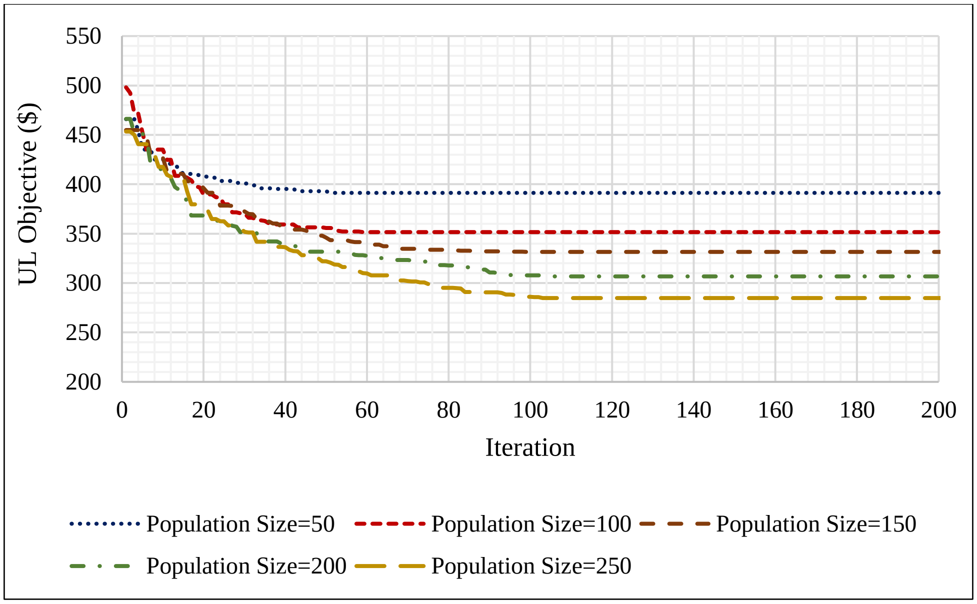


Figure 7-7 Objective function value of first part of formulation over iterations for different population sizes for 20 intersections with demand profile 2

Figure 7-8 presents the objective value of the first part of formulation over iterations for 3 different crossover percentage values, 0.4, 0.5, and 0.6, for the case study of 20 intersections with a semi-saturated demand profile (demand profile 2). The mutation percentage is fixed at 0.2, and the population size is 250. Since finding smaller values for the objective is better, a crossover percentage of 0.5 is better than 0.4, and a crossover percentage of 0.4 is better than 0.6.

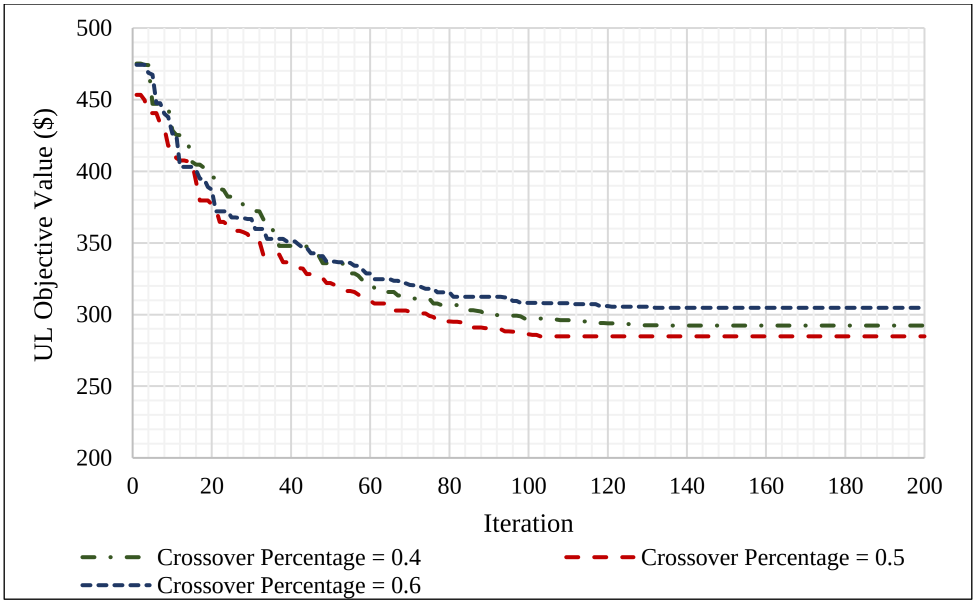


Figure 7-8 Objective function value of first part of formulation over iterations for different crossover percentages for the case study of 20 intersections with demand profile 2

Figure 7-9 presents the objective value of the first part of formulation over iterations for 3 different mutation percentage values, 0.2, 0.3, and 0.4, for the case study of 20 intersections with a semi-saturated demand profile (demand profile 2). We fixed the crossover percentage to 0.5 and the population size to 250. We observed that a mutation percentage of 0.2 has the best performance in terms of the objective value.

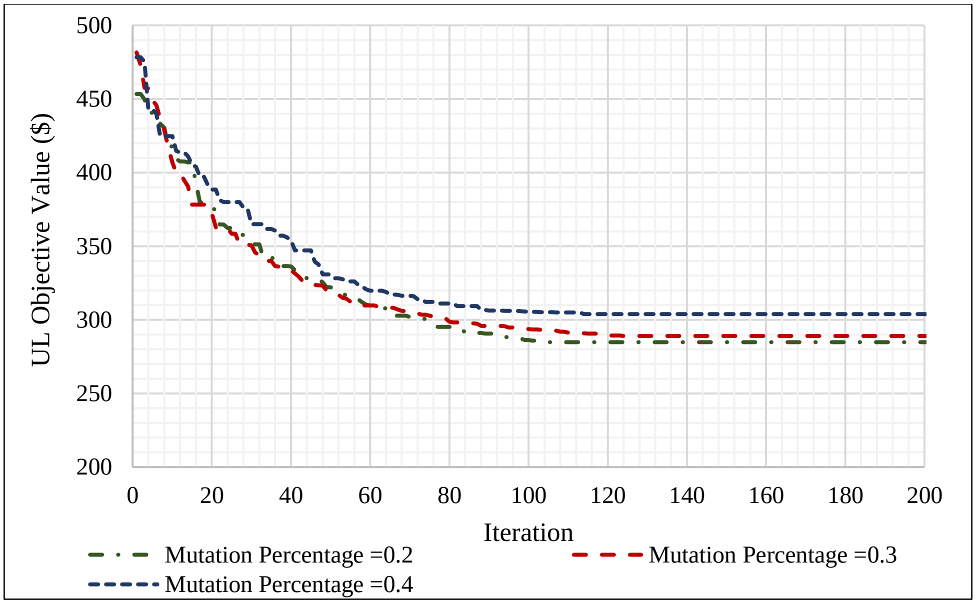


Figure 7-9 Objective function value of first part of formulation over iterations for different mutation percentages for the case study of 20 intersections with demand profile 2

