NORTHWESTERN UNIVERSITY

Dynamic Trucking Equilibrium in a Freight Exchange Platform with Hyperpath Routing

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILOSOPHY

Field of Civil and Environmental Engineering

By

John Vaala Miller

EVANSTON, ILLINOIS

September 2018

ABSTRACT

Dynamic Trucking Equilibrium in a Freight Exchange Platform with Hyperpath Routing

John Vaala Miller

This dissertation proposes a new hyperpath-based truck equilibrium assignment model with bidding. Unique from other trucking assignment models that examine assigning trucks for a large fleet of vehicles under a company's control, this model considers the case of many independent but homogeneous carriers that are competing for loads by bidding in auctions administered by an online freight exchange. The competitors follow optimal bidding hyperpaths to maximize their expected profit. The carriers all utilize an Online freight exchange (OFEX) platform that allows shippers to advertise loads to a large number of freight operators in search of loads. Such platforms often serve the purpose of matching demand and supply for freight in real-time.

The dissertation starts by developing a hyperpath truck routing problem that leverages the power of an OFEX platform. The OFEX routing problem seeks to determine a hyperpath in a space-time expanded network that maximizes the expected profit for a given origin-destination pair and tour duration. At the core of the OFEX routing problem is a combined pricing and bidding model that simultaneously (1) considers the probability

of winning a load at a given bid price and current market competition; (2) anticipates the future profit corresponding to the present decision; and (3) prioritizes the bidding order among possible load options. Results from numerical experiments, constructed using real-world data from a Chinese OFEX platform, indicate that the proposed routing model could (1) improve trucks' expected profits up to 400%, compared to the myopic and recursive benchmark solutions built to represent the state of the practice; and (2) enhance the robustness of the overall profitability against the impact of market competition and spatial variations.

The interference of the hyperpath routing model is that as more trucks utilize the method, the less advantageous the hyperpaths become due to the increased competition. Therefore, the dissertation continues to explain an equilibrium model that leverages the hyperpath truck routing solution to enable a solution for all trucks. This dissertation proposes a new hyperpath-based dynamic trucking equilibrium (DTE) assignment model. Unlike existing freight assignment models, this model focuses on the interactions between individual truck operators that solely compete for loads advertised on an online freight exchange. The competing trucks are assumed to follow optimal bidding and routing strategies - represented using a hyperpath - to maximize their expected profit. The proposed DTE model helps (1) predict system-wide truck flows (including empty truck flows), (2) identify efficiency improvements gained by network-wide visibility, and (3) lay the foundation for building a system optimal model. We rewrite the DTE conditions as a variational inequality problem (VIP) and discuss the analytical properties of the formulation, including solution existence. A heuristic solution algorithm is developed to solve the VIP, which consists of three modules: a dynamic network loading procedure

for mapping hyperpath flows onto the freight network, a column generation scheme for creating hyperpaths as needed, and a method of successive average for equilibrating profits on existing hyperpaths. The model and the solution algorithm are validated by numerical experiments constructed from empirical data collected in China. The results show that the DTE solutions outperform with wide margin the benchmark solutions that either ignore inter-truck interaction or operate trucks according to suboptimal routing and bidding decisions.

Acknowledgements

Completing this dissertation would not be a reality without the help of so many people. First, I give all of this work and time and any possible credit for anything good given to me to God. Second, I want to thank my wife, Katie, who has loved, helped, and supported me to follow my dreams while making sacrifices of her own to empower our family to thrive. She continues to amaze me with how she raises our children, Mae, Matthew, and Makaela, who I also need to thank for motivating me to be a better father, and a better person each day. I also need to thank my Mom and Dad, who never cease in supporting my family and I and helping us in every possible way.

I would like to give my utmost thanks to my advisor and professor, Marco Nie. Without his guidance, intellect, and work ethic to support our research, this work would still be on the ground. I have learned so much from his example about teaching, research, and academia from him. He always has a positive outlook and has a passion for research, and he motivated me through the Ph.D. journey.

I need to thank and my dissertation committee members of Professor Karen Smilowitz and Professor Amanda Stathopoulos for their time and expertise shared with me. I have learned much from them, and the research has refined from their influence.

I also need to thank the Guiyang Huochebang Technology Co Ltd. (Truck Alliance), for supporting our research, and for their desires to improve both their company and to help the truck drivers that support them.

I also need to give thanks to the students, faculty, and staff of the Northwestern University Transportation Center for the support, guidance, and fun during my time here. So many people made this time go too fast.

Last but not least, I need to thank the U. S. Air Force and the Department of Management at the Air Force Academy for sponsoring me to pursue my Ph.D. I cannot wait to return to the faculty and continue in research, learning, and to help build the next generation of leaders who never stop learning.

Finally, the views expressed in this article are mine and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U. S. Government.

Table of Contents

List of Tables

List of Figures

CHAPTER 1

Introduction

1.1. Motivation

Trucking is the backbone of freight transportation in many parts of the world. In the U.S., the trucking industry generated over $700 billion of revenue in 2015, which is more than 80% of the entire freight-related transportation bill [1]. In China, trucks moved about 31.5 billion tons of freight (roughly three-quarters of the total freight tonnage) in 2015 [72], with estimated revenue of nearly $1 trillion, based on the total cost of freight transportation in China [8]. The strong growth in e-commerce, expected to account for 17% of retail business by the end of 2022 [83], is continuing to drive up the demand for trucking. According to United States Department of Transportation in 2016 [78], by 2040 freight volume in the U.S. will increase by 45% to almost 30 billion tons.

One challenge to the efficiency of the trucking industry is the high fragmentation. According to the American Trucking Associations in 2015 [1], there are over 1.25 million for-hire and private carriers in the U.S., of which more than 90% operate six or fewer trucks. The fragmentation in China is even more significant. A survey conducted in 2014 showed that 85% of trucks are owned and operated by small for-hire carriers, with each carrier maintaining an average of 1.6 trucks [40]. Running in such an intensely competitive market is challenging, especially for the small carriers who have neither much bargaining power nor viable means for consolidation or collaboration. The key to surviving thus lies in the ability to discover high-value loads as quickly as possible and the flexibility to follow them wherever they arise. Small carriers are also at a disadvantage compared to big players when it comes to accessing load information across the entire network. Not surprisingly, when left to their own devices, these carriers often fail to match their assets to the most profitable loads in time, which not only hurts their profitability but also drags down the efficiency of the entire industry.

One of the promising solutions to breaking this information barrier, enabled by mobile computing and communication technologies, is the online freight exchange (OFEX) system. This system provides a platform that allows shippers to advertise loads publicly or privately to a large number of freight operators in search of loads. Like Uber for passengers, OFEX systems serve the purpose of matching demand (loads) and supply (truck capacity) for freight in real time. Well-known OFEX examples include Teleroute in France (http://teleroute.com/en_en, the first of its kind according to Wikipedia), Coyote in the US (www.coyote.com, acquired by UPS in 2015), 56QQ in China (www.56qq.cn, also known as Truck Alliance) and most recently Uber Freight (https://freight.uber.com/). The OFEX platforms provide a wide range of services. Some, such as Coyote and Uber Freight, mainly broker and execute transactions as an online third-party logistics provider. Teleroute and 56QQ, on the other hand, are primarily focused on advertising (for shippers) and consulting (for trucks) services. Either way, the fundamental question is *how to guide the carriers - most have no contractual relationship with the platform - to the most suitable loads, by leveraging on the platform's visibility of network-wide demand and supply information.* This question is at the core of OFEX's value-added services, and also what motivates this dissertation.

The question is framed in the context of inter-city truckload (TL) shipping. Currently, only about 11% of TL trucking is matched through OFEX systems in the United States but expected to continue rapid growth (Richardson 2017). An OFEX platform, operating in a closed shipping network consisting of a certain number of cities, is assumed to have access to detailed information of each load it ever recorded, such as origin-destination (O-D), weight, price, shipped date, etc. Suppose also that, based on this historical database, the OFEX could project for a future planning horizon: (1) the number of loads between any O-D pair at any given time, (2) the number of trucks looking for loads at each city at any given time, and (3) the distribution of load prices between any O-D pair. The question that can be asked is the following: *given the above information, what guidance could be given to a freight operator to maximize its profitability in a shipping tour with a given duration and starting/ending cities?*

Central to understanding the above question is the shipping imbalance inherent in any OFEX's network: some locations have more loads coming in than out (sink nodes), and others have more going out than in (source nodes). Sink nodes make the probability of getting a follow-on load from that location lower, in addition to driving the profits down on the few loads that are available. Source nodes make an almost certain probability for follow-on loads near that location, along with higher earnings on average. Utilizing these probabilities and profit expectations can allow a freight broker to recommend a bidding strategy for the next job. The policy is determined by calculating the maximum expected profit of all available loads and all follow-on decisions until the desired end of a tour (regarding both time and space). This method can significantly improve upon the current status quo where a truck strives to find the most profitable job available at the time, with little thought about the subsequent earnings effect of that decision. With a probabilistic recursive approach that optimizes profit over the whole route chain, a truck could significantly improve profitability.

Routing in an OFEX platform bears some similarities to the selective traveling salesperson problem (TSP) in that both problems aim to determine a set of cities to visit to optimize an objective function. However, there are several critical differences. One difference is the revenue at each city is not deterministic, but instead dependent upon the trucks' decisions. Trucks may decide to haul a load at the below-market price to a city, wait for a better deal in the next time interval, or move to a source node without carrying a load. More importantly, our OFEX problem endogenizes the competition in the market, through an embedded biding process that determines the winning probability.

The OFEX problem produces plans for trucks to follow to make as much profit as possible. However, it can be seen that the well-meaning guidance prescribed by the OFEX could be self-defeating. For example, a city that is usually a preferred stop, thanks to its relatively scarce trucking supply, may become flooded with incoming flows of trucks tipped by the platform, which in turn ruins the promise of demand surplus. For the platform that seeks to profit from the improved system efficiency, it is essential to anticipate and respond to the changing spatial-temporal supply-demand patterns caused by the interventions. Motivated by this need, this dissertation aims to propose a descriptive model of the OFEX system called the OFEX dynamic trucking equilibrium problem (DTE) problem. In this model, all (or a portion) of its registered trucks compete for loads while seeking to maximize their expected profit through dynamic routing and bidding.

1.2. Problem Statement

The DTE problem is defined as finding an equilibrium flow pattern of trucks over a network where all trucks utilize an OFEX platform that brings shippers and trucks together. The equilibrium occurs as optimal plans are identified, trucks change to those identified plans, which in turn decreases the expected profit. This decrease enables new plans to be more profitable, creating a cycle. We assume the cycle continues until an equilibrium state is reached, in which all hyperpaths with positive flow for an origin-destination (O-D) pair will have equal expected profit. We call finding this state the DTE. The OFEX platform operates in a region that consists of n cities. Each city pair i to j is connected by a route with a fixed travel time τ_{ij}. The ultimate goal is to create delivery plans for all trucks when each specifies (1) the origin city $\hat{\imath}$, (2) the destination city $\hat{\jmath}$, (3) the starting time and (4) the planning horizon T measured in the unit of operating interval Δ (e.g., a day or an hour). With these inputs defined, the model assigns the trucks to the delivery plans in a way such that all trucks' expected profit is equal for each strategy available for trucks with the same starting ending locations and times.

1.3. Contributions

With the aspiration of constructing and putting into practice a planning tool to aid truckers and shippers utilizing an OFEX platform, we make the following significant contributions in this dissertation:

(1) Develop a hyperpath-based truck routing model that is based on optimized bidding within an OFEX platform. Current planning and truck allocation models assume full control of the trucks or include constraints that all jobs must be filled

or penalties are incurred. The OFEX routing model formulation provides a way to generate robust routing plans with full recourse for truckers in addition to being very fast even for realistically sized networks.

(2) Demonstrate the practical significance of the methodology on specific instances using real OFEX data from China. The OFEX routing model generates the expected optimal profit routing hyperpath and bidding strategy for a single truck in an online freight exchange platform that can increase earnings by over 200% from the status quo. Furthermore, the DTE problem shows the network could raise profits for all trucks by 10%, potentially increasing profits by hundreds of millions of dollars in China's highly fragmented trucking network.

(3) Establish a mathematical foundation for a new research area ripe for application and analysis with potential for major economic and environmental impacts. Research can build upon this foundation to analyze many other networks: from a localized Uber driver network to examine efficient and fair time-dependent pricing in a city, to a national shipping network to find the optimal place for a new distribution center to alleviate shipping imbalance, many possibilities exist.

1.4. Organization

The dissertation structure is designed to develop the elements necessary to solve the trucking equilibrium problem for a real OFEX. Chapter 2 provides a thorough literature review of the related routing problem and the dynamic user equilibrium problem literature. Chapter 3 presents a formal statement of the OFEX routing problem with bidding and introduces necessary notations, assumptions, and models. Chapter 4 builds upon the

OFEX routing problem and presents a formulation for the dynamic user equilibrium problem, to include the dynamic loading algorithm necessary to produce the network results from hyperpath strategies. Chapter 5 presents the data, numerical analysis of the OFEX routing and equilibrium problems. Chapter 6 concludes the dissertation with a summary of the contributions, and equally valuable, the potential areas of future research.

CHAPTER 2

Literature Review

This chapter introduces the literature of some models relevant to the OFEX routing problem and transportation models related to the dynamic user equilibrium problem. The trucking industry is a fundamental part of transportation research, and many papers and models have been created in this field. However, previous research does not incorporate all of the types of objectives, constraints, and perspectives that are critically important to the problem defined in this research. This chapter describes the relationships between the different types of formulations related to the OFEX routing problem and dynamic trucking equilibrium problems.

2.1. OFEX Routing Problem Literature Review

The OFEX routing problem is a transportation optimization problem with four main features: (1) selective profit-maximizing routing, (2) dynamic pickup and delivery, (3) bidding in an online auction for probabilistic (or stochastic) service requests, and (4) solving for a set of attractive routing and pricing strategies, referred to as hyperpaths. While problems with individual or some mixtures of multiple features have unquestionably been studied before, the full combination of all aspects is new, to the best of our knowledge.

Problems in the literature that solve vehicle flows on arcs and nodes are known as vehicle routing problems. Many variations and formulations of routing problems exist. The goal of this research is to utilize a formulation that handles the elements of time windows, single and multiple vehicles, multiple depots, open routes, stochasticity, selective pickup for a pickup and delivery, and the ability to be solved and resolved quickly to account for a dynamically changing environment to maximize the benefit received. The aim is to accurately classify the OFEX routing problem in research to know what work has been previously completed and what methods need to be explored in solving the problem.

Routing problems are known as classic and fundamental operations research problems. The OFEX routing problem is therefore related to this classical research stream, including the traveling salesperson problem (TSP).

2.1.1. Traveling Salesperson Problem

Our research problem is related to a class of problems known as the Traveling Salesman Problem (TSP). TSPs also aim to optimize an objective function with a routing plan.

The standard traveling salesperson problem requires the solution to visit each node of a set of nodes, or cities, for example, exactly once, initializing from, and returning to the starting location [70]. The objective is to minimize the total distance traveled by the salesperson. To solve these problems when the number of nodes becomes large is very difficult. The known exact algorithms that search the whole solution space can be as difficult as exhaustive search in the worst case. Known approximate algorithms avoid searching the whole solution space and attempt only to find a solution that is within a certain proximity of an optimal solution. It is a mathematical conjecture that the complexity class of the TSP is Nondeterministic Polynomial-time Complete or *NP-complete*. A problem is in class NP if it is possible to check in polynomial time whether a “lucky guess” is actually an optimal solution. For a given directed graph with edge costs, the problem of determining whether there is a path of cost c (for any given c) is in NP - if a path is supplied, it is possible to check in polynomial time whether the cost of that path is c. Any TSP can be solved by determining a sequence of these "does there exist a path of cost c" special problems by choosing the c's as a binary sequence. A problem is *NP-complete* if whenever there is an engine that solves this problem in polynomial time, you can use this engine as part of an algorithm that solves any other problem in class *NP* in polynomial time. This means that it is not possible to find a solution algorithm that gives an optimal solution in a time that has a polynomial variation with the number of nodes n, as in $n * x$. The best solution currently available is to solve the problem in a way that grows exponentially with the number of nodes, as in 2^n. This challenge makes solving TSP problems intractable when n is very large. Because the TSP is a *NP-complete* problem, it is nearly impossible to solve optimally for a large n. For example, a complete graph with n vertices requires $(n-1)!$ ways to choose a circuit of length n. As n becomes large, the number of possible paths explodes.

Algorithms do exist that are usually good at solving the TSP precisely, especially for smaller-sized problems. The branch and bound algorithm is frequently used to solve the TSP exactly. However, in the worst case, branch and bound is equal to exhaustive search. The origin of the branch and bound algorithm goes back to the work of Dantzig, Fulkerson, and Johnson [13]. The branch and bound method solves a discrete optimization problem by breaking up its feasible set into successively smaller subsets, calculating bounds on the objective function value over each subset, and using them to discard specific subsets from further consideration.

The TSP is a good starting point for exploring models and the corresponding properties, but the OFEX routing problem has significant differences from the standard and TSP variants. The OFEX routing model has revenues at each city that are not deterministic, but instead dependent upon the trucks' decisions. More importantly, our OFEX problem incorporates the competition in the market, through an embedded biding process that determines the winning probability, very different than the TSP. Finally, the OFEX routing problem contains profit maximization, while the TSP is a cost minimization.

2.1.2. Orienteering Problem

Another area of research related to profit maximization routing falls under the name of the orienteering problem (OP). While the OP was named for and applied to the sport of orienteering, it has practical applications in vehicle routing with profits as discussed by Golden et al. [30]. They also proved that the OP is *NP-hard*. Work has been done on exact methods for the OP such as integer programming, dynamic programming, and branch-and-cut algorithms. Although these approaches have yielded solutions to smaller sized problems, as in other *NP-hard* problems, the computational limitations of exact algorithms encourage the exploration of heuristic procedures in the OP literature as well. The Team Orienteering Problem (TOP) is the multiple vehicle extension to the OP. In the multiple vehicle case, this relates to the classic vehicle routing problem classification. The TOP framework is closer related work to the OFEX routing problem but does not capture the probabilistic or pickup and deliver nature to the question of interest for this dissertation.

2.1.3. Probabilistic Traveling Salesperson and Orienteering Problems

A research problem that accounts for the stochastic nature of the problem is the probabilistic traveling salesperson problem (PTSP). The PTSP is a stochastic extension of the TSP where each node will be visited with a certain probability. Jaillet in 1988, [36] is known for creating the original formulation of the PTSP. Laporte et al. [39] formalized the methodologies and analysis of the PTSP in 1994. Recently, building upon the PTSP body of work, Angelelli et al. [2] developed the Probabilistic Orienteering Problem (POP). They formulate the POP as a linear integer stochastic problem and establish heuristics to solve many instances of the problem and develop a recourse function to handle the stochastic nature of finding points that do not materialize. While the stochastic elements are present in these models, the pickup and delivery and bidding aspects are not included in these models.

2.1.4. Pickup and Delivery Problem

The literature mentioned so far has not covered the vehicle pickup and delivery (PDP) aspect of the desired formulation. PDPs are broken into three main categories, many-to-many, one-to-many-to-one, and one-to-one. The OFEX routing problem is similar to the pickup and delivery problem (PDP) - which is a special case of the vehicle routing problem (VRP) [16] - in that each load is identified by a pickup (origin) and a drop-off (destination) location and that the optimal decision involves a tour of multiple loads. Because the literature of VRP/PDP is quite extensive, no attempt is made here to provide a comprehensive review. The reader is referred to Ritzinger et al. [66] and Ulmer [77] for recent reviews.

Recent and relevant research in this area begins with Yang et al. [84] who consider a dynamic multi-vehicle PDP that incorporates the penalty cost on the delayed completion and job rejections. Using a rolling horizon framework, they propose re-optimization strategies (by solving a mixed integer programming formulation) and local dispatching policies for real-time operations. In 2004, Yang et al. [85] proposed a new optimization-based policy for the dynamic multi-vehicle PDP and developed a simulation platform to test its effectiveness. Within a similar dynamic PDP framework, Mitrović-Minić and Laporte in 2004, [49] examined the problem of organizing and distributing a vehicle's idle time in its schedule. They proposed and compared four different waiting strategies and concluded that the optimal waiting strategy depends on proper route partition. Figliozzi et al. [24] introduced the VRP in a competitive environment, where they utilize a stochastic dynamic programming model to solve the problem for a carrier bidding on arriving loads. Built on Yang et al.' work from [84] and [85], Zolfagharinia and Haughton in 2014, [88], introduced the notion of a truck's home base when designing dispatching rules. Their model can also handle load rejection, truck diversion, and advance load information. They further examined the effect of incomplete advance load information and proposed a simple but effective heuristic policy that is integrated with a mixed integer programming formulation [89]. More recently, they analyzed the value of advanced load information, decision interval, and the diversion capability in the context of dynamic PDP for truckloads [90]. They found that the advance load information and decision interval significantly influence the total cost. Lastly, in 2017, Qui et al. [65] and Gansterer et al. [29] consider profitable PDP problems, which decide not only how to route and schedule vehicles to serve the loads, but also which loads to accept or reject.

2.1.5. Vehicle Assignment Problem

Our problem is also similar to the dynamic vehicle allocation problem (VAP). In its most straightforward form, such a problem assigns trucks to meet the demand for moving loads between cities within a region, which can be reduced to a transshipment problem when the future requirements are known with certainty, as in 1969 and 1972 models by White [81, 80]. In 1984 [63], Powell et al. introduced uncertainty into the demand but assumed the vehicle flows are determined before the actual realization of the demand. This assumption was relaxed by Powell in 1986 [57], who proposed to differentiate empty truck flows from those intended to move with loads. Accordingly, the actual number of loaded truck flows between any O-D pair would be bounded by the realized demand in the model. In 1987 [58], Powell suggested a fundamentally different approach to managing the fleet in a real-time truckload trucking environment. The problem posed was to optimize load acceptance/rejection and truck movements for the current condition, while using the historical data to anticipate the consequence of a decision made now. The key is to include the so-called "regional impact" in the current decision, or the expected value of sending a vehicle to a region, which may depend on all the possible future movements once the vehicle arrives at the region. Powell et al. later implemented this real-time VAP model in [62] as LOADMAP and deployed by the Commercial Transport Division of North American Van Lines. In 1996 [59], Powell extended the above framework by explicitly integrating the forecasted demand into the model, and confirmed that this stochastic, dynamic model with predicted demand outperforms standard myopic models. In 2000 [64], Powell et al. compared two solution strategies for dynamic vehicle assignment: a deterministic myopic re-optimization based on the data revealed up to the decision time (as in Yang et al. [85]) and local greedy-type heuristics. They found that in the presence of uncertainty (especially user noncompliance) optimal solutions to static subproblems offer small to negative benefits compared to crude local heuristics. In 2006, Tjokroamidjojo et al. revisited the dynamic VAP and proposed a formulation revised from an aircraft scheduling model [74]. Their focus was to quantify the value of knowing the requests in advance. In 2007, Topaloglu and Powell [75], incorporate the pricing decision of a carrier into the stochastic VAP. The pricing decision in this dissertation is fundamentally different because, in Topaloglu and Powell, the carrier is determining a price, which influences how many loads arrive, while in this dissertation, an auction determines the price from competing carriers, with no bearing on the arrival of loads.

2.1.6. Agent-based Control Models

Most studies reviewed hitherto are concerned with a centralized decision problem of assigning a fleet of trucks to most valuable loads (in the case of dynamic vehicle allocation problems) or optimal tours (in the case of dynamic PDP). In this dissertation, we conceptualize the problem from the perspective of an individual trucker who attempts to plan for a tour based on historical information observed from an OFEX platform. In the literature, the line of work that resembles this approach most closely was initiated by Mes et al. [47], who model truckload operations as a multi-agent system of individual truck operators and job dispatchers. Each truck operator competes for loads by submitting bids to a job dispatcher, who evaluates all bids and decides which one to accept. A bid is priced by each operator "on the fly" according to the difference in cost between the updated schedule (with the new load) and the current schedule. When compared to a centralized control system, the multi-agent system was found as equally or more efficient. In a follow-up study [46], Mes et al. extended the truck operator's decision problem, attempting to take into account not only the direct cost of a job insertion (as in [47]) but also the potential gains and costs that may arise within a look-ahead period. Mes et al. continued to focus on the truck's "next optimal move" given the information of available loads and its current work schedule [46]. However, no attempt was made to build an optimal tour plan that consists of multiple moves for a planning horizon. Therefore, their modeling approach's ability to anticipate the effects of possible future events is limited, leading to potentially myopic dispatching and bidding decisions.

2.1.7. OFEX Problem Formulation Considerations

The truckload routing and scheduling problems discussed above are typically formulated and solved either as a mixed integer program (MIP) or a (stochastic) dynamic program (DP). In this dissertation, we formulate the OFEX routing problem as a DP because it allows us to build recourse into strategic planning through dynamic recursion. Instead of a fixed tour, our model generates a series of routing and bidding strategies represented as a hyperpath over a space-time expanded network. The hyperpath is determined a priori but contains adaptive decisions. Specifically, while the decision at any time is made by anticipating the aggregate effects of all possible future decisions, the actual trajectory of a truck in the network is unknown until the strategies are executed.

The concept of hyperpath routing originates from the common bus lines problem from Chriqui and Robillard in 1975 [9], in which passengers strategically select the first arriving bus from a set of attractive lines to minimize the total expected travel time. Spiess and

Florian extended the idea in 1989 to a general transit network [73], and Nguyen and Pallottino in 1988 [52] formally described such a strategy as a hyperpath. To the best of our knowledge, most hyperpath applications are related to public transportation, see Trozzi et al. 2013 [76], Li et al. 2015 [41], and Chen and Nie 2015 [7], but few have considered it in truckload trucking.

2.2. OFEX Trucking Equilibrium Problem Literature Review

The routing problems and solution methodologies covered so far are good for optimally routing single or multiple vehicles when the probabilities are static or independent of the vehicles' actions. However, as more vehicles begin to optimize expected profits, and move to the same optimal paths, the competition creates more demand for the initially optimal locations, and those probabilities and the corresponding expected revenues would decrease as the number of vehicles at those locations increase. Borrowing the ideas and principles from John Wardrop and the field of traffic assignment research, it is possible to find this type of equilibrium in a profit maximization case as well.

The DTE problem is a transportation problem with three main features: (1) solving for a set of attractive routes and routing strategies, referred to as hyperpaths, for trucks within an OFEX, (2) loading flows of trucks onto a network from hyperpath flows, and (3) finding a network trucking equilibrium by balancing the truck flows across hyperpaths. The DTE problem mimics the traffic assignment problem (TAP), from Wardrop's seminal paper in 1952 [79] and Beckman et al.'s in 1955 [3], in that the equilibrium may be viewed as the result of two competing processes. On the one hand, like travelers seeking shortest paths in TAP, the trucks make profit-maximizing decisions based on predicted information about the distribution of load/trucks in the network, as well as shipping prices. On the other hand, like travelers inducing congestion, each truck's decision affects his/her expected profits by shifting the demand/supply balance in each spot market, which changes not only the market price for loads but also the likelihood of winning a bid. While characterizing this equilibrium behavior is interesting in its own right, we note that understanding how interventions may impact system efficiency and profitability is also crucial to the OFEX's value-added services.

2.2.1. Time-Dependent Shortest Path Problems

The analysis of transportation networks and finding equilibrium solutions to date has required the computation of shortest paths. The shortest path problem is one of the most fundamental problems in computer science and combinatorial optimization. In the basic variants of the problem, it has very efficient algorithms to solve it, including dynamic programming, Dijkstra's algorithm [17], and the Bellman-Ford algorithm [4] and [25]. For the research in this dissertation, the time-dependent case is considered because the probabilities at the locations change based on time of day. Furthermore, this adds the ability to account for the length of travel time to change over time, adding realism to the problem. Also, the time-dependent formulation enables the use of a directed acyclic graph (DAG) which to calculate and solve the maximum profitable path, (or longest path problem) is a necessary condition for solving the problem in polynomial time. The algorithm has to be as efficient as possible to be able to calculate optimal routes dynamically.

2.2.1.1. Generalizing the Static Shortest Path Problem. The initial paper covering the time-dependent shortest path problem was in 1966 by Cooke and Halsey [11].

Following in Bellman's principle of optimality [4], they developed an algorithm that calculated the shortest path from every node in the network to one destination node for a set of discrete departure time steps. The proposed algorithm has a time complexity of $O(V^3M)$, where V is the number of vertices (nodes) and M is the number of discrete time units. Later, in 1969, Dreyfus [19] built upon Dijkstra's [17] static label-setting algorithm. Dreyfus generalized the static case that could calculate the time-dependent shortest path between two nodes for one departure time step with the same complexity as the static case $O(V^2M)$ but has the same complexity as Cooke and Halsey in the case for all the nodes.

Orda and Rom, in 1990, [53] studied the problem accounting for waiting-at-nodes scenarios, and developed algorithms to solve these cases. They proved that if waiting is allowed at nodes, then the FIFO property is not required. Building upon these efforts, in 1993, Ziliaskopoulos and Mahmassani [87] provided an efficient solution approach to the problem of the FIFO property by the discretization of time into τ time periods and developed a pseudo-polynomial time algorithm for the all-to-one case having a complexity of $O(m^3\tau^2)$, for a network having m source nodes. This algorithm can solve for the minimum cost paths in networks with general link travel costs that do not necessarily satisfy the FIFO property, for a given set of starting times.