7.5.3. Increasing the Network Size

We also tested the 40 intersections with 632 cells, 780 links, 500 time steps, and 25 OD pairs. Table 7-5 presents the objective function of the first part of the formulation, runtime, total travel time, and a lower level for the network of 40 intersections with 3 demand profiles. By increasing the number of intersections from 20 to 40, the number of decision variables has increased from more than ~4 million to ~14 million. The objective function value has increased by increasing the demand from profiles 1 to 3. The run time has not been affected much by changes in demand levels. The solutions are compared with a lower bound in the last two columns. We can observe that the total travel time of our approach has a 2.78-4.15% gap with SODTA which indicates that our approach behavior is close enough to SODTA.

Table 7-5 The performance of the case study with 40 intersections for three demand profiles

Demand	First part Objective Function ($)	Runtime (min)	Total Travel Time (hr)	Lower Bound - SODTA Objective Value (hr)
Demand Profile 1	2033.57	295.32	183.75	178.16
Demand Profile 2	3360.61	297.37	289.33	281.50
Demand Profile 3	6257.89	294.69	524.12	503.25

Table 7-6 shows the computation time for each stage and operation in the solution technique for three demand patterns, profile 1 (undersaturated), profile 2 (semi-saturated), and profile 3 (oversaturated) for the case study of 40 intersections. The run times are reported for 200 iterations which is the termination criterion. The population size is 250. Similar to the case study with 20 intersections, the computation time of evaluation with the projection algorithm has the highest runtime among all operations. The evaluation operations are parallelized to accelerate the algorithm speed. The increase in demand from profiles 1 to 3 has not necessarily led to an increase in runtime.

Table 7-6 Breakdown of runtimes for the case study with 40 intersections

Stage	Operations	Demand Profile 1	Demand Profile 2	Demand Profile 3
Initialization (min)	Solution pool	0.00	0.00	0.00
	Feasibility	0.03	0.06	0.06
	Evaluation	0.79	0.76	0.74
Variation through Cross Over (min)	Crossover	0.03	0.11	0.06
	Feasibility	0.48	0.13	0.50
	Evaluation	150.74	156.97	144.64
Variation through Mutation (min)	Mutation	0.23	0.28	0.32
	Feasibility	0.83	1.10	1.20
	Evaluation	142.17	137.95	147.16
Selection (min)	Removal	0.00	0.01	0.01

CHAPTER 8. CONCLUSIONS

Dynamic Traffic Assignment (DTA) models can be employed for congestion management of transportation networks. The reduction in traffic congestion leads to less environmental pollutants, travel times, and energy consumption. This dissertation has proposed several approaches to optimize time-dependent link/path flows for vehicles traveling in a transportation network and developed an optimization-based approach for bottleneck congestion management to improve traffic operations in urban transportation networks.

DTA problems have large temporal and spatial scales with complex traffic dynamics, especially if they follow a relatively accurate network loading model. Hence, solving these problems requires high computational resources. One main goal of this dissertation is to present decomposition and distributed approaches to tackle the large decision spaces in System Optimal (DTA) problems. This line of research has proposed analytical and simulation-based methodologies to address the challenges of solving the SODTA problem while capturing realism in traffic flow modeling. The shared idea among proposed methodologies in the three chapters is to convert the problem into several sub-problems using decomposition techniques. The optimization of sub-problems needs fewer computations due to small decision spaces compared to the original problem. The sub-problems are then coordinated through either master problems, information exchange graphs, or direct communications. This dissertation has employed the notion of an information exchange graph that coordinates sub-problems without requiring a central component. The dissertation also has taken the distributed and decomposition idea to the next stage by solving the problem in real-time with tight optimality gaps. More details follow.

In the first study, this research has employed the Dantzig-Wolfe decomposition principle to solve the SODTA problem. The approach proposes an origin and destination-based decomposition scheme to generate independent Sub-Problems (SPs) with less computational complexity compared to the original problem. This decomposition also facilitates their parallelization on high-performance computing machines. The relaxed interactions among SPs are handled by solving a Master Problem (MP). This study has resolved the limited scalability of existing solution techniques, guarantees convergence to the global optimality in a finite number of iterations, and is not restricted to specific network properties.

The next study has proposed the development of a distributed gradient-based approach. This approach overcomes the main drawback of the existing techniques by having a fully distributed framework that does not require a centralized component, i.e., MP. The approach employs an information exchange graph to update the value of shared decision variables among SPs, considers the gradient of the objective function to minimize travel time, and projects the value of decision variables on the feasible region of each SP simultaneously. This approach has provided an independent computational complexity from the number of nodes and links in the network, and it is proved the approach converges to the optimal solution in an infinite number of iterations.

Furthermore, the development of a real-time approach has been also studied. This work employs an intersection-level decomposition, distributed coordination, and rolling horizon technique that leads to less computational complexity and real-time performance. The information is coordinated among SPs by updating the objective function and constraints of SPs at each horizon. This information is gained by simulating the network and using optimized route decisions from SPs.

This line of research has resulted in heuristic and optimization-based approaches that solve larger problems in less time with lower computational resources compared to the existing state-of-the-art approaches. The approaches allow researchers to optimize paths of vehicles in real-size networks with accurate network flow models. The presented methodologies can assist decision-makers of transportation systems to manage the flow of vehicles efficiently with more accurate network loading models or incorporate methodologies inside their applications like transportation management systems as a sub-module.