2.2.1.2. Decreasing Order of Time Algorithm. In 1998, Chabini analyzed the different types of problems and developed the decreasing order of time (DOT) algorithm [6]. Chabini proposed that in the case of all travel times are represented as positive integers, the labels corresponding to time step t never update labels corresponding to time steps greater than t. This insight implies the acyclic property along the time dimension of the time-space expansion of a discrete dynamic network. Chabini shows that the time complexity of the DOT algorithm is $O(M(V^2 + V + m))$, where m is the number of arcs. He also proves that this method is the optimal worst-case-run-time complexity for the all-to-one shortest path problem.

2.2.1.3. Dynamic Longest and Profit Maximizing Paths. None of the papers discussed have explored the case of finding the longest path or profit maximized route in a time-dependent network. This area of research remains mostly unexplored for two reasons: first, the longest path problem in the general case is *NP-hard* as shown [68]. If it could be solved in polynomial time, the methodology could be used to solve the traveling salesperson problem. Therefore, less effort has been placed on researching the longest path problem and has been limited to exceptional cases when the Hamiltonian Path Problem can be solved in polynomial time. Second, in the case of problems with a directed acyclic graph structure, it is possible to transform the graph from the longest path problem to an equivalent shortest path problem by changing every length or cost in the network to its negation, and applying the algorithms of the shortest path problem.

A typical example of this methodology in practice is determining the critical path from project management. The longest path signifies the critical path of the project and represents the fastest time possible for project completion. For the problem of finding optimized profit routes, this dissertation will focus on DAG networks. Very little research can be found for the dynamic or time-dependent maximum profit path problem. Therefore, a plan for solving the problem is to utilize the extensive research from the time-dependent shortest path problem in DAGs and apply the algorithms inversely to account for profits or the longest path in a time-dependent DAG network.

2.2.2. Static User Equilibrium in the Traffic Assignment Problem

The intensely studied traffic assignment problem studies the interactive process between the traveler demand and the transportation supply. The supply is the transportation infrastructure, and the demand is the travelers desire to move from point to point within a network. This supply and demand interaction creates an equilibrium point that Wardrop specified in his first principle in 1952 [79]. Wardrop also identified a second principle that states there is also a state in which the total users' travel cost is minimized concerning the choice of routes. This research utilizes these same principles, but for profit instead of cost. With the increase of available data to predict future demand for freight shipments, truckers have the newfound ability to plan routes to maximize their expected profit intelligently. However, just as in the traffic assignment problem, as more trucks follow the same profit-maximizing techniques, the routes that were initially the most profitable become less profitable with increased competition. However, since very little applicable research is currently found for equilibrium problems with profits, this section will look at the research related to the traffic assignment problem.

In 1955, Beckmann et al. formulated the static user equilibrium with fixed demand as an optimization problem to find equivalent routes [3]. Sheffi, in 1985, wrote a book with detailed analysis and the standard for the static UE and the stochastic UE problems [69]. Many algorithms and models have been developed to solve the static user equilibrium problems. The algorithms' ability to converge to the UE solution depends on the problem structure. Patriksson defines three favorable structures to include origin-destination (O-D) pair separability, cost separability, and the primal-dual relationship [54]. When models have these favorable structures, then the use of a decomposition algorithm, where the algorithm fixes all O-D flows except the one being updated is prudent. This situation exploits the stronger monotonicity structure on the smaller subspace being updated than on the entire network.

2.2.3. Dynamic User Equilibrium in the Traffic Assignment Problem

While the static user equilibrium provides many valuable insights, the assumptions of the static UE problem were too restrictive for modeling real-time traffic. This issue sparked interest in researching the dynamic traffic assignment problem and the dynamic user equilibrium. A time-dependent network had to be utilized, to account for the changing demand rate. The early analytical formulations of the time-dependent user equilibrium model extended Wardrops condition to departure time choice as well as the static conditions for unutilized routes. The first known attempt to formulate the dynamic traffic assignment problem was by Merchant and Nemhauser in 1978 [44] and [45].

Another class of formulations for the dynamic UE problem assumes the knowledge of O-D demands a priori to running the model. This assumption effectively removes the time choice from the model to make it tractable. The earliest instance in this area was by Janson in 1991, [37]. However, the solutions to these types of models can lead to unrealistic travel behavior in some cases due to the reliance on static link performance functions and static O-D demand.

To overcome the weaknesses and simplifications of the dynamic TAP models, simulation-based models became a prevalent methodology to solve the dynamic UE problem. Now they are the most widely used models in practice because of the greater realistic representation of traffic within the simulated network. Mahmassani and Peeta developed a

DTA model that used a mesoscopic traffic simulator, called DYNASMART [55], which has gone through many enhancements since creation. There are several other similar simulation-based dynamic traffic assignment models, to include DynaMIT, CONTRAM, and METROPOLIS to name a few. A quick summary of the foundational simulation-based models is found in [56]. Most of the advantage of simulation-based models is the utilization of advanced realistic traffic simulation. The dynamic model for the profit-maximization model proposed in this research is a more extended scale model that is less concerned about the realistic traffic but could utilize simulation to model a more realistic bidding process that is not possible with a mathematical model implementation.

2.2.4. Freight Network Equilibrium

In the literature, the intercity freight flow equilibrium problem is typically concerned with the interactions between shippers (demand side) and carriers (supply side). A classical example is the Freight Network Equilibrium Model (FNEM) [26], which assumes that shippers compete for the service of carriers non-cooperatively and carriers maximize profit by choosing the production of transportation services. The shipper-carrier interaction may be modeled sequentially (as in FNEM), or simultaneously. The latter approach often treats the shipper's problem as an equilibrium constraint in the carrier's cost optimization problem, which leads to a non-convex bi-level formulation [see, e.g., 27, 28].

In a more general setting, [34] formulate the demand for shipping as a result of equilibrating spatial commodity prices. [35] focus on carriers' pricing strategies in freight network equilibrium and show that the most efficient flow distribution can be decentralized if carriers use the optimal nonlinear tariffs. More recently, [23] revisited the FNEM problem and proposed an integrated shipper-carrier equilibrium model that is similar in many ways to the combined mode choice/trip distribution/traffic assignment model [69].

The proposed DTE model fundamentally differs from these existing freight equilibrium models in that it takes the perspective of an OFEX platform that primarily functions as a marketplace for infinitely many small shippers and truckers. No player, including the platform itself, has the power to dominate the market. The proposed model also captures highly disaggregated time-dependent decisions of individual trucks, such as which loads and at what price to bid, whether and how long to wait for a desirable load at a given stop, and where and when to move empty to the next city. Accordingly, it offers mechanisms to endogenize the movement of empty trucks (as part of the trucks' decision making) and load pricing (as a result of the competitive bidding in the market).

Finally, we note that hyperpath (or strategy) is not new to equilibrium analysis [see, e.g., 14, 82, 10, 43, 31, 33], but few had considered it in the context of freight flow equilibrium.

2.3. Summary

This chapter provided a review of the relevant literature for the OFEX routing and DTE equilibrium problems proposed in this dissertation. Both of these research questions have aspects that have been heavily researched, but both have elements that make them new problems. The main components not addressed in the literature include the combination of utilizing an OFEX to create routing strategies with optimal bidding plans and recourse, referred to as hyperpaths for trucks. Furthermore, research gaps exist for modeling a trucking freight network equilibrium by balancing the truck flows across the generated hyperpaths. The following chapters plan to fill these gaps with the purpose of creating a new OFEX routing model and new OFEX equilibrium model to improve trucking operations.

CHAPTER 3

OFEX Routing Problem

An online freight exchange platform provides a modern auction form. Trucks bid for loads advertised on the platform, typically from a smartphone, and the low bidder wins. Traditionally, independent owner-operator trucks use these services when trying to fill their schedule with loads, or try to carry at least some load on a back-haul to the next destination. However, as the services have grown in popularity, so too did the utility of using the service full-time. As trucks utilize the OFEX platform exclusively, the opportunity arose to create a bidding strategy for loads that also accounts for the impact of that decision on the potential of future earnings, along with all recourse. Developing this ability could potentially give the power of network visibility that large carriers possess to independent trucks. This chapter explains how these routes and bidding strategies can be generated.

3.1. Problem Statement

This section explains the stochastic dynamic decision problem that represents the decisions for a single independent truck and the with an example problem, introduces some assumptions to scope the problem for this dissertation, and introduces the chosen solution representation. Consider a truck that utilizes an OFEX platform that brings shippers and trucks together to acquire loads. Shippers can post information about their loads, to which the truck has access, both through the platform. The truck operates in a region that consists of n cities. Each city pair i to j is connected by a route with a fixed travel time τ_{ij}. When the truck is at i, it is offered a set of loads represented by $\mathcal{Q}$ that the truck can bid to move from i to j at time t. The truck may also reject all loads and stay in place until time $t+1$, or move empty to any node j. The goal is to create a bid order and pricing plan for the truck when it specifies (1) the origin city $\hat{i}$, (2) the destination city $\hat{j}$, (3) the starting time and (4) the planning horizon T measured in the unit of operating interval Δ (e.g., a day or an hour). A table of notation for reference is given in Appendix A.

3.1.1. Markov Decision Process Model

The problem can be generically modeled as a stochastic dynamic decision problem following similar notation set by Powell, 2007 [60]. The stages are represented by the time increment t, the state variable S_t is the current location of the truck, and the set of decisions, $\mathbf{x} \in X$, represent the order of loads to bid and for what price. Specifically, $\mathbf{x} = (x, \phi)$ where x_{ij}^t = bid price for load from node i to node j at time t (= 0 for all empty moves and waiting option) and ϕ_x^t = ranking of all feasible x moves. The objective is to maximize the expected vehicle profit measured as the difference in revenues collected, x_{ij}^t, and delivery costs c_{ij}^t, represented by σ. The general optimization problem can be represented as:

$$\max_{\pi \in \Pi} \mathbb{E}^{\pi}\{\sum_{t=0}^{T} \sigma(S_t, X_t^{\pi}(S_t))|S_0\}, \tag{3.1}$$

where decisions are made according to a policy, represented as π. The transition function depends on the state and information pre-decision $(S_t, \bar{W}_t)$ and post-decision (denoted

$S_t^{\mathbf{x}}, \bar{W}_{t+1})$ which is the state and information immediately after making a decision. This relationship is represented as: $S_t^{\mathbf{x}} = S^{M,\mathbf{x}}(S_t, \mathbf{x}_t)$ and the transition function is $S_{t+\tau_{\mathbf{x}}} = S^{M,\bar{W}}(S_t^{\mathbf{x}}, \bar{W}_{t+1})$, where $\bar{W}_{t+1}$ is the information that becomes known at time t, after decision $\mathbf{x}$, which for this problem represents if a bid for a desired load is won or lost for a given bid price and order. The superscript M represents the generic "model" of information, as common in the literature. Then assume there is a probability transition matrix $p(s'|s, \mathbf{x}) = P[S_{t+\tau_{\mathbf{x}}} = s'|S_t = s, \mathbf{x}_t = x]$ for $s, s' \in S$. Since $\bar{W}_t$ depends on the decisions $\mathbf{x}_t$, specifically the bid price, and the state information S_t, in Equation 3.1 includes $\mathbb{E}^{\pi}$ to represent this dependency. Also note, that the transition time depends on the travel time of the specific decision x, so the travel time can be greater than 1 period.

The constraints to Problem 3.1 include restrictions on the bid value, $x \in \mathbb{R}$, bid order $\phi \in \mathbb{Z}$, and only one loaded or empty move can be completed at each stage, $X^{\pi}(S_t) \in X_t$. In general, the problem can be solved with Bellman's equation:

$$V_t(S_t) = \max_{\mathbf{x}_t \in X}(\sigma(S_t, \mathbf{x}_t) + \sum_{s' \in S} p(s'|S_t, \mathbf{x}_t) V_{t+\tau_{s'}}(s')). \tag{3.2}$$

where one can assume $V_T(S_T) = 0$ and solving backward in time. Once all $V_t(S_t)$ are computed for all times t and states $S_t \in S$, a policy can be generated by:

$$X_t^*(S_t) = \arg\max_{\mathbf{x}_t \in X}(\sigma(S_t, \mathbf{x}_t) + \sum_{s' \in S} p(s'|S_t, \mathbf{x}_t) V_{t+\tau_{s'}}(s')). \tag{3.3}$$

The challenge to solving this problem 3.2 in a practical way is to find a pragmatic way to represent the probabilities of transitioning from winning or losing bids that are dependent on the bid price, order, and competition behavior. The remainder of the dissertation discusses techniques to handle the dependent relationships. The first step is by presenting the operating assumptions of the solution methodology.

3.1.2. Assumptions

We shall assume that the following inputs to the OFEX routing problem can be reliably predicted between now ($t = 0$) and $t = T$, based on historical information:

- Number of loads scheduled to depart from i at $t = 0, \cdots, T$ destined for city j, denoted as g_{ij}^t;
- Number of vehicles available to carry loads at each city i at $t = 0, \cdots, T$, denoted as H_i^t; and
- Cumulative distribution function of the unit load price between any city pair ij, denoted F_{ij}^t.

We further make the following assumptions about the operations of the OFEX platform in order to simplify the problem setting.

(1) All decisions are tied to an operating interval Δ. Hence, all travel times τ_{ij} are rounded up the next Δ.

(2) Drivers are limited to 12 hours of work per day including waiting, with a working time window between 6:00 AM to 6:00 PM.

(3) All loads are advertised on the OFEX platform and executed without breaks within the working time window. Therefore the pickup time window is strictly enforced once a truck wins a bid, and consequently, the delivery time windows are never violated.

(4) The cost of traveling loaded or empty is a linear function of travel time, which is a linear function of distance, similar to [49].

(5) The truck only carries full truckloads, one at a time.

(6) Handling (loading and unloading) takes a fixed amount of time. In this dissertation, it is set to 2Δ (one for loading and one for unloading) unless otherwise specified.

The set of real and dummy cities in STEN is denoted as $\mathcal{L}$ and $\mathcal{L}'$ respectively. There is a dummy connector from i' to i at each time interval, relocating empty trucks to the real city so that they can be available for loads. The set of dummy connectors in STEN is denoted as $\mathcal{A}_d$. Other than the dummy connector, there are three types of links in STEN: links between i at time t_1 and j at time $t_2 > t_1$ representing a loaded move ($\mathcal{A}_f$); links between i at t_1 and j' at $t_2 > t_1$ representing an empty move ($\mathcal{A}_e$); and links between i and t_1 and i at $t_2 > t_1$ representing waiting ($\mathcal{A}_w$). Finally, the STEN is denoted as $\mathfrak{G}(\mathcal{N}, \mathcal{E}, T)$, where $\mathcal{N}$ and $\mathcal{E} = \mathcal{A}_w \bigcup \mathcal{A}_d \bigcup \mathcal{A}_e \bigcup \mathcal{A}_f \bigcup$ represent the set of all nodes and links, respectively.

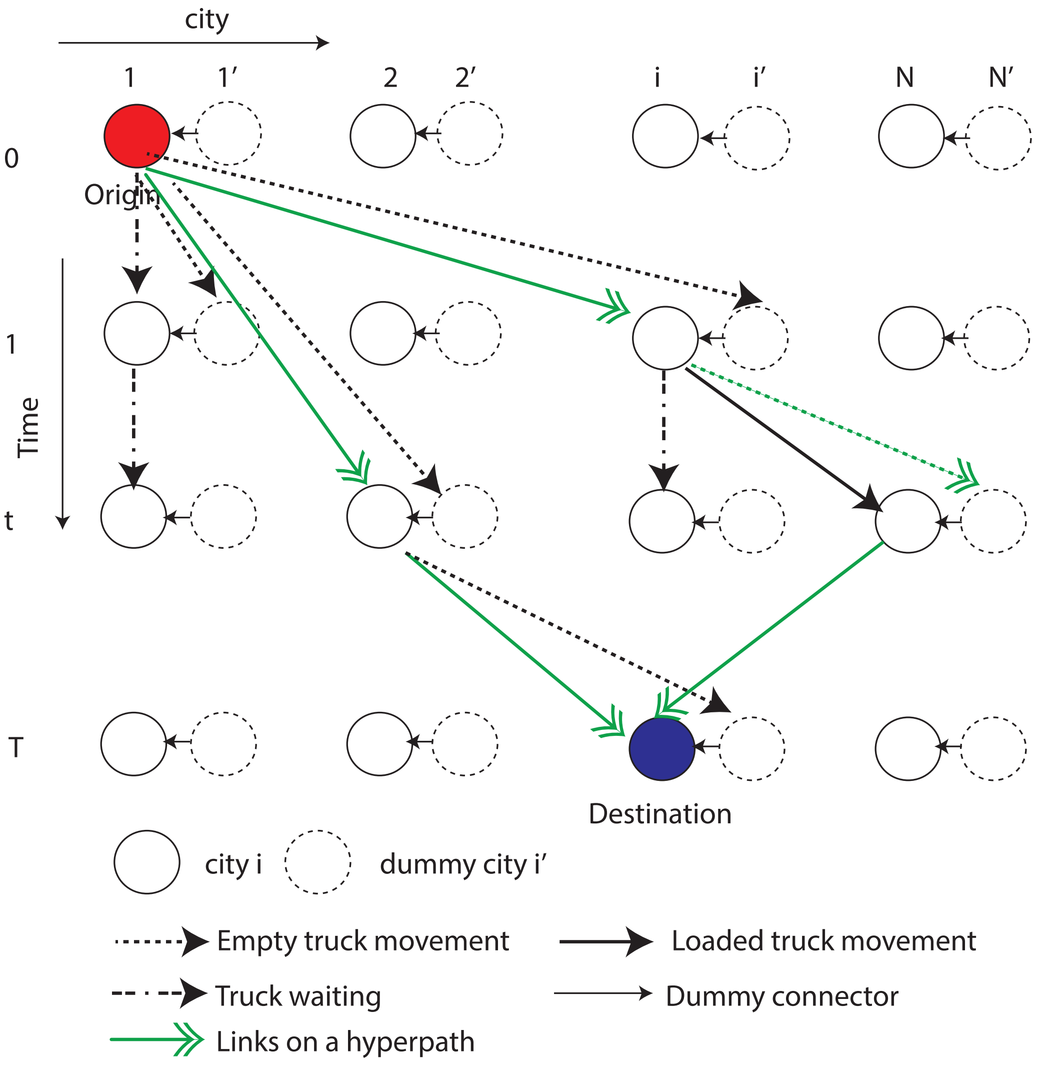
(7) Probabilities of winning a load solely depend on the bidding order and price. We solve Problem 3.1 with dynamic programming and representing the network with a space-time expanded network (STEN) as illustrated in Figure 3.1. Without loss of generality, we assume that the tour always starts at the current operating interval (i.e., $t = 0$), which reduces the number of input parameters to three. At each time interval t, a city has two copies in the network, denoted as i and i'. The former (i) represents the “real” city corresponding to loaded moves whereas the latter (i') is a dummy associated with empty moves.

The set of real and dummy cities in STEN is denoted as $\mathcal{L}$ and $\mathcal{L}'$ respectively. There is a dummy connector from i' to i at each time interval, relocating empty trucks to the real city so that they can be available for loads. The set of dummy connectors in STEN is denoted as $\mathcal{A}_d$. Other than the dummy connector, there are three types of links in STEN: links between i at time t_1 and j at time $t_2 > t_1$ representing a loaded move ($\mathcal{A}_f$); links between i at t_1 and j' at $t_2 > t_1$ representing an empty move ($\mathcal{A}_e$); and links between i and t_1 and i at $t_2 > t_1$ representing waiting ($\mathcal{A}_w$). Finally, the STEN is denoted as $\mathfrak{G}(\mathcal{N}, \mathcal{E}, T)$, where $\mathcal{N}$ and $\mathcal{E} = \mathcal{A}_w \bigcup \mathcal{A}_d \bigcup \mathcal{A}_e \bigcup \mathcal{A}_f \bigcup$ represent the set of all nodes and links, respectively.

3.1.3. Space-time Expanded Network Representation

We solve Problem 3.1 with dynamic programming and representing the network with a space-time expanded network (STEN) as illustrated in Figure 3.1. Without loss of generality, we assume that the tour always starts at the current operating interval (i.e., $t = 0$), which reduces the number of input parameters to three. At each time interval t, a city has two copies in the network, denoted as i and i'. The former (i) represents the "real" city corresponding to loaded moves whereas the latter (i') is a dummy associated with empty moves.

The set of real and dummy cities in STEN is denoted as $\mathcal{L}$ and $\mathcal{L}'$ respectively. There is a dummy connector from i' to i at each time interval, relocating empty trucks to the real city so that they can be available for loads. The set of dummy connectors in STEN is denoted as $\mathcal{A}_d$. Other than the dummy connector, there are three types of links in STEN:



Note 1: not all links are plotted here.

Figure 3.1. Illustration of the OFEX hyperpath problem

links between i at time t_1 and j at time $t_2 > t_1$ representing a loaded move ($\mathcal{A}_f$); links between i at t_1 and j' at $t_2 > t_1$ representing an empty move ($\mathcal{A}_e$); and links between i and t_1 and i at $t_2 > t_1$ representing waiting ($\mathcal{A}_w$). Finally, the STEN is denoted as $\mathfrak{G}(\mathcal{N}, \mathcal{E}, T)$, where $\mathcal{N}$ and $\mathcal{E} = \mathcal{A}_w \bigcup \mathcal{A}_d \bigcup \mathcal{A}_e \bigcup \mathcal{A}_f \bigcup$ represent the set of all nodes and links, respectively.

The optimal solution to the OFEX problem is represented by a hyperpath between the origin-destination (O-D) pair, as illustrated by links with green double-head arrows in Figure 3.1. The hyperpath contains a series of adaptive strategies at each decision point, starting from the origin. In the example in the figure, at $t = 0$ and $i = 1$, the optimal strategy contains a loaded move to either city 2 or city i. Which city the truck will visit with certainty depends on the result of the bidding process. We shall return to this problem later.

3.2. Optimal Bidding Model

Let us focus on a truck's decision problem at city i. To simplify the notation, we shall drop the subscript t from this section. At most a truck has $2n - 1$ possible options at any city. As seen in Figure 3.1, each option corresponds to a link in set $\mathcal{A} = \{i1, i1', \cdots, ii, \cdots, ij, ij', \cdots, in, in'\}$ (the head node must lie "below" the tail node on the STEN). For the notational convenience, a link in the STEN will mostly be denoted by a, with the tail node $a^- = i \in \mathcal{L}$ and the head node $a^+ = j \in \mathcal{L} \bigcup \mathcal{L}'$.

Each option a thus corresponds to an expected profit associated with the option $a \in \mathcal{A}$ (denoted as σ_a), the number of loads on each option (denoted as g_a), and the profit expected to receive at city a^+ (denoted as V_{a^+}). The number of trucks looking for load at i is H_i. These trucks are competing for the total number of $G_i = \sum_{a \in \mathcal{A}} g_a$ loads. Note that the profit $\sigma_a = x_a - c_a$, where c_a and x_a are the cost and revenue associated with choosing option a, given by

$$x_a \begin{cases} \in [l_a, u_a] & a \in \mathcal{A}_f \bigcap \mathcal{A} \\ = 0 & a \in \mathcal{A}_w \bigcap \mathcal{A} \\ = 0 & a \in \mathcal{A}_e \bigcap \mathcal{A} \end{cases} \quad c_a = \begin{cases} \alpha_f \tau_a + 2\gamma & a \in \mathcal{A}_f \bigcap \mathcal{A} \\ \alpha_w & a \in \mathcal{A}_w \bigcap \mathcal{A} \\ \alpha_e \tau_a & a \in \mathcal{A}_e \bigcap \mathcal{A} \end{cases}, \tag{3.4}$$

where α_f, α_e, α_e and γ are the cost for a loaded move, empty move, waiting, and handling, all measured in \$ per operating interval Δ. Note that the total handling cost is 2γ because per Assumption 6, the handling time is 2Δ.

Equation (3.4) states that for (and only for) a loaded move a, x_a is a variable (with a range of $[l_a, u_a]$) determined from an auction. In particular, each truck is assumed to participate in spot procurement auctions for available loads for one or more options. The truck must place a bid for each auction one at a time. This is necessary for our problem because a truck cannot serve more than one load at a time per Assumption 5. All auctions are single-round, sealed-bid, and first-price auction [see, e.g. 38].

Because the bid affects the value of a loaded move, the bidding order and price are interdependent variables and should be determined simultaneously. For narrative convenience, however, ordering and pricing decisions will be treated separately. Trucks only bid for a set of feasible loaded options, denoted as $\mathcal{Q}$, where $\|\mathcal{Q}\| = m$. In what follows, we shall first address the pricing issue for a given biding order $\phi \in \Phi$, where Φ is the set of all feasible bidding orders for all options in $\mathcal{Q}$. For simplicity, the options will be re-indexed for each order; hence the first in the list is always indexed as 1.

Let b_a be the number of trucks bidding for option a. It should be understood that b_a is a function of order ϕ because b_a is different for the same option under a separate order.

For now, b_a is assumed to be given, and we shall discuss how to determine b_a later. If $g_a \geq b_a$, we assume that each truck will get a load regardless of the bid price (within a reasonable bidding range set by the shipper). If not, some trucks will fail and will have to bid on the next desirable load.

When a truck bids for a load of option a, s/he has to choose a bid price of x_a, which will determine the winning probability p_a. Without loss of generality, we assume that $p_a = F_a(x_a)$ is a strictly decreasing function of x_a and that through the platform, the truck could estimate function F_a from the observed price of winning bids. We emphasize, without explicitly reflecting it in notations for simplicity, that $F_a(\cdot)$ are determined by b_a (how many trucks are competing for loads in option a) and g_a (number of loads available in a), with b_a depending on the bidding order ϕ.

For each option, the truck is assumed to choose the optimal bid price to maximize the expected profit, while accurately anticipating the results in later rounds of auctions. This pricing problem is naturally formulated as the following dynamic program:

$$\text{(3.5a)} \qquad z_a^* = \max_{x_a}\{(F_a(x_a)(V_{a^+} + \sigma_a) + z_{a+1}^*(1 - F_a(x_a))\}, a = 1, \cdots, m$$

$$\text{(3.5b)} \qquad z_a^* = V_{a^+} + \sigma_a, a = m + 1,$$

where z_a^* is the optimal expected profit after the ath round of auction. For any given order, the maximum expected profit is achieved in the first round, i.e., z_1^*.

Equation (3.5b) states that $a = m + 1$ is the fallback option in case all other bids fail to come through. The fallback option is set to the option that has the highest expected profit $(V_{a^+} + \sigma_a)$ and belong to either $\mathcal{A}_e$ (empty move) or $\mathcal{A}_w$ (waiting).

3.2.1. Winning Probability Function

For modeling convenience in this dissertation, we shall assume that the prevailing price of loads between a given city pair is uniformly distributed. As we shall show later, the empirical data we have collected suggest that this is a reasonable assumption. The price distribution reflects the random factors inherent in freight type, shipper characteristics, seasonal fluctuations and so on.

With the above assumption, the probability of winning a "standard load" at a price offer x_a is given by the winning probability function:

$$F_a(x_a) = \frac{u_a - x_a}{u_a - l_a}, \tag{3.6}$$

where l_a and u_a are lower and upper bounds of the price range.

Function (3.6) implies that an average price bid always has a winning probability of 0.5 (as predicted by the uniform distribution). This percentage most likely is not the case when accounting for the competition vying for the bid. For example, if a truck were the only bidder, and made a bid within the acceptable range of bids, that truck would win the bid guaranteed. However, if a trucker was competing with four other trucks, assuming all trucks made bids within the acceptable range randomly, that truck may have a much lower chance of winning with an average price bid. The following lemma is introduced to formalize the discussion.