Lastly, this dissertation has presented a formulation and a solution technique to manage congestion at bottlenecks. This formulation minimizes total travel time and maximizes revenue by assigning travelers, who submit their willingness to pay values, to the network paths of a transportation network. The first part of the formulation assigns travelers with the willingness to pay submission to network paths with bottlenecks, and the second part of the formulation estimates other travelers' route selection behavior by solving a User Equilibrium (UE) dynamic traffic assignment problem. UEDTA is modeled with a variational inequality approach. The heuristic approach including projection and the genetic algorithm is designed to solve this nonlinear formulation. The algorithm evaluates a set of solutions, using the projection algorithm iteratively, generated by the genetic algorithm.

One future research direction is to study distributed approaches in the presence of noisy communication and stochastic errors. Noises and errors can be the result of limited accessibility to the true value of the gradient of the objective function and the exchange of noise-corrupted information among subproblems or subnetworks. It is interesting to show the robustness of the approach by proving the convergence to an optimal solution or a good quality solution in the presence of perturbations.

The proposed methodologies can be employed and generalized to solve DTA formulations with other network loading models, DTA formulations for multi-modal transportation systems (Pi et al., 2019), and formulations with mixed fleets of automated and human-driven vehicles (Mirheli et al., 2019; Mohebifard and Hajbabaie, 2021a, 2021b, 2020; Niroumand et al., 2020; Tajalli et al., 2022; Tajalli and Hajbabaie, 2021b). The methodologies can also be adopted to solve formulations for other congestion management approaches including traffic metering (Mohebifard and Hajbabaie, 2018a, 2018b), speed optimization (Tajalli et al., 2020; Tajalli and Hajbabaie, 2018a, 2018b), trajectory optimization (Mohebifard and Hajbabaie, 2021a), and signal timing optimization (Al Islam and Hajbabaie, 2017; Islam et al., 2020; Mehrabipour and Hajbabaie, 2022b). The methodologies can also be applied to solve formulations for different applications of DTA such as policy evaluation, evacuation planning, and environmental-related studies.

The proposed methodologies have reduced the problem complexity significantly. As such, the methodologies can allow solving cooperative traffic assignments in which different congestion management techniques are modeled as one formulation for the possibility of higher improvements in the network performance. Mohebifard et al. (2019) and Tajalli et al. (2020) have shown by solving cooperative congestion management problems, we can observe more reductions in traffic congestion.

This dissertation distributed the traffic assignment problem into several intersection-level sub-problems. Other network partitioning approaches when decomposing the network or formulation are other subjects to explore. Different network partitioning approaches or sub-network sizes can affect the convergence and run-time of our approaches. A tradeoff between convergence rate and computation time exists that is affected by the sub-network complexity. A similar study to research by Yahia et al., (2018) for proposed solutions techniques can provide insights into the effect of various partitioning strategies on solutions.

Another extension for proposed approaches is to develop a solution technique that does not require a simulation for coordinating sub-problems. Since the simulation is network-level, even though it is very fast, increasing the network size can affect its performance. Moreover, we have developed approaches whose complexity is not dependent on study period length and the number of cells, but they are still dependent on the number of Origin-Destination (OD) pairs. The development of an approach with independent complexity on OD pairs can increase the scalability of algorithms for solving the SODTA problem. We can also study the possibility of estimating the expected number of iterations to achieve a solution within a small gap and how to deal with an excessive number of OD pairs and what would be their effects on the problem's complexity in future studies. We mainly focused on distributing the problem to SPs such that we can parallelize SPs for faster performance. Designing efficient high-performance computing architectures is a future direction for this dissertation.

Moreover, studying the bottleneck congestion management problem under uncertainty associated with demand and capacity is also an interesting topic for further exploration. Considering real-time data feeds and real-time decision-making in algorithm development for the proposed approach can be an interesting future study.

CHAPTER 9. REFERENCES

Aashtiani, H.Z., Magnanti, T.L., others, 1983. A linearization and decomposition algorithm for computing urban traffic equilibria.

Abrahamsson, T., 1998. Estimation of origin-destination matrices using traffic counts-a literature survey.

Adeli, H., 2000. High-performance computing for large-scale analysis, optimization, and control. J. Aerosp. Eng. 13, 1–10.

Adeli, H., Hung, S.-L., 1993. A concurrent adaptive conjugate gradient learning algorithm on MIMD shared-memory machines. Int. J. Supercomput. Appl. 7, 155–166.

Adeli, H., Kamal, O., 2014. Parallel processing in structural engineering. CRC Press.

Adeli, H., Kamal, O., 1992a. Concurrent analysis of large structures—I. Algorithms. Comput. Struct. 42, 413–424.

Adeli, H., Kamal, O., 1992b. Concurrent analysis of large structures—II. Applications. Comput. Struct. 42, 425–432.

Adeli, H., Kumar, S., 1998. Distributed computer-aided engineering. CRC Press.

Adeli, H., Kumar, S., 1995. Distributed finite-element analysis on network of workstations—algorithms. J. Struct. Eng. 121, 1448–1455.

Ahuja, R.K., Magnanti, T.L., Orlin, J.B., 1993. Network flows: theory, algorithms, and applications.

Al Islam, S.M.A. Bin, Hajbabaie, A., 2021. An enhanced cell transmission model for multi-class signal control. IEEE Trans. Intell. Transp. Syst.

Al Islam, S.M.A. Bin, Hajbabaie, A., 2017. Distributed coordinated signal timing optimization in connected transportation networks. Transp. Res. Part C Emerg. Technol. 80, 272–285.

Aziz, H.M., Ukkusuri, S. V, 2012. Integration of environmental objectives in a system optimal dynamic traffic assignment model. Comput. Civ. Infrastruct. Eng. 27, 494–511.

Ban, X.J., Liu, H.X., Ferris, M.C., Ran, B., 2008. A link-node complementarity model and solution algorithm for dynamic user equilibria with exact flow propagations. Transp. Res. Part B Methodol. 42, 823–842.

Barton, R.R., Hearn, D.W., Lawphongpanich, S., 1989. The equivalence of transfer and generalized benders decomposition methods for traffic assignment. Transp. Res. Part B Methodol. 23, 61–73.

Basar, G., Cetin, M., 2017. Auction-based tolling systems in a connected and automated vehicles environment: Public opinion and implications for toll revenue and capacity utilization. Transp. Res. Part C Emerg. Technol. 81, 268–285.

Beard, C., Ziliaskopoulos, A., 2006. System optimal signal optimization formulation. Transp. Res. Rec. J. Transp. Res. Board 102–112.

Bertsekas, D.P., Gafni, E.M., 1982. Projection methods for variational inequalities with application to the traffic assignment problem, in: Nondifferential and Variational Techniques in Optimization. Springer, pp. 139–159.

Boyles, S.D., 2012. Bush-based sensitivity analysis for approximating subnetwork diversion. Transp. Res. Part B Methodol. 46, 139–155.

Brownstone, D., Ghosh, A., Golob, T.F., Kazimi, C., Van Amelsfort, D., 2003. Drivers' willingness-to-pay to reduce travel time: evidence from the San Diego I-15 congestion pricing project. Transp. Res. Part A Policy Pract. 37, 373–387.

Carey, M., 1992. Nonconvexity of the dynamic traffic assignment problem. Transp. Res. Part B Methodol. 26, 127–133.

Carey, M., 1987. Optimal time-varying flows on congested networks. Oper. Res. 35, 58–69.

Carey, M., 1986. A constraint qualification for a dynamic traffic assignment model. Transp. Sci. 20, 55–58.

Carey, M., Subrahmanian, E., 2000. An approach to modelling time-varying flows on congested networks. Transp. Res. Part B Methodol. 34, 157–183.

Chakraborty, S., Rey, D., Moylan, E., Waller, S.T., 2018. Link Transmission Model-Based Linear Programming Formulation for Network Design. Transp. Res. Rec. 0361198118774753.

Chen, R.-J., Meyer, R.R., 1988. Parallel optimization for traffic assignment. Math. Program. 42, 327–345.

Chiu, Y.-C., Zheng, H., 2007. Real-time mobilization decisions for multi-priority emergency response resources and evacuation groups: model formulation and solution. Transp. Res. Part E Logist. Transp. Rev. 43, 710–736.

Chow, A.H.F., 2009. Properties of system optimal traffic assignment with departure time choice and its solution method. Transp. Res. Part B Methodol. 43, 325–344.

Daganzo, C.F., 1995. The cell transmission model, part II: network traffic. Transp. Res. Part B Methodol. 29, 79–93.

Daganzo, C.F., 1994a. The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. Transp. Res. Part B Methodol. 28, 269–287.

Daganzo, C.F., 1994b. The cell transmission model: a simple dynamic representation of highway trafc. Transp Res Part B 28 269287.

Dantzig, G.B., Wolfe, P., 1960. Decomposition principle for linear programs. Oper. Res. 8, 101–111.

de Palma, A., Proost, S., Seshadri, R., Ben-Akiva, M., 2018. Congestion tolling-dollars versus tokens: A comparative analysis. Transp. Res. Part B Methodol. 108, 261–280.

Di, X., Liu, H.X., Ban, X.J., 2016. Second best toll pricing within the framework of bounded rationality. Transp. Res. Part B Methodol. 83, 74–90.

Doan, K., Ukkusuri, S., Han, L., 2011. On the existence of pricing strategies in the discrete time heterogeneous single bottleneck model. Transp. Res. Part B Methodol. 45, 1483–1500.

Doan, K., Ukkusuri, S. V, 2015. Dynamic system optimal model for multi-OD traffic networks with an advanced spatial queuing model. Transp. Res. Part C Emerg. Technol. 51, 41–65.

Doan, K., Ukkusuri, S. V, 2012. On the holding-back problem in the cell transmission based dynamic traffic assignment models. Transp. Res. Part B Methodol. 46, 1218–1238.

Facchinei, F., Pang, J.-S., 2003. Finite-dimensional variational inequalities and complementarity problems. Springer.

Fan, W., Gurmu, Z., 2014. Combined decision making of congestion pricing and capacity expansion: genetic algorithm approach. J. Transp. Eng. 140, 4014031.

Fan, W.D., 2016. Optimal congestion pricing toll design under multiclass transportation network schemes: Genetic algorithm approaches. Case Stud. Transp. Policy 4, 78–87.

Federal Highway Administration, 2005. Highway economic requirements system—State version.

Florian, M., Constantin, I., Florian, D., 2009. A new look at projected gradient method for equilibrium assignment. Transp. Res. Rec. J. Transp. Res. Board 10–16.

Ford Jr, L.R., Fulkerson, D.R., 1958. A suggested computation for maximal multi-commodity network flows. Manage. Sci. 5, 97–101.

Friesz, T.L., Luque, J., Tobin, R.L., Wie, B.-W., 1989. Dynamic network traffic assignment considered as a continuous time optimal control problem. Oper. Res. 37, 893–901.

Gartner, N.H., Messer, C., Rathi, A., 2002. Traffic Flow Theory. A revised State of the Art Report.

Transp. Res. Board Spec. Report, http//www. tfhrc. gov/its/tft/tft. htm.

Ghali, M.O., Smith, M.J., 1995. A model for the dynamic system optimum traffic assignment problem. Transp. Res. Part B Methodol. 29, 155–170.

Gibert, A., 1968. A method for the traffic assignment problem. Rep. LBSTNT-95, Transp. Netw. Theory Unit, London Bus. Sch. London.

Goldberg, D.E., 1989. Genetic algorithms in search, optimization, and machine learning. Addison. Reading.

Gubin, L.G., Polyak, B.T., Raik, E. V, 1967. The method of projections for finding the common point of convex sets. USSR Comput. Math. Math. Phys. 7, 1–24.

Hajbabaie, A., 2012. Intelligent dynamic signal timing optimization program. University of Illinois at Urbana-Champaign.

Hajbabaie, A., Abdel-Rahim, A., Sorour, S., 2021. HIERARCHICAL PRIORITY--BASED CONTROL OF SIGNALIZED INTERSECTIONS IN SEMI-CONNECTED CORRIDORS.

Hajbabaie, A., Benekohal, R.F., 2013. Traffic signal timing optimization. Transp. Res. Rec. 2355, 10–19. https://doi.org/10.3141/2355-02

Hajbabaie, A., Benekohal, R.F., 2011a. Common or Variable Cycle Length Policy for a More Efficient Network Performance?, in: Transportation and Development Institute Congress 2011: Integrated Transportation and Development for a Better Tomorrow. pp. 1138–1146.