Lemma 1. *Suppose (1) the option a has g_a loads available and receives b_a bids and (2) all trucks bid randomly within the price range $[l_a, u_a]$. For a truck bidding at the expected price, i.e. $x_a = (u_a + l_a)/2$, the probability of winning the bid, denoted as p_a^0, is 1 if*

$b_a \leq g_a$. When $b_a > g_a$,

$$p_a^0 = 0.5^{b_a} \sum_{k=1}^{g_a} C_{b_a-1}^{k-1}. \tag{3.7}$$

where

$$C_b^k \equiv \frac{b!}{(b-k)!(k)!}$$

Proof. Proof of Lemma 1. It is trivial to see that when $b_a \leq g_a$, everyone with a feasible bid $x_a \in [l_a, u_a]$ will get a load. When $b_a > g_a$, let's first consider $g_a = 1$. Since there is only one load, the truck must be the lowest bid in order to win it. Since everyone else bids randomly, the probability that $b_a - 1$ bids are larger than $(u_a + l_a)/2$ is 0.5^{b_a-1}. When $g_a = 2$, the probability of winning equals the sum of the probabilities of the average bid being the lowest and the second lowest. The probability of the bid being the second lowest can be computed as $0.5^{b_a-1}C_{b_a-1}^1$. To see why this is the case, note that $C_{b_a-1}^1$ gives the number of possible combinations that place exactly one truck's bid higher than $(u_a + l_a)/2$. Since each independent bid has 50% chance of being either higher or lower than the expected value, the total probability equals $0.5^{b_a-1}C_{n-1}^1$. With the same logic, we can show that when $g_a = k$, the probability of being the kth lowest bid should be $0.5^{b_a-1}C_{b_a-1}^{k-1}$. The probability of winning a load is the sum of probabilities of being the lowest, the second lowest until the kth lowest bid. This completes the proof. □

p_a^0 gives the winning probability of an average price bid when the peer competition is considered. To estimate the winning probability for any feasible bid price in the range $[l_a, u_a]$, we propose to use the following rational function that takes p_a^0 as a parameter

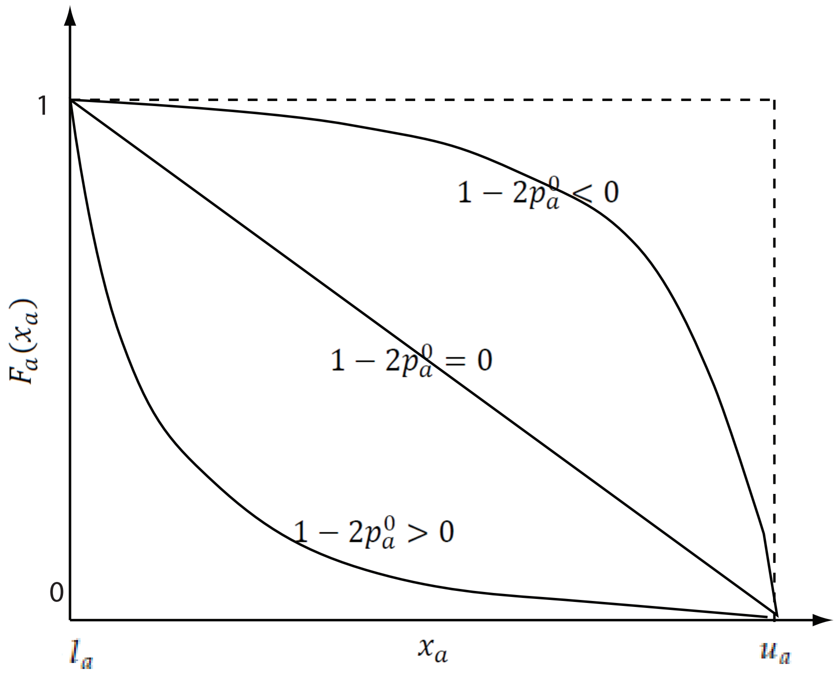


Figure 3.2. Illustration of the proposed winning probability function

$$F_a(x_a) = \frac{p_a^0(u_a - x_a)}{(1-2p_a^0)(x_a - l_a) + p_a^0(u_a - l_a)}. \tag{3.8}$$

We leave it to the reader to verify that $F_a(x_a) = p_a^0$ when $x_a = (u_a + l_a)/2$. Figure 3.2 illustrates how the value of p_a^0 affects the shape of the function. When $p_a^0 < 0.5$ (i.e., an average priced bid leads to a probability of winning lower than 50%), the function is convex, suggesting that one has to bid at a rather low price to achieve a relatively high winning probability. When $1 > p_a^0 > 0.5$, the function is concave, suggesting that a high bid would still likely to be accepted with a high probability due to a low level of competition. When $p_a^0 = 0.5$, the function is reduced to a linear function equal to the cumulative distribution function of a uniform distribution.

Proposition 1. *$F_a(x_a)$ given by Equation (3.8) is a well-defined cumulative distribution function for the option a if $b_a > g_a > 0$.*

Proof. Proof of Proposition 1. We first show that when $0 < g_a < b_a$, p_a^0 should always range between $(0, 1)$. It is easy to see that when $g_a = 0$ $p_a^0 = 0$ and when $g_a = b_a$, $p_a^0 = 1$. If $0 < g_a < b_a$, p_a^0 is always positive. Because $\sum_{k=1}^{g_a} C_{b_a-1}^{k-1} < 2^{b_a} - 1$, $p_a^0 = 0.5^{b_a}(2^{b_a} - 1) = 1 - 0.5^{b_a} < 1$. It can be easily verified that $F_a(l_a) = 1$ and $F_a(u_a) = 0$ for any $p_a^0 \in (0, 1)$. Further, the derivative of $F_a(x_a)$ is

$$\frac{dF_a(x_a)}{dx_a} = \frac{p_a^0(l_a - u_a + p_a^0(u_a - l_a))}{((1 - 2p_a^0)(x_a - l_a) + p_a^0(u_a - l_a))^2}$$

This derivative is strictly negative when $p_a^0 \in (0, 1)$, which implies that $F_a(x_a)$ is a strictly decreasing function between $[l_a, u_a]$. This completes the proof. □

Equation 3.7 is valid only when b_a and g_a are integers. In practice, the number of available trucks H_a, hence b_a, may be fractional because it may be obtained from other computational procedures (e.g., take an average over several observations). The same is true for the number of available loads, g_a. One can round a fractional number to the nearest integer, but doing so may create nontrivial rounding errors, especially when the numbers are small. To address this issue, we propose to approximate the binomial function based on the normal distribution. Specifically, a binomial distribution with parameters g_a, b_a, and p may be approximated by a normal distribution with $x = g_a$, the mean $\mu = b_a p$ and the standard deviation $\sigma = \sqrt{b_a p(1 - p)}$. See [21] for the method and recent developments in the approximation. Furthermore, we apply Yates' continuity correction, which is most accurate for when p is near 0.5, as is our case [86]. The resulting approximate formula reads:

$$p_a^0 = 0.5^{b_a} \sum_{k=1}^{g_a} C_{b_a-1}^{k-1} \approx \frac{1}{\sqrt{0.5(b_a-1)\pi}} \int_{-\infty}^{g_a-\frac{1}{2}} e^{\frac{-(t-0.5(b_a-1))^2}{0.5(b_a-1)}} dt. \tag{3.9}$$

3.2.2. Pricing Problem

We proceed to discuss the truck's sequential bidding (or pricing) problem (3.5). It is easy to see that the problem has to be solved backward, starting from the fallback option, for which the expected profit is fixed. Then for each option $a < m+1$, we solve the following nonlinear program for x_a

$$\max z_a = F_a(x_a)(x_a - c_a + V_{a^+}) + (1 - F_a(x_a))z_{a+1} \tag{3.10a}$$

subject to

$$l_a \leq x_a \leq u_a \tag{3.10b}$$

$$a \leq m \tag{3.10c}$$

where c_a is defined in (3.4) and z_{a+1} is obtained by solving the same problem for option $a+1$. When $a = m$, z_{a+1} is simply the expected value of the fallback option. Clearly, solving (3.10) will produce an optimal bid price x_a^* for each a, and correspondingly, the expected profit for each option is $\sigma_a^* = x_a^* - c_a$.

Problem (3.10) can be solved in closed form because the objective function is a well-defined quadratic function. We formally state its solution as follows.

Proposition 2. *The optimal solution to Problem (3.10) is given by*

(3.11)

$$x_a^* = \begin{cases} \frac{\zeta - \sqrt{\theta}}{4p_a^0 - 2} & \text{if } p_a^0 \neq 0.5,\ \theta \geq 0\ \&\ \frac{\zeta-\sqrt{\theta}}{4p_a^0-2} \in [l_a, u_a] \\ \frac{u_a + c_a + z_{a+1} - V_{a^+}}{2} & \text{if } p_a^0 = 0.5\ \&\ \frac{u_a+c_a+z_{a+1}-V_{a^+}}{2} \in [l_a, u_a] \\ l_a & \text{if } \frac{\zeta-\sqrt{\theta}}{4p_a^0-2} < l_a,\ p_a^0 \neq 0.5\ \&\ \theta \geq 0 \text{ or } \frac{u_a+c_a+z_{a+1}-V_{a^+}}{2} < l_a \text{ when} \\ & \quad p_a^0 = 0.5 \text{ or } z_{a+1} > x_a - c_a + V_{a^+} \text{ when } \theta < 0\ \&\ p_a^0 \neq 0.5 \\ u_a & \text{if } \frac{\zeta-\sqrt{\theta}}{4p_a^0-2} > u_a,\ p_a^0 \neq 0.5\ \&\ \theta \geq 0 \text{ or } \frac{u_a+c_a+z_{a+1}-V_{a^+}}{2} > u_a \text{ when} \\ & \quad p_a^0 = 0.5 \text{ or } z_{a+1} \leq x_a - c_a + V_{a^+} \text{ when } \theta < 0\ \&\ p_a^0 \neq 0.5, \end{cases}$$

where $\zeta = 2u_a p_a^0 + 2l_a p_a^0 - 2l_a$ *and*

$$\theta = \begin{cases} (2l_a - 2u_a p_a^0 - 2l_a p_a^0)^2 - (8p_a^0 - 4)(c_a u_a - V_{a^+} u_a - l_a c_a + l_a V_{a^+} - l_a u_a - c_a u_a p_a^0 + l_a c_a p_a^0 \\ \quad + V_{a^+} u_a p_a^0 - l_a V_{a^+} p_a^0 + l_a u_a p_a^0 + {u_a}^2 p_a^0 + l_a p_a^0 z_{a+1} - u_a p_a^0 z_{a+1} - l_a z_{a+1} + u_a z_{a+1}) \end{cases}$$

Proof. Proof of Proposition 2. Because Equation 3.10a is a continuous second order polynomial within $[l, u]$, we can solve the maximization problem by simply setting the first derivative equal to 0 and solving for x_a.

Set $\frac{dz(x_a^*)}{dx_a} = 0$

$$\frac{dz(x_a^*)}{dx_a} = \frac{p_a^0(l_a - u_a + p_a^0(u_a - l_a))(x_a - c_a + V_{a^+})}{((1-2p_a^0)(x_a - l_a) + p_a^0(u_a - l_a))^2} + \frac{p_a^0(u_a - x_a)}{(1-2p_a^0)(x_a - l_a) + p_a^0(u_a - l_a)} - \frac{p_a^0(l_a - u_a + p_a^0(u_a - l_a))z_{a+1}}{((1-2p_a^0)(x_a - l_a) + p_a^0(u_a - l_a))^2} = 0$$

Solving this equation we get

$$x_a = \frac{\zeta \pm \sqrt{\theta}}{4p_a^0 - 2},$$

where

$$\zeta = 2u_a p_a^0 + 2l_a p_a^0 - 2l_a$$

and

$$\theta = \begin{cases} (2l_a - 2u_a p_a^0 - 2l_a p_a^0)^2 - (8p_a^0 - 4)(c_a u_a - V_{a^+} u_a - l_a c_a + l_a V_{a^+} - l_a u_a - c_a u_a p_a^0 + l_a c_a p_a^0 \\ \qquad + V_{a^+} u_a p_a^0 - l_a V_{a^+} p_a^0 + l_a u_a p_a^0 + {u_a}^2 p_a^0 + l_a p_a^0 z_{a+1} - u_a p_a^0 z_{a+1} - l_a z_{a+1} + u_a z_{a+1}) \end{cases}$$

Note: only the minus critical point in $\frac{\zeta \pm \sqrt{\theta}}{4p_a^0 - 2}$ is a maximum point. The positive critical point is the minimum point of a convex section of the objective function after the asymptotic discontinuity. The only exception being when $\theta = 0$, then only one critical point exists. To verify, the sign of $\frac{d^2 z(x_a^*)}{d{x_a}^2}$ of the negative critical point, when real, is strictly negative, thus x_a^* is a maximum point of the concave function when it is a real number, and $p_a^0 \neq \frac{1}{2}$. The sign of $\frac{d^2 z(x_a^*)}{d{x_a}^2}$ of the positive critical point, when real, is strictly positive, which signifies a minimum point.

When $p_a^0 = \frac{1}{2}$: $F_a(x_a)$ is a linear function and setting $\frac{dz(x_a^*)}{dx_a} = 0$ yields $x_a = \frac{u_a + c_a + z_{a+1} - V_{a^+}}{2}$. These two cases yield the optimal bid points, but limited by the lower and upper bid prices $[l_a, u_a]$. With these four points, the optimal bid price can be summarized as given in the Proposition 2. □

3.2.3. Bidding Strategies

Recall that a critical input to the sequential bidding problem is b_a, i.e., the number of trucks bidding for each loaded move. It is worth emphasizing that from the perspective of the truck that is actively solving the OFEX routing problem, all H_i trucks available at city i are competitors. Predicting these trucks' bidding behaviors (hence determining b_a) is a challenging exercise for many reasons. First, not all trucks attempt to maximize the expected profit as defined by Problem (3.5). Second, even if they do, each truck may have a unique O-D or desired shipping duration, which may lead to very different future expected profit, V_{a^+}. Third, many attributes of the trucks that could affect their bidding priority are simply not captured in the model. In reality, the OFEX platform could estimate b_a by mining microscopic truck behavioral data, such as call records and mobile app usage. Since such data are not readily available in this dissertation, a simplifying method is given below to generate an "educated guess" to test the OFEX routing model. To this end, it is assumed (1) that the number of competing trucks for each load can be determined by logit model that gives preference for loads with high profitability more load availability; and (2) that the competing trucks do not know their competitors for each option. As a result of (1), we further assume that the utility of an option depends on its present expected profit, $\bar{\sigma}_a$, and the number of loads, g_a. To estimate $\bar{\sigma}_a$, the following maximization problem is solved for each option $a \in \mathcal{Q}$

$$\max F_a(x_a, p_a^0)(x_a - c_a), \text{subject to } x_a \in [l_a, u_a], \tag{3.12}$$

and $\bar{\sigma}_a$ is set to $x_a^* - c_a$, where x_a^* is the optimal solution to (3.12). In the above problem, $F_a(x_a, p_a^0)$ is the winning probability at a bid price x_a, given the chance of winning at an average price bid p_a^0. Note that, because of the second assumption made above, these trucks cannot use Lemma 1 (which assumes the knowledge of b_a) to determine p_a^0. Instead, p_a^0 is replaced by a parameter (denoted as $\bar{p}_0$) that reflects the average winning probability of an average price bid across the platform (see Section 3.4.3 for how $\bar{p}_0$ is determined). Once $\bar{\sigma}_a$ and g_a are given, a multinomial logit model [18] may be used to estimate the distributions of H_i among all feasible options; see Appendix B for details.