Hajbabaie, A., Benekohal, R.F., 2011b. Does traffic metering improve network performance efficiency?, in: 2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC). IEEE, pp. 1114–1119. https://doi.org/10.1109/ITSC.2011.6083011

Hajbabaie, A., Medina, J.C., Benekohal, R.F., NEXTRANS, C., 2011. Traffic signal coordination and queue management in oversaturated intersections.

Hajbabaie, A., Mohebifard, R., Consortium, P.N.T., others, 2020. Dynamic Metering in Connected Urban Street Networks: Improving Mobility.

Hajbabaie, A., Sattarov, S., Mohebifard, R., Consortium, P.N.T., others, 2018. Safety and Operations Assessment of Various Left-Turn Phasing Strategies.

Han, L., Ukkusuri, S., Doan, K., 2011. Complementarity formulations for the cell transmission model based dynamic user equilibrium with departure time choice, elastic demand and user heterogeneity. Transp. Res. Part B Methodol. 45, 1749–1767.

Hearn, D.W., 1984. Practical and theoretical aspects of aggregation problems in transportation planning models. Publ. Elsevier Sci. Publ. BV.

Ho, J.K., 1980. A successive linear optimization approach to the dynamic traffic assignment problem. Transp. Sci. 14, 295–305.

Holland, J., 1975. adaptation in natural and artificial systems, university of michigan press, ann arbor,". Cité page 100, 33.

Islam, S., Hajbabaie, A., Aziz, H., 2020. A real-time network-level traffic signal control methodology with partial connected vehicle information. Transp. Res. Part C Emerg. Technol. 121. https://doi.org/10.1016/j.trc.2020.102830

Islam, S., Tajalli, M., Mohebifard, R., Hajbabaie, A., 2021. The Effects of Connectivity and Traffic Observability on an Adaptive Traffic Signal Control System. Transp. Res. Rec. Accepted.

Jafari, E., Boyles, S.D., 2016. Improved bush-based methods for network contraction. Transp. Res. Part B Methodol. 83, 298–313.

Jafari, E., Pandey, V., Boyles, S.D., 2017. A decomposition approach to the static traffic assignment problem. Transp. Res. Part B Methodol. 105, 270–296.

Joksimovic, D., Bliemer, M.C.J., Bovy, P.H.L., 2005. Optimal toll design problem in dynamic traffic networks with joint route and departure time choice. Transp. Res. Rec. 1923, 61–72. https://doi.org/10.3141/1923-07

Jou, R.-C., Chiou, Y.-C., Chen, K.-H., Tan, H.-I., 2012. Freeway drivers' willingness-to-pay for a distance-based toll rate. Transp. Res. Part A Policy Pract. 46, 549–559.

Karakitsiou, A., Mavrommati, A., Migdalas, A., 2004. Efficient minimization over products of simplices and its application to nonlinear multicommodity network problems. Oper. Res. 4, 99.

Karoonsoontawong, A., Waller, S.T., 2010. Integrated network capacity expansion and traffic signal optimization problem: robust bi-level dynamic formulation. Networks Spat. Econ. 10, 525–550.

Kockelman, K.M., Kalmanje, S., 2005. Credit-based congestion pricing: a policy proposal and the public's response. Transp. Res. Part A Policy Pract. 39, 671–690.

Kumar, S., Adeli, H., 1995. Distributed Finite-Element Analysis on Network of Workstations—Implementation and Applications. J. Struct. Eng. 121, 1456–1462.

Larsson, T., Patriksson, M., 1995. An augmented Lagrangean dual algorithm for link capacity side constrained traffic assignment problems. Transp. Res. Part B Methodol. 29, 433–455.

Larsson, T., Patriksson, M., 1992. Simplicial decomposition with disaggregated representation for the traffic assignment problem. Transp. Sci. 26, 4–17.

Larsson, T., Patriksson, M., Rydergren, C., 2004. A column generation procedure for the side constrained traffic equilibrium problem. Transp. Res. Part B Methodol. 38, 17–38.

Lawphongpanich, S., Hearn, D.W., 2004. An MPEC approach to second-best toll pricing. Math. Program. 101, 33–55.

Leventhal, T., Nemhauser, G., Trotter Jr, L., 1973. A column generation algorithm for optimal traffic assignment. Transp. Sci. 7, 168–176.

Li, Y., Waller, S.T., Ziliaskopoulos, T., 2003. A decomposition scheme for system optimal dynamic traffic assignment models. Networks Spat. Econ. 3, 441–455.

Li, Z., Hensher, D.A., Rose, J.M., 2010. Willingness to pay for travel time reliability in passenger transport: A review and some new empirical evidence. Transp. Res. part E Logist. Transp. Rev. 46, 384–403.

Lin, D.-Y., Unnikrishnan, A., Waller, S.T., 2011a. A dual variable approximation based heuristic for dynamic congestion pricing. Networks Spat. Econ. 11, 271–293.

Lin, D.-Y., Valsaraj, V., Waller, S.T., 2011b. A Dantzig-Wolfe Decomposition-Based Heuristic for Off-line Capacity Calibration of Dynamic Traffic Assignment. Comput. Civ. Infrastruct. Eng. 26, 1–15.

Lin, W.-H., Ahanotu, D., 1995. Validating the basic cell transmission model on a single freeway link. PATH Tech. note; 95-3.

Liu, Z., Meng, Q., Wang, S., 2013. Speed-based toll design for cordon-based congestion pricing scheme. Transp. Res. Part C Emerg. Technol. 31, 83–98.

Liu, Z., Song, Z., 2019. Strategic planning of dedicated autonomous vehicle lanes and autonomous vehicle/toll lanes in transportation networks. Transp. Res. Part C Emerg. Technol. 106, 381–403.

Lo, H.K., Szeto, W.Y., 2002. A cell-based variational inequality formulation of the dynamic user optimal assignment problem. Transp. Res. Part B Methodol. 36, 421–443.

Long, J., Chen, J., Szeto, W.Y., Shi, Q., 2016. Link-based system optimum dynamic traffic assignment problems with environmental objectives. Transp. Res. Part D Transp. Environ.