3.2.4. Combined Ordering and Pricing Optimization Problem

We are now ready to present the combined optimization problem with both ordering and pricing as follows:

$$\max_{\phi \in \Phi} z_1(\phi) \tag{3.13a}$$

where $z_1(\phi)$ is the solution to the dynamic program

$$\max_{x_a \in [l_a, u_a]} z_a(\phi) = F_a(x_a, \phi)(x_a - c_a + V_{a^+}) + \tag{3.13b}$$

$$(1 - F_a(x_a, \phi) z_{a+1}(\phi)), a = 1, 2, \cdots, m;$$

$$z_a(\phi) = V_{(a)^+} + \sigma_a, a = m + 1. \tag{3.13c}$$

In the above formulation, each ϕ represents an order of the m options corresponding to the feasible loaded moves in $\mathcal{Q}$. $F_a(x_a, \phi)$ depends on ϕ through b_a^ϕ defined in Equations (B.5) and (3.8). The fallback option $m + 1$ is always the waiting/empty move that has the highest expected future profit and it does not vary with the order. Once Problem (3.13) is solved for each $a \leq m$, the optimal bidding price (x_a^*) and the optimal expected profit (z_a^*) for each round of auction are obtained. In addition, the probability of choosing option a, denoted as π_a, can be computed as follows:

$$\pi_1 = F_1(x_1, \phi), \tag{3.14a}$$

$$\pi_a = F_a(x_a, \phi) \Pi_{a'=1}^{a-1} (1 - F_{a'}(x_{a'}, \phi)), a = 2, \cdots, m + 1. \tag{3.14b}$$

Problem (3.13) can be solved using the Bidding and Pricing Algorithm presented in Figure 3.3. Located among all the inputs specified on line 1, $\bar{p}_0$ is the probability of

Algorithm 3.1 Combined Ordering and Pricing Algorithm (COPA) for city i at time t

```
1: Input: $\mathcal{Q}, H_i, g_a, l_a, u_a, V_{a+}, \tau_a \forall a \in \mathcal{Q}$, the fallback option $\bar{f}$ and its expected profit
   $z_{\bar{f}}$, $\bar{p}_0$.
2: Output: $x^*_a, \pi^*_a, \forall a \in \mathcal{A}, z^*$.
3: Initialize:
4: Set $\pi^*_a = 0, z^* = -1, x_a = 0.5(l_a + u_a), \forall a \in \mathcal{A}$.
5: Set the estimated profit $\bar{\sigma}_a$ for each option $a \in \mathcal{A}_f$ by solving Problem (3.12) while
   setting $p^0_a = \bar{p}_0$.
6: Set $\bar{b}_a$ (the number of competing trucks for each option) using the method described
   in Appendix B.
7: Set the fallback option $m + 1 = \bar{f}$.
8: Main loop:
9: for every order $\phi \in \Phi$ do
10:     Set $\pi^\phi_{m+1} = 1, z^\phi_{m+1} = z^0_f$.
11:     for $a = m$ to 1 do
12:         Solve Problem (3.10) by finding $x^\phi_a$ using Equation (3.11).
13:         Evaluate $F_a(x^\phi_a)$ using Equations (3.8) and (3.9) and $z^\phi_a$ using Equation (3.10a).
14:         Set $\pi^\phi_a = F_a(x^\phi_a)$, update $\pi^\phi_{a+1} = \pi^\phi_{a+1}(1 - F_a(x^\phi_a))$.
15:     end for
16:     if $z^\phi_a > z^*$ then
17:         Set $z^* = z^\phi_a, x^*_a = x^\phi_a, \pi^*_a = \pi^\phi_a$.
18:     end if
19: end for
20: return $x^*_a, \pi^*_a, \forall a \in \mathcal{A}, z^*$.
```

Figure 3.3. Combined Ordering and Pricing Algorithm (COPA)

winning a bid at the average price, averaged over all bids. Line 3 - 7 initializes the solution variables and the number of competing trucks. The main iteration of the algorithm (line 9 - 19) enumerates all possible bidding orders and solves the pricing problem for each order. The order that yields the most substantial expected profit is retained as the incumbent for the optimal solution (line 16 - 18).

Unfortunately, COPA has extremely high time complexity because all orders must be enumerated. The number of possible orders is $n!$, which is an astronomically large number even for modest n (2×10^{32} for $n = 30$.) A complete enumeration of orders may be unnecessary, if, for example, a predetermined order based on a certain metric can ensure optimality (as in a greedy algorithm). Alternatively, one may design an iterative procedure that can quickly converge to the optimal order by searching only a small subset of the feasible space. It remains an open question whether or not either approach is feasible, however. Instead, a simple greedy heuristic algorithm is proposed in this dissertation to compare orders in a small predefined set. We define these special orders in what follows:

(1) The truck follows the order defined by ranking $\bar{\sigma}_a/(\tau_a + 2)$ denoted as $\bar{\phi}_0$. This order gives priority to the options with higher estimated profitability. τ_a and 2 are the travel and handling time measured in the unit of the operating interval (the handling time equals twice the operation interval as per Assumption 6), respectively. $\sigma_a/(\tau_a + 2)$, rather than σ_a, is a better metric for profitability because the former takes into account the total time required to serve the load. Clearly, for the same profit σ_a, a shorter trip is more desirable than a longer one.

(2) The truck follows the order defined by ranking $V_{a^+}, \forall a \in \mathcal{A}_f$, denoted as $\bar{\phi}_1$. With this order, the truck gives priority to the options with greater future profit potential, while ignoring the present profits.

(3) The truck follows the order defined by ranking $V_{a^+} + 0.5(l_a + u_a) - c_a, \forall a \in \mathcal{A}_f$, denoted as $\bar{\phi}_2$. This order implies that the truck gives priority to options with higher total expected profit, but ignores the competition in the current bid.

(4) The truck follows the order defined by ranking $V_{a^+} + \bar{\sigma_a}, \forall a \in \mathcal{A}_f$, where $\bar{\sigma_a}$ is obtained by solving Problem (3.12) while setting $p_a^0 = \bar{p}_0$. Denoted as $\bar{\phi}_3$, this

order allows the truck to take into consideration both the future and estimated present profits when prioritizing options.

For each $\bar{\phi}_i$, we can also define a $\bar{\phi}'_i$ that represents its reverse order. Let $\bar{\Phi} = \{\bar{\phi}_i, \bar{\phi}'_i, i = 0, 1, 2, 3\}$. A heuristic version of COPA is obtained by simply replacing Φ on line 9 with $\bar{\Phi}$.

In COPA, each pricing problem requires ordering $\mathcal{Q}$ once and solving Problem (3.10) up to $\|\mathcal{Q}\| = m$ times. In the worst case, $m = n$, so the total effort amount to $O(n \log n + n)$. Since the pricing problem must be solved for up to $\|\Phi\| = n!$ times in the exact algorithm, the complexity is roughly $O((n+1)!(\log +1))$ for large n. For the heuristic algorithm, the complexity is proportional to the size of $\bar{\Phi}$, i.e., $O(\|\bar{\Phi}\|(n \log n + n))$.

3.3. Hyperpath problem

Having examined the pricing and bidding problem for city i at time t, we now turn to the OFEX routing problem, i.e., finding the optimal shipping tour - represented as a hyperpath - in the STEN shown in Figure 3.1 for a given O-D pair rs. Since each expanded network is constructed in response to a specific request, the tour always starts from city r at $t = 0$ and ends at city s at $t = T$. Following [52], we first define a hyperpath between r at $t = 0$ and s at $t = T$ as follows.

Definition 1 (Hyperpath in the space-time expanded network). *A subgraph of $\mathfrak{G}(\mathcal{N}, \mathcal{E}, T)$, denoted as $\mathfrak{G}_h(\mathcal{N}^h, \mathcal{E}^h, \pi^h)$, where $\mathcal{N}^h \subset \mathcal{N}, \mathcal{E}^h \subset \mathcal{E}$, and $\pi^h = (\pi^h_a, a \in \mathcal{E}^h)$ a real value vector, is a hyperpath connecting r at time 0 and s at time T if*

(1) *$\mathfrak{G}_h$ is acyclic with at least one arc;*

(2) *the origin r has no predecessors and the destination s no successors in $\mathfrak{G}_h$;*

(3) *for every node $i \in \mathcal{N}^h$, there is a path from r to s traversing i; and*

(4) *the characteristic vecotr π^h satisfies*

$$\sum_{a \in \mathcal{E}^h, a^- = i} \pi_a^h = 1; \quad \pi_a^h \geq 0, \forall a \in \mathcal{E}^h.$$

The set of all hyperpaths connecting r at time 0 and s at time T is denoted as $\mathcal{H}_{rs}^T$. π_a^h is the probability that a truck ends up on arc a starting from node a^-, which equals the winning probability defined in (3.14). Define σ_a^h as the profit earned on arc $a \in \mathcal{E}^h$, which equals $x_a^h - c_a$ (where x_a^h is the optimal bidding price, obtained from COPA). Let $\mathcal{K}_h$ denote the set of all simple paths of $\mathfrak{G}_h$ connecting r and s. For any $k \in \mathcal{K}_h$, the probability of traversing k is

$$\lambda_k^h = \prod_{a \in \mathcal{E}^h} \pi_a^h \delta_{ak}^h, \forall k \in \mathcal{K}_h, \tag{3.15}$$

where $\delta_{ak}^h = 1$ if a is on path $k \in \mathcal{K}_h$ and 0 otherwise. With the profit associated with path k being defined as

$$\omega_k^h = \sum_{a \in \mathcal{E}^h} \sigma_a^h \delta_{ak}^h, \forall k \in \mathcal{K}_h, \tag{3.16}$$

the total profit of a hyperpath h is given by

$$\acute{W}_h = \sum_{k \in \mathcal{K}_h} \omega_k^h \lambda_k^h. \tag{3.17}$$

Thus, the objective of the hyperpath problem is to find

$$h^* = \text{argmax}\{\acute{W}_h, h \in \mathcal{H}_{rs}^T\}. \tag{3.18}$$

This problem can be solved as a dynamic program following decreasing order of time (DOT), as described in Figure 3.4. Let V_i^t denote the maximum expected profit earned starting from city i at t and returning to city s at T. The Hyperpath DOT (HyDOT) algorithm starts by initializing all V_i^T and $V_{i'}^T$ as $-\infty$, except for the destination s at $t = T$, where it is set to 0 (line 4). Any node with a negative infinity V_i^t is called *unreachable*, in the sense that a truck starting from that node cannot find a path to arrive at the destination s by T.

In the main loop, the algorithm sequentially visits each layer in a descending order (starting from $t = T$). For every city it scans on a layer, HyDOT first creates a list of loaded moves ($\mathcal{Q}$) that leads to *reachable* nodes and identifies a *reachable* fallback option $\bar{f}$ (line 8 - 20). If $\mathcal{Q}$ is not empty, HyDOT calls COPA to maximize the expected profit earned at the node (line 22 - 23), and then adds the links and nodes corresponding to options with positive choice probabilities into the hyperpath (line 24 - 34). Note that whenever a dummy node is added into the hyperpath, its dummy connector should also be added to ensure connectivity (line 29 - 31). Also, whenever a "real" node becomes reachable, so does the corresponding dummy node (line 36 - 38). After all layers are scanned, we add node r into the hyperpath to complete the construction of the graph.

Algorithm 3.2 Hyperpath DOT Algorithm (HyDOT)

```
Inputs: $\mathfrak{G}(\mathcal{N}, \mathcal{E}, T)$, $H_i^t, g_a^t, l_a^t, u_a^t, \forall a \in \mathcal{A}_f, V_{a^+}, \tau_a^t, \forall a \in \mathcal{E}$, $\bar{p}_0$.
Outputs: $\mathfrak{G}_{h^*}, \pi_a^{h^*}, x_a^{h^*}, \forall a \in G_{h^*}$. $V_i^t, \forall i \in \mathcal{N}^{h^*}, t = 0, \cdots, T$.
Initialize: set $V_i^t = V_{i'}^t = -\infty, i = 1, \cdots, n, t = 0, \cdot, T$, $V_s^T = 0$, $N^{h^*} = \emptyset, E^{h^*} = \emptyset$.
for $t = T$ to 0 do
    for all $i = 1, \cdots, n$ do
        for all $a$ in $\mathcal{A}$ starting from city $i$ at time $t$ do
            Initialize list of feasible options $\mathcal{Q} = \emptyset$, fallback option $\bar{f} = -1, z_{\bar{f}} = -\infty$.
            Set $e = t + \tau_a^t + 2$
            if $e \leq T$ and $V_{a^+}^e > -\infty$ then
                if $a \in \mathcal{A}_f$ then Set $\mathcal{Q} = \mathcal{Q} \bigcup a$
                else
                    if $V_{a^+} - c_a > z_{\bar{f}}$ then Set $\bar{f} = a, z_{\bar{f}} = V_{a^+} - c_a$.
                    end if
                end if
            end if
        end for
        if $\mathcal{Q} \neq \emptyset$ then
            Run COPA to obtain $x_a^*, \pi_a^*, z^*$ for $\mathcal{Q}$ (loaded moves) & $\bar{f}$ (fallback option)
            Set $V_i^t = z^*$
            for all $a \in \mathcal{Q}$ do
                if $\pi_a^* > 0$ then
                    Set $\mathcal{E}^{h^*} = \mathcal{E}^{h^*} \bigcup a$, set $x_a^{h^*} = x_a^*, \pi_a^{h^*} = \pi_a^*$.
                    if $a^+ \notin \mathcal{N}^{h^*}$ then
                        Set $\mathcal{N}^{h^*} = \mathcal{N}^{h^*} \bigcup a^+$.
                        if $a^+ \in \mathcal{L}'$ then
                            Add the dummy connector starting from $a^+$ into $\mathcal{E}^{h^*}$
                        end if
                    end if
                end if
            end for
        end if
        if $V_i^t < \infty$ then
            Set $V_{i'}^t = V_i^t$.
        end if
    end for
end for
Add node $r$ at $t = 0$ into $\mathcal{N}^{h^*}$.
Return $\mathfrak{G}_{h^*}$.
```

Figure 3.4. DOT Hyperpath Algorithm.

Proposition 3. *The HyDOT algorithm presented in Figure 3.4 finds the optimal hyperpath h^*.*

Proof. Proof We first show that the algorithm always constructs a feasible hyperpath. Requirement 1 in Definition 1 is satisfied as long as $T \geq 1$ since the STEN itself is acyclic. Requirement 2 is obvious by construction: for s at T, no outgoing options are feasible; and for r at $t = 0$, it is added at the end, so none of its incoming links would ever be examined. Requirement 3 is always met because only reachable nodes can be added into the hyperpath and by definition, a reachable node can always find a path to arrive at s by T. Requirement 4 is guaranteed by the definition of choice probability (3.14). A hyperpath h so constructed must have the maximum $\acute{W}_h$ because whenever V_i^t is maximized, all $V_i^{t'}, t' > t$ would already have been maximized (guaranteed by the DOT sequence). In other words, all sub-hyperpaths of the optimal hyperpath are always optimal. This ensures optimality based on backward induction. □

Finally, we note that the computational effort of the HyDOT algorithm is dominated by the number of COPA calls, which in the worst case equals nT.