Long, J., Wang, C., Szeto, W.Y., 2018. Dynamic system optimum simultaneous route and departure time choice problems: Intersection-movement-based formulations and comparisons. Transp. Res. Part B Methodol. 115, 166–206.

Lotito, P.A., 2006. Issues in the implementation of the DSD algorithm for the traffic assignment problem. Eur. J. Oper. Res. 175, 1577–1587.

Lu, C.-C., Liu, J., Qu, Y., Peeta, S., Rouphail, N.M., Zhou, X., 2016. Eco-system optimal time-dependent flow assignment in a congested network. Transp. Res. Part B Methodol. 94, 217–239.

MacCluer, C.R., 2006. Honors calculus. Princeton University Press.

Medina, J., Hajbabaie, A., Benekohal, R., 2013. Effects of metered entry volume on an oversaturated network with dynamic signal timing. Transp. Res. Rec. J. Transp. Res. Board 53--60.

Mehrabipour, M., A.H., 2017. A Distributed-Coordinated Approach For Real-time Signal Control, in: Transportation Research Board 96th Annual Meeting Transportation Research Board.

Mehrabipour, M., 2018. Real-time Network-level Signal Timing Optimization. Washington State University.

Mehrabipour, M., Hajbabaie, A., 2022a. A Distributed Gradient Approach for System Optimal Dynamic Traffic Assignment. IEEE Trans. Intell. Transp. Syst. 1–15. https://doi.org/10.1109/TITS.2022.3163369

Mehrabipour, M., Hajbabaie, A., 2022b. A Distributed Methodology for Cell Transmission-based System Optimal Dynamic Traffic Assignment, in: Transportation Research Board 101th Annual Meeting Transportation Research Board.

Mehrabipour, M., Hajbabaie, A., 2020. A Distributed Gradient Approach for Cell Transmission Model-based System Optimal Dynamic Traffic Assignment, in: Transportation Research

Board 99th Annual Meeting Transportation Research Board.

Mehrabipour, M., Hajbabaie, A., 2017. A Cell-Based Distributed-Coordinated Approach for Network-Level Signal Timing Optimization. Comput. Civ. Infrastruct. Eng. 32, 599–616. https://doi.org/10.1111/mice.12272

Mehrabipour, M., Hajibabai, L., Hajbabaie, A., 2019a. A decomposition scheme for parallelization of system optimal dynamic traffic assignment on urban networks with multiple origins and destinations. Comput. Civ. Infrastruct. Eng. 34, mice.12455. https://doi.org/10.1111/mice.12455

Mehrabipour, M., Hajibabai, L., Hajbabaie, A., 2019b. A Decomposition Algorithm for System Optimal Dynamic Traffic Assignment on Urban Networks, in: Transportation Research Board 98th Annual Meeting Transportation Research Board.

Merchant, D.K., Nemhauser, G.L., 1978a. A model and an algorithm for the dynamic traffic assignment problems. Transp. Sci. 12, 183–199.

Merchant, D.K., Nemhauser, G.L., 1978b. Optimality conditions for a dynamic traffic assignment model. Transp. Sci. 12, 200–207.

Merchant, D.K., Nemhauser, G.L., 1976. A model and an algorithm for the dynamic traffic assignment problem, in: Traffic Equilibrium Methods. Springer, pp. 265–273.

Mirheli, A., Tajalli, M., Hajibabai, L., Hajbabaie, A., 2019. A consensus-based distributed trajectory control in a signal-free intersection. Transp. Res. Part C Emerg. Technol. 100, 161–176. https://doi.org/10.1016/j.trc.2019.01.004

Mohebifard, R., Bin Al Islam, S.M.A., Hajbabaie, A., 2019. Cooperative Traffic Signal and Perimeter Control in Semi-Connected Urban-Street Networks. Transp. Res. part C Emerg. Technol. 104, 408--427.

Mohebifard, R., Hajbabaie, A., 2021a. Trajectory Control in Roundabouts with a Mixed-fleet of Automated and Human-driven Vehicles. Comput. Civ. Infrastruct. Eng. Accepted.

Mohebifard, R., Hajbabaie, A., 2021b. Connected automated vehicle control in single lane roundabouts. Transp. Res. part C Emerg. Technol. 131, 103308.

Mohebifard, R., Hajbabaie, A., 2020. Effects of Automated Vehicles on Traffic Operations at Roundabouts., in: The IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC).

Mohebifard, R., Hajbabaie, A., 2019a. Optimal network-level traffic signal control: A benders decomposition-based solution algorithm. Transp. Res. Part B Methodol. 121, 252–274. https://doi.org/10.1016/j.trb.2019.01.012

Mohebifard, R., Hajbabaie, A., 2019b. Distributed Optimization and Coordination Algorithms for Dynamic Traffic Metering in Urban Street Networks. IEEE Trans. Intell. Transp. Syst. 20, 1930–1941. https://doi.org/10.1109/TITS.2018.2848246

Mohebifard, R., Hajbabaie, A., 2018a. Dynamic traffic metering in urban street networks: Formulation and solution algorithm. Transp. Res. Part C Emerg. Technol. 93, 161–178. https://doi.org/10.1016/j.trc.2018.04.027

Mohebifard, R., Hajbabaie, A., 2018b. Real-Time Adaptive Traffic Metering in a Connected Urban Street Network, in: Transportation Research Board 97th Annual Meeting. The National Academies of Sciences, Engineering, and Medicine, Washington DC, United States.

Mohebifard, R., others, 2021. Traffic Metering in Urban Street Networks.

Muñoz, J.C., Laval, J.A., 2006. System optimum dynamic traffic assignment graphical solution method for a congested freeway and one destination. Transp. Res. Part B Methodol. 40, 1–15.

Nedić, A., Olshevsky, A., 2014. Distributed optimization over time-varying directed graphs. IEEE Trans. Automat. Contr. 60, 601–615.

Nedic, A., Ozdaglar, A., Parrilo, P.A., 2010. Constrained consensus and optimization in multi-agent networks. IEEE Trans. Automat. Contr. 55, 922–938.

Nguyen, S., Dupuis, C., 1984. An efficient method for computing traffic equilibria in networks with asymmetric transportation costs. Transp. Sci. 18, 185–202.