3.4. Implementation Issues

In this section, we discuss several issues arising from implementing the proposed model using empirical data. We have access to real freight movement data collected by a Chinese OFEX platform. The datasets used in this dissertation cover 31 major cities in China within two two-week periods in December of 2015 and July 2016. The data includes, among other things, the following information for each load: at which time it was posted on the platform, how long it stayed online, its origin and destination, the freight type, the preferred truck type, the distance, the weight, and the price. In total, the datasets record about 113,000 loads.

Recall that one of the main inputs for the OFEX routing problem is the number of trucks available at each city H_i^t. This information, however, is not directly available in the data sets. In Section 3.4.1, we propose a method to estimate H_i^t from the freight movement. Section 3.4.2 examines the empirical distribution of load prices and Section 3.4.3 discusses how to estimate the overall average probability of winning loads with an average price bid.

3.4.1. Trucks Available at a City

For each city i, we know both the posted outgoing loads scheduled to depart from i at t, as well as the incoming loads scheduled to complete delivery at i at t. Assuming that all trucks arriving at city i with loads become available afterward, a starting point to estimate the number of available trucks at t is to use the total number of incoming trucks less that of outgoing trucks at $t-1$. Formally,

$$H_i^t = \sum_{t'=0}^{t-1} \sum_{j=1}^{n} \sum_{a, a^- = j, a^+ = i, t' + \tau_a^{t'} = t-1} g_a^{t'} - \sum_{a^- = i} g_a^{t-1}. \tag{3.19}$$

Equation (3.19) ignores the fact that some trucks may move between cities without a load, which would not be counted in g_a^t. A more realistic estimation can be achieved if the empty truck flows can be estimated and included in the calculation. There is an extensive volume of literature devoted to the empty truck flow estimation problem, beginning with [81] which led to a dynamic transshipment representation by [80]. [48] is credited with the first formulation of the problem as a static linear programming problem, which is the approach utilized in this dissertation. See [15] for a summary of early studies on the subject. In this dissertation, we design a simple linear program to minimize the total distance of empty trips, while satisfying the long-term truck balance constraint at each city. The average empty truck flows are then scaled to match empirical data and distributed to each node in the STEN.

A simple observation drives the proposed model: a city with a positive balance (a sink node) of trucks needs to generate at least that number of empty trucks leaving the city; and a city with a negative balance (a source node) needs to attract enough empty trucks to service the surplus of loads available. While the model can be defined as dependent on time, our data suggested that the distribution of loads does not vary continuously with time. Instead, it shows two distinctive patterns: one for weekdays and the other for weekends. In light of this, the data are averaged by day for weekdays and weekends separately, and two models are created accordingly using the data to estimate empty truck flows in the two cases. The linear program reads:

$$\min \sum_{i=1,\cdots,n} \sum_{j=1,\cdots,n,j\neq i} \tau_{ij} y_{ij} \tag{3.20a}$$

subject to:

$$\sum_{j=1,\cdots,n,j\neq i} \bar{g}_{ij} + y_{ij} = \sum_{j=1,\cdots,n,j\neq i} \bar{g}_{ji} + y_{ji} \ \forall i = 1,\cdots,n \tag{3.20b}$$

$$y_{ij} \geq 0 \quad \forall i = 1,\cdots,n, j = 1,\cdots,n, j \neq i, \tag{3.20c}$$

where $\bar{g}_{ij}$ is the average daily number of loads between i and j in the data, and y_{ij} is the average daily empty truck flows between i and j. The objective function (3.20a) is the total travel time spent on moving empty. Constraints (3.20b) impose the truck flow conservation condition at each node. Constraints (3.20c) ensure nonnegativity. With y_{ij} being solved from (3.20), we propose to estimate the additional available trucks due to empty movement at city i at t as

(3.21)
$$e_i^t = \rho_0 \left(\sum_{j=1,\cdots,n,j\neq i} y_{ji} + \rho_1 \sum_{j=1,\cdots,n,j\neq i} \bar{g}_{ji} \right) \equiv \rho_0(Y_i + \rho_1 \bar{G}_i), \forall i = 1,\cdots,n, t = 1,\cdots,T$$

In the above, the parameter ρ_0 distributes the daily average to each operating interval, and ρ_1 is used to scale up the empty flows (proportional to the total loaded flow) to match empirical data. ρ_1 is introduced because the total distance traveled by empty trucks may not be minimized in reality as dictated by the model. Publicly available information often reveals the percentage of miles traveled by empty trucks in all truck miles, denoted as ϵ, which can be computed as:

$$\epsilon = \frac{\sum_i (Y_i + \rho_1 \bar{G}_i)}{\sum_i (Y_i + \rho_1 \bar{G}_i) + \sum_i \bar{G}_i}. \tag{3.22}$$

Suppose ϵ is given, ρ_1 can be obtained as

$$\rho_1 = \frac{\epsilon}{1-\epsilon} - \frac{\sum_i Y_i}{\sum_i \bar{G}_i}. \tag{3.23}$$

In this dissertation, we set $\epsilon = 0.35$ for our data set, as [50] estimate the empty truck miles traveled at about 30% to 40% of total truck miles traveled in China. This leads to $\rho_1 = 0.245$. ρ_0 is set to 0.5, implying about 50% of the empty trucks estimated to arrive within a workday would bid for any operating interval within the day.

Finally, the trucks available at node i at time t is given by

$$H_i^t = \sum_{t'=0}^{t-1} \sum_{j=1}^{n} \sum_{a, a^-=j, a^+=i, t'+\tau_a^{t'}=t-1} g_a^{t'} - \sum_{a^-=i} g_a^{t-1} + e_t^i. \tag{3.24}$$

3.4.2. Empirical Distribution of Prevailing Price

This section examines empirical distributions of prevailing prices. The data used here are the total price of each load for each city pair, which is listed when a shipper posts the load on the OFEX platform. We note that this price is not the final price, which is not available in this dataset. However, the price initially offered by the shipper seems a good surrogate for the prevailing price on the market.

Figure 3.5 reports the data and the fitting results for the provincial capital city pair Hubei-Henan, which has the highest number of price observations. The resulting best fit by likelihood was the beta distribution that closely resembles the shape of a uniform distribution.

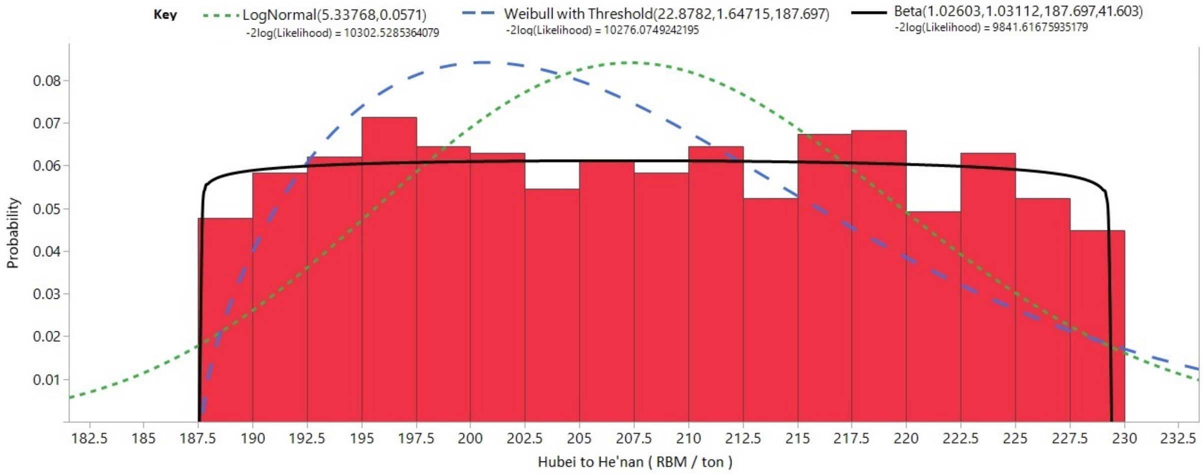


Figure 3.5. Empirical and fitted distributions of prevailing prices for the provincial capital city pair Hubei-Henan

This pattern of fitting is consistent over all provincial capital city pairs tested with at least 50 price observations. In particular, in all tested cases, the uniform distribution fits the data better than other distributions except the beta distribution. Therefore, the assumption that the prevailing price is uniformly distributed is properly justified.

3.4.3. Choice of $\bar{p}^0$

Section 3.2.3 introduces an empirical parameter, $\bar{p}^0$, to represent the overall average probability of winning at an average price bid. This parameter is used to help estimate how the competing trucks evaluate different options, i.e., determining $\bar{\sigma}_a$. We now discuss how $\bar{p}^0$ is determined empirically from the data.

Our data show that on average, 4.20 loads exist at any given location and time. Using the empty truck ratio $\epsilon = 0.35$ yields about 5.67 trucks competing for those loads on average. Applying the approximation in Equation (3.9) produces an average winning probability of 0.897. In other words, when an average-price bid is made with the average load/truck ratio, the likelihood of winning is 89.7%. Hence, $\bar{p}^0$ is set to 0.9 in all numerical experiments.

3.4.4. Benchmark Problems

An alternative routing model is presented in this section to classify the benchmark of the benefit of the proposed OFEX routing model. The benchmark model retains the hyperpath structure but employs different pricing and bidding methods that mimics how most trucks today compete for loads in the spot market. Precisely, at each city, a truck will follow a predefined order and bid for each option at the average price. This method means that it does not take advantage of the winning probability function to maximize profits, and hence will always have a winning probability of p_a^0 for any option a. Figure 3.6 details this Bidding with Average-Price Algorithm (BAPA).

Algorithm 3.3 Bidding with Average-Price Algorithm (BAPA) for City i and time t

```
1: Input: $\mathcal{Q}, H_i, g_a, l_a, u_a, V_{a^+}, c_a, \tau_a \forall a \in \mathcal{Q}$, the fallback option $\bar{f}$ and its expected profit
   $z_f$, $\bar{p}_0$, routing type (Myopic or Recursive).
2: Output: $x_a^*, \pi_a^*, \forall a \in \mathcal{A}, z^*$.
3: Initialize:
4: Set $\pi_a^* = 0, z^* = -1, x_a = 0.5(l_a + u_a), \forall a \in \mathcal{A}$.
5: Set the estimated profit $\bar{\sigma}_a$ for each option $a \in \mathcal{A}_f$ by solving Problem (3.12) while
   setting $p_a^0 = \bar{p}_0$.
6: if Myopic then
7:     Rank all options in $\mathcal{Q}$ in the descending order of $\bar{\sigma}_a/(\tau_a + 2)$.
8: else
9:     Rank all options in $\mathcal{Q}$ in the descending order of $\bar{\sigma}_a + V_{a^+}$.
10: end if
11: Set $\bar{b}_a$ (the number of competing trucks for each option) using the method described
    in Appendix B.
12: Set the fallback option $m + 1 = \bar{f}$.
13: Main loop:
14: Set $P = 1.0, z^* = 0$
15: for $a = 1$ to $m$ and $P > 0$ do
16:     Set $\pi_a^*$ using Equation (3.9), $x_a^* = 0.5(l_a+u_a)$, and $z^* = z^*+P\times\pi_a^*\times(x_a^*-c_a+V_{a^+})$.
17:     Update $\pi_a^* = \pi_a^* \times P$
18:     Update $P = P - \pi_a^*$.
19: end for
20: if $P > 0$ then
21:     Set $z^* = z^* + (V_{m+1}^+ - c_{m+1}) \times P$
22:     Set $\pi_{m+1}^* = P, x_{m+1}^* = 0$.
23: end if
24: return $x_a^*, \pi_a^*, \forall a \in \mathcal{A}, z^*$.
```

Figure 3.6. Bidding with Average-Price Algorithm (BAPA)

The BAPA algorithm has two variants (see line 6 - 10). The first, called `Myopic`, ranks feasible options in descending order of $\frac{\sigma_a^t}{\tau_a^t+2}$. Because the truck does not consider the future profits at all, the decision is myopic and likely to produce rather poor results. The second variant, called `Recursive`, attempts to be forward-looking by following the order dictated by $\bar{\sigma}_a + V_{a^+}$, where V_{a^+} is the profits that one anticipates in the next stage.

In a nutshell, the different profits made by `Myopic` and `Recursive` represents the benefits of the forward-looking strategy, and the difference between `Recursive` and COPA can be attributed to the bid pricing strategy.

3.5. Summary

The hyperpath-based decreasing order of time model is an effective and efficient methodology to solve the OFEX routing problem as a dynamic program by a polynomial time algorithm. The HyDOT algorithm allows a truck to make the best series of decisions now that account for the future ramifications of that decision. Also, it delivers all the recourse information necessary if and when the decision outcomes are realized. While this formulation has been successfully developed, its utility is best when used by a small percentage of the trucks in a trucking network. The next step required is to utilize the OFEX routing methodology as a building block for an equilibrium model that can benefit all trucks in a network.

CHAPTER 4

Equilibrium Problem Formulations

The OFEX routing model presented in Chapter 3 can give an immense competitive advantage to a truck that utilizes it for planning when other trucks do not. This situation is similar to when a commuter is driving to work, but assume that commuter is the only driver who knows the current traffic conditions. In this condition allows that commuter to exploit the knowledge most productively. However, when all drivers know the current conditions, the traffic conditions are less beneficial, as most of the viable ways to work will converge to about the same amount of time as commuters adjust routes based on the conditions. Similarly, as more trucks follow the model's guidance, it may impact the marginal gain from the initial adopters. However, as all trucks use the guidance, the network and logistics system as a whole becomes more efficient, similar to how all commuters benefit from knowing the current traffic conditions, as the variation in the drive to work, is reduced, and a lower average is expected overall. The OFEX trucking equilibrium (DTE) has great promise to create a similar effect. As all trucks plan routing strategies instead of simple matching, and guided the current best strategy, the whole network benefits, and all trucks should realize less variation in profits, creating a more stable and profitable transportation system.

This chapter explores how to model the DTE. It explores the problem background and challenges. We develop the model, the variational inequality formulation and complementarity system for the problem. The problem is found to be non-monotone, and the profit function has no closed form, so we develop a dynamic loading model that generates arc flows through the network from hyperpaths generated by the OFEX routing model. This chapter also explores some of the mathematical properties of the variational inequality problem to include integrability, continuity, and existence of a solution. Finally, the balancing algorithms and the overall DTE solution algorithms are explained.