Nie, X., Zhang, H.M., 2005. A comparative study of some macroscopic link models used in dynamic traffic assignment. Networks Spat. Econ. 5, 89–115.

Nie, Y.M., 2011. A cell-based Merchant--Nemhauser model for the system optimum dynamic traffic assignment problem. Transp. Res. Part B Methodol. 45, 329–342.

Nie, Y.M., Zhang, H.M., 2010. Solving the dynamic user optimal assignment problem considering queue spillback. Networks Spat. Econ. 10, 49–71.

Niroumand, R., Tajalli, M., Hajibabai, L., Hajbabaie, A., 2020. Joint optimization of vehicle-group trajectory and signal timing: Introducing the white phase for mixed-autonomy traffic stream. Transp. Res. Part C Emerg. Technol. 116, 102659. https://doi.org/10.1016/j.trc.2020.102659

Oregon Department of Transportation (DOT), 2004. The value of travel-time: Estimates of the hourly value of time for vehicles in Oregon 2003.

Pandey, V., Wang, E., Boyles, S.D., 2020. Deep reinforcement learning algorithm for dynamic pricing of express lanes with multiple access locations. Transp. Res. Part C Emerg. Technol. 119, 102715.

Peeta, S., Bulusu, S., 1999. Generalized singular value decomposition approach for consistent on-line dynamic traffic assignment. Transp. Res. Rec. J. Transp. Res. Board 77–87.

Peeta, S., Mahmassani, H.S., 1995. System optimal and user equilibrium time-dependent traffic assignment in congested networks. Ann. Oper. Res. 60, 81–113.

Peeta, S., Zhou, C., 2006. Stochastic quasi-gradient algorithm for the off-line stochastic dynamic traffic assignment problem. Transp. Res. Part B Methodol. 40, 179–206.

Peeta, S., Ziliaskopoulos, A.K., 2001. Foundations of dynamic traffic assignment: The past, the present and the future. Networks Spat. Econ. 1, 233–265.

Pi, X., Ma, W., Qian, Z.S., 2019. A general formulation for multi-modal dynamic traffic assignment considering multi-class vehicles, public transit and parking. Transp. Res. Part C Emerg. Technol. 104, 369–389.

Pigou, A.C., 1912. Wealth and welfare. Macmillan and Company, limited.

Polyak, B.T., 1987. Introduction to Optimization. Optimization Software. Inc., Publ. Div. New York 1.

Qian, Z.S., Shen, W., Zhang, H.M., 2012. System-optimal dynamic traffic assignment with and without queue spillback: Its path-based formulation and solution via approximate path marginal cost. Transp. Res. part B Methodol. 46, 874–893.

Qu, Y., Zhou, X., 2017. Large-scale dynamic transportation network simulation: A space-time-event parallel computing approach. Transp. Res. Part C Emerg. Technol. 75, 1–16.

Ramadurai, G., Ukkusuri, S., 2011. B-Dynamic: An Efficient Algorithm for Dynamic User Equilibrium Assignment in Activity-Travel Networks 1. Comput. Civ. Infrastruct. Eng. 26, 254–269.

Ramadurai, G., Ukkusuri, S. V, Zhao, J., Pang, J.-S., 2010. Linear complementarity formulation for single bottleneck model with heterogeneous commuters. Transp. Res. Part B Methodol. 44, 193–214.

Rickert, M., Nagel, K., 2001. Dynamic traffic assignment on parallel computers in TRANSIMS.

Futur. Gener. Comput. Syst. 17, 637–648.

Rockafellar, R.T., 1976. Monotone operators and the proximal point algorithm. SIAM J. Control Optim. 14, 877–898.

Rockafellar, R.T., 1970. Convex analysis. Princeton university press.

Rudin, W., 1976. Principles of mathematical analysis 3rd edn McGraw-Hill.

Saleh, A., Adeli, H., 1997. Robust parallel algorithms for solution of Riccati equation. J. Aerosp. Eng. 10, 126–133.

Saleh, A., Adeli, H., 1996. Parallel eigenvalue algorithms for large-scale control-optimization problems. J. Aerosp. Eng. 9, 70–79.

Saleh, A., Adeli, H., 1994. Parallel algorithms for integrated structural/control optimization. J. Aerosp. Eng. 7, 297–314.

Sbayti, H., Lu, C.-C., Mahmassani, H., 2007. Efficient implementation of method of successive averages in simulation-based dynamic traffic assignment models for large-scale network applications. Transp. Res. Rec. J. Transp. Res. Board 22–30.

Schade, J., Schlag, B., 2003. Acceptability of urban transport pricing strategies. Transp. Res. Part F Traffic Psychol. Behav. 6, 45–61.

Sharon, G., Levin, M.W., Hanna, J.P., Rambha, T., Boyles, S.D., Stone, P., 2017. Network-wide adaptive tolling for connected and automated vehicles. Transp. Res. Part C Emerg. Technol. 84, 142–157.

Shen, W., Nie, Y., Zhang, H., 2007a. Dynamic network simplex method for designing emergency evacuation plans. Transp. Res. Rec. J. Transp. Res. Board 83–93.

Shen, W., Nie, Y., Zhang, H.M., 2007b. On path marginal cost analysis and its relation to dynamic system-optimal traffic assignment, in: Transportation and Traffic Theory 2007. Papers

Selected for Presentation at ISTTT17.

Shen, W., Nie, Y., Zhang, H.M., 2006. Path-based system optimal dynamic traffic assignment models: formulations and solution methods, in: Intelligent Transportation Systems Conference, 2006. ITSC'06. IEEE. pp. 1298–1303.

Shen, W., Zhang, H.M., 2009. On the morning commute problem in a corridor network with multiple bottlenecks: Its system-optimal traffic flow patterns and the realizing tolling scheme. Transp. Res. Part B Methodol. 43, 267–284.

Shepherd, S., Sumalee, A., 2004. A genetic algorithm based approach to optimal toll level and location problems. Networks Spat. Econ. 4, 161–179.

Soegiarso, R., Adeli, H., 1994. Impact of vectorization on large-scale structural optimization. Struct. Optim. 7, 117–125.

Srivastava, K., 2012. Distributed optimization with applications to sensor networks and machine learning. University of Illinois at Urbana-Champaign.

Srivastava, K., Nedić, A., Stipanović, D.M., 2010. Distributed constrained optimization over noisy networks, in: Decision and Control (CDC), 2010 49th IEEE Conference On. pp. 1945–1950.

Su, P., Park, B.B., 2015. Auction-based highway reservation system an agent-based simulation study. Transp. Res. Part C Emerg. Technol. 60, 211–226.

Sumalee, A., 2001. Analysing the design criteria of charging cordons.

Szeto, W.Y., Lo, H.K., 2004. A cell-based simultaneous route and departure time choice model with elastic demand. Transp. Res. Part B Methodol. 38, 593–612.

Tajalli, M., Hajbabaie, A., 2021a. Traffic signal timing and trajectory optimization in a mixed autonomy traffic stream. IEEE Trans. Intell. Transp. Syst.

Tajalli, M., Hajbabaie, A., 2021b. A Lagrangian-based Signal Timing and Trajectory Optimization in a Mix Traffic Stream of Connected Self-driving and Human-driven Vehicles. IEEE Trans. Intell. Transp. Syst. 1–14. https://doi.org/10.1109/TITS.2021.3058193

Tajalli, M., Hajbabaie, A., 2018a. Dynamic Speed Harmonization in Connected urban Networks. Comput. Aided Civ. Infrastruct. Eng. an Int. J. In press.

Tajalli, M., Hajbabaie, A., 2018b. Dynamic Speed Harmonization in Connected Urban Street Networks. Comput. Civ. Infrastruct. Eng. 33, 510–523.

Tajalli, M., Mehrabipour, M., Hajbabaie, A., 2020. Network-Level Coordinated Speed Optimization and Traffic Light Control for Connected and Automated Vehicles. IEEE Trans. Intell. Transp. Syst. 1–12. https://doi.org/10.1109/TITS.2020.2994468

Tajalli, M., Mehrabipour, M., Hajbabaie, A., 2019. Cooperative signal timing and speed optimization in connected urban-street networks, in: Transportation Research Board 98th Annual Meeting Transportation Research Board.

Tajalli, M., Niroumand, R., Hajbabaie, A., 2022. Distributed cooperative trajectory and lane changing optimization of connected automated vehicles: Freeway segments with lane drop. Transp. Res. Part C Emerg. Technol. 143, 103761.

Tajtehranifard, H., Bhaskar, A., Nassir, N., Haque, M.M., Chung, E., 2018. A path marginal cost approximation algorithm for system optimal quasi-dynamic traffic assignment. Transp. Res. Part C Emerg. Technol. 88, 91–106.

Tomlin, J.A., 1971. A mathematical programming model for the combined distribution-assignment of traffic. Transp. Sci. 5, 122–140.

Tomlin, J.A., 1966. Minimum-cost multicommodity network flows. Oper. Res. 14, 45–51.

Ukkusuri, S. V, Han, L., Doan, K., 2012. Dynamic user equilibrium with a path based cell transmission model for general traffic networks. Transp. Res. Part B Methodol. 46, 1657–

1684.

Ukkusuri, S. V, Waller, S.T., 2008. Linear programming models for the user and system optimal dynamic network design problem: formulations, comparisons and extensions. Networks Spat. Econ. 8, 383–406.

Verhoef, E.T., 2002. Second-best congestion pricing in general networks. Heuristic algorithms for finding second-best optimal toll levels and toll points. Transp. Res. Part B Methodol. 36, 707–729. https://doi.org/10.1016/S0191-2615(01)00025-X

Vickrey, W.S., 1969. Congestion theory and transport investment. Am. Econ. Rev. 59, 251–260.

Wardrop, J.G., 1952. Some theoretical aspects of road traffic research, in: Inst Civil Engineers Proc London/UK/.

Wie, B.-W., Tobin, R.L., Friesz, T.L., 1994. The augmented Lagrangian method for solving dynamic network traffic assignment models in discrete time. Transp. Sci. 28, 204–220.

Wollmer, R.D., 1969. The Dantzig-Wolfe decomposition principle and minimum cost multicommodity network flows.

Wu, D., Yin, Y., Lawphongpanich, S., Yang, H., 2012. Design of more equitable congestion pricing and tradable credit schemes for multimodal transportation networks. Transp. Res. Part B Methodol. 46, 1273–1287.

Xie, C., Jiang, N., 2016. Relay requirement and traffic assignment of electric vehicles. Comput. Civ. Infrastruct. Eng. 31, 580–598.

Yahia, C.N., Pandey, V., Boyles, S.D., 2018. Network Partitioning Algorithms for Solving the Traffic Assignment Problem using a Decomposition Approach. Transp. Res. Rec. 0361198118799039.

Yang, H., Wang, X., 2011. Managing network mobility with tradable credits. Transp. Res. Part B

Methodol. 45, 580–594.

Zhan, X., Ukkusuri, S. V, 2019. Multiclass, simultaneous route and departure time choice dynamic traffic assignment with an embedded spatial queuing model. Transp. B Transp. Dyn. 7, 124–146.

Zhang, X., Yang, H., 2004. The optimal cordon-based network congestion pricing problem. Transp. Res. Part B Methodol. 38, 517–537.

Zheng, H., Chiu, Y.-C., 2011. A network flow algorithm for the cell-based single-destination system optimal dynamic traffic assignment problem. Transp. Sci. 45, 121–137.

Zhou, B., Bliemer, M.C.J., Bell, M.G.H., He, J., 2016. Two New Methods for Solving the Path-Based Stochastic User Equilibrium Problem. Comput. Civ. Infrastruct. Eng. 31, 100–116.

Ziliaskopoulos, A., Kotzinos, D., Mahmassani, H.S., 1997. Design and implementation of parallel time-dependent least time path algorithms for intelligent transportation systems applications. Transp. Res. Part C Emerg. Technol. 5, 95–107.

Ziliaskopoulos, A.K., 2000. A linear programming model for the single destination system optimum dynamic traffic assignment problem. Transp. Sci. 34, 37–49.

Ziliaskopoulos, A.K., Waller, S.T., 2000. An Internet-based geographic information system that integrates data, models and users for transportation applications. Transp. Res. Part C Emerg. Technol. 8, 427–444.