4.1. DTE Problem Discussion

The DTE problem is defined as finding an equilibrium flow pattern of trucks over a network of trucks that all utilize an OFEX platform that brings shippers and trucks together, that results from the two competing processes mentioned in Chapter 2. The supply and demand processes are inherently dependent. Once optimal paths are identified, more trucks will change to those identified paths, which in turn decreases the expected profit. This change enables new paths to be more profitable, creating a cycle. We assume the cycle continues until an equilibrium state is reached, in which all hyperpaths with positive flow for an origin-destination (O-D) pair will have equal expected profit. We call finding this state the OFEX trucking equilibrium problem. The platform operates in a region that consists of n cities. Each city pair i to j is connected by a route with a fixed travel time τ_{ij}. The goal is to create delivery plans for all trucks when each specifies (1) the origin city $\hat{i}$, (2) the destination city $\hat{j}$, (3) the starting time and (4) the planning horizon T measured in the unit of operating interval Δ (e.g., a day or an hour). Once these parameters are given, the platform constructs a space-time expanded network (STEN) as illustrated in Figure 3.1 for the OFEX routing problem.

The assumption is that the trucks have a choice of if, when, and where to travel to pick up and deliver goods. These decisions depend on how much competition (supply, number of available trucks, and demand number of available shipping loads from that location) exist at the different locations. Modeling and solving the two interactions between truckers decisions and the competition simultaneously can produce the truck flow pattern throughout the logistics network of interest. This flow pattern can help study the logistics network operating and control policies to make the network more efficient, better for the environment, and more profitable for both shipping companies and truck drivers. The chief operating characteristics of interest in the network include profits, the percentage of goods delivered, and empty truck miles. It is possible to analyze the profitability using a profit function that explains how the profitability decreases with an increased supply of trucks.

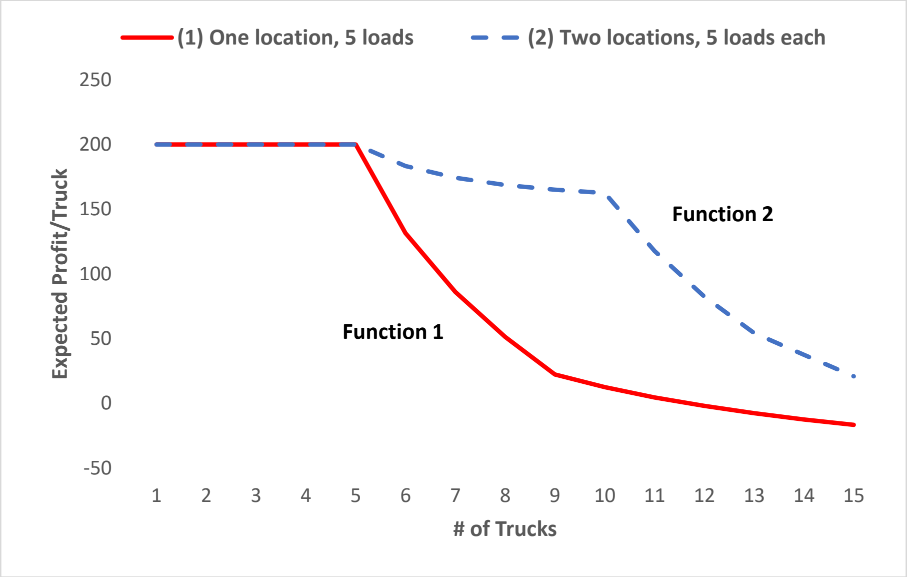


Figure 4.1. Two example node performance functions

Because of increased competition, the expected profitability of a node is defined as a decreasing function of the flow through the node, called a performance function. Two performance functions for a node are shown in Figure 4.1. Function (1) with the solid red line has only one type of load option with five loads that have a max profit of 200. Function (2) with a dashed line shows a node with two types of load options, one type has five loads with a max profit of 200, and the second type has five loads with a max profit of 150. The expected profit when there are no trucks present until the total number of loads available of the best type is flat, which represents the expected profit without competition present, where the trucks can bid the highest amount the shipper accepts. However, once the truck supply exceeds the available shipping demand, then the trucks compete, and some trucks will not make any profit. Additionally, the winning bid price also decreases from the competition, causing the expected profit to decrease dramatically from the double effect of the supply increase.

4.1.1. Challenges for the DTE Problem

The simple framework utilizing node performance functions makes it easy to solve an equilibrium problem of one O-D pair with homogeneous trucks in a simple network where only one hyperpath is active, and no loads carryover over in time to the next period. However, the node performance function framework breaks down when a location contains trucks from different hyperpaths that may bid differently, because of different requirements, and strategies downstream in the network. Hyperpaths may have different fallback options, different bid prices, thus nullifying any way to create a unifying node performance function at every node. This framework also breaks when loads unfilled carryover to the next period, creating node interactions.

Attempting to create link performance functions are even more problematic, as the links are interdependent on the other links extending from a node, in addition to being dependent on the number of trucks at a node. Therefore, a different framework is needed that can harmonize many competing hyperpaths in a network, and generate the expected profits needed to analyze the performance of the trucks and account for the dependencies of the arc flows on the hyperpath flows. The derivation of arc flows from hyperpath flows requires a specific algorithm which is explained in Section 4.3.

The ability to solve the DTE problem and apply proven solution techniques is severely limited because the profit function does not have a closed form. Another problem is accounting explicitly for the hyperpath flows, which can be an exponential number, to derive the profit function. We will look further into the characteristics and mathematical properties of the problem in Section 4.4.

4.2. DTE Model

To define the equilibrium conditions, let us first review the key inputs. On the demand side, we have g_a^t for each $a \in \mathcal{A}_f, a^- = (i,t), a^+ = (j, t+\tau_{ij})$ as the number of loads ready for pickup at t for the city pair ij represented by a. On the supply side, $B_w^{t_r}$ denotes the number of trucks that start at city r at time t_r and wish to return to city s at T. w is introduced as a shortcut to the O-D pair $r - s$, with $w^- = r$ and $w^+ = s$. The set of unique origin-destination pairs is denoted as $\mathcal{W}$. Thus, a truck is identified in this dissertation only by the pair of the origin/destination cities w and the starting time t_r. The desired arrival time of all trucks at the destination is always the end of the planning horizon T. While not considered herein, we note that the model can be easily extended to accommodate different arrival times, truck types and sizes, as long as data justify it.

Not all loads ready to be delivered in t will find a truck willing to take them. In light of this, we denote the number of loads left undelivered for link a at the end of time t as e_a^t. These loads will be carried over to the next time interval, hence increasing the total number of loads in that interval. Let d_a^t be the number of loads for link a at t. We have

$$d_a^t = e_a^{t-1} + g_a^t;\ \forall a \in \mathcal{A}_f, \forall t = 1, 2, \cdots, T, \tag{4.1a}$$

$$e_a^t = d_a^t - v_a^t;\ \forall a \in \mathcal{A}_f, \forall t = 0, 1, 2, \cdots, T, \tag{4.1b}$$

$$d_a^0 = g_a^0,\ \forall a \in \mathcal{A}_f, \tag{4.1c}$$

$$e_a^t = d_a^t = 0;\ \forall a \in \mathcal{A}_w \bigcup \mathcal{A}_e \bigcup \mathcal{A}_d, \forall t = 1, 2, \cdots, T, \tag{4.1d}$$

where v_a^t is the truck flow on link a departing at time t. Equations (4.1a - 4.1c) specifies the relationship between d_a^t, e_a^t and v_a^t. Equation (4.1d) states that activities of load movements are zero except on the loaded links. For the loaded links, the truck flow must be restricted by

$$v_a^t \leq d_a^t,\ \forall a \in \mathcal{A}_f, \forall t = 1, 2, \cdots, T. \tag{4.2}$$

The total number of trucks available at city i' and i at time t can be computed as

$$\hat{b}^t_{i'} = \sum_{a, a^+ = (i',t), a^- = (j, t-\tau_a)} v^{t-\tau_a}_a, \ \forall i' \in \mathcal{L}', t = 1, 2, \cdots, T, \tag{4.3a}$$

$$v^t_a = \hat{b}^t_{i'}, \ \forall a^- = (i', t), a^+ = (i, t), t = 1, 2, \cdots, T, \tag{4.3b}$$

$$\hat{b}^t_i = \sum_{a, a^+ = (i,t), a^- = (j, t-\tau_a)} v^{t-\tau_a}_a + \sum_{w \in W, w^- = i} B^{t_i}_w, \ \forall i \in \mathcal{L}, t = 1, 2, \cdots, T. \tag{4.3c}$$

Equation (4.3a) specifies that, at a dummy city i, the total number of trucks available equals the number of trucks arriving without carrying a load. Equation (4.3b) states that the flow on a dummy connector equals the total number of trucks available at its tail node in the same interval. Equation (4.3c) includes two parts: the first part includes all trucks arriving from other cities, carrying a load or not, and the second part is the number of trucks starting their tour at t (these trucks were not in the workforce in the earlier intervals).

Each truck in $B^{t_r}_w$ will follow a profit-maximizing hyperpath by solving the OFEX routing problem. Denote such a hyperpath as $h^{wt_r} \in \mathcal{H}^{t_r}_w$. Further, let $\bar{p}^{wt_r}_{ah}$ be the expected probability of choosing link a on $h^{t_r}_w$ as defined as:

$$\bar{p}_{ah} = \sum_{k \in \mathcal{K}_h} \lambda_{kh} \delta^k_{ah}, \forall a \in \mathcal{E}_h, \tag{4.4}$$

and $x^{wt_r}_{ah}$ be the bidding price for link a on $h^{t_r}_w$. Due to competitions, $B^{t_r}_w$ may be distributed to multiple hyperpaths. Let $f^{wt_r}_h$ be the flows of trucks in $B^{t_r}_w$ assigned to hyperpath h^{wt_r}, and $u^{wt_r}_h$ be the *actual profit* realized by following the routing and bidding decisions

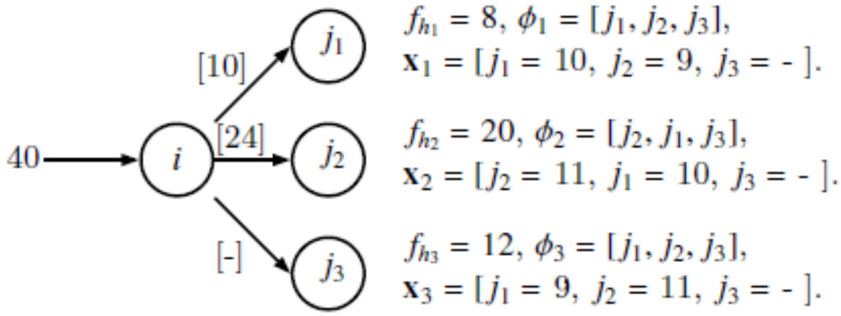

40
i
[10]
[24]
[-]
j1
j2
j3
f_{h1} = 8, ϕ1 = [j1, j2, j3],
x1 = [j1 = 10, j2 = 9, j3 = -].
f_{h2} = 20, ϕ2 = [j2, j1, j3],
x2 = [j2 = 11, j1 = 10, j3 = -].
f_{h3} = 12, ϕ3 = [j1, j2, j3],
x3 = [j1 = 9, j2 = 11, j3 = -].

Figure 4.2. Illustration of hyperpath flow loading

prescribed in h^{wt_r}. The flow conservation condition dictates that

$$B_w^{t_r} = \sum_{h^{wt_r} \in \mathcal{H}_w^{t_r}} f_h^{wt_r}, \forall w \in W, t_r = 0, 1, \cdots, T. \tag{4.5}$$

Let $\mathbf{f}$, $\mathbf{v}$ and $\mathbf{u}$ be the vectors of hyperpath truck flows, link truck flows, and hyperpath profits respectively. The critical question here is how to evaluate $\mathbf{x}$ and $\mathbf{u}$ provided $\mathbf{f}$. The simplest way would be to use:

$$v_a = \sum_{w \in W} \sum_{t_r \in \mathcal{T}} \sum_{h \in \mathcal{H}_w^{t_r}} \bar{p}_{ah}^{wt_r} f_h^{wt_r}, \forall a \in \mathcal{E}, \tag{4.6}$$

$$u_h^{wt_r} = \sum_{a \in h^{wt_r}} \bar{p}_{ha_t}^{w} (x_{ah}^{wt_r} - c_a), \forall h^{wt_r} \in \mathcal{H}^{wt_r}, \tag{4.7}$$

where $\bar{p}_{ah}^{wt_r}$ is defined in Equation (4.4). This naive method, however, is questionable, because it does not properly account for the competitions among flows from different hyperpaths. This issue can be best illustrated using an example. Figure 4.2 shows three hyperpaths that are competing for loads on three links originated at city i. According to the bidding order and the price associated with these hyperpaths, flows on both hyperpaths 1 and 3 have the highest preference for link ij_1. Hence, in total, there will be 20 trucks bidding for the ten loads on ij_1. Because the bidding price for hyperpath 3 is lower, all loads will be won by flow from hyperpath 3. This effectively means that the actual probability of using link ij_1 is zero for hyperpath 1, which differs from the original winning probability (as a link with a zero choice probability would not be included in the hyperpath). It is clear from this example that distributing loads to each hyperpath requires executing (or simulating) the bidding process according to a physically meaningful rule. This procedure is called dynamic network loading (DNL), to borrow a term from the literature on dynamic traffic assignment.

We will propose a DNL procedure in the next section. It suffices here to say that, using DNL, we can determine $p_{ah}^{wt_r}$, the actual percentage of flows from hyperpath h^{wt_r} that use link a. Hence, the link flows, and hyperpath profits can be redefined as:

$$v_a = \sum_{w \in W} \sum_{t_r \in \mathcal{T}} \sum_{h \in \mathcal{H}_w^{t_r}} p_{ah}^{wt_r} f_h^{wt_r}, \ \forall a \in \mathcal{E}, \tag{4.8}$$

$$u_h^{wt_r} = \sum_{a \in h^{wt_r}} p_{ah}^{wt_r} (x_{ah}^{wt_r} - c_a), \ \forall h^{wt_r} \in \mathcal{H}^{wt_r}. \tag{4.9}$$

With the above definitions, the DTE conditions can be stated as follows:

$$u_h^{wt_r} \leq \nu^{wt_r} \ \forall h^{wt_r} \in \mathcal{H}^{wt_r}, w \in W, t_r \in \mathcal{T}, \tag{4.10a}$$

$$f_h^{wt_r} > 0 \rightarrow u_h^{wt_r} = \nu^{wt_r}, \forall h^{wt_r} \in \mathcal{H}^{wt_r}, w \in W, t_r \in \mathcal{T}, \tag{4.10b}$$

where $f_h^{wt_r}$ satisfies the flow conservation conditions (4.5) as well as non-negativity conditions. $u_h^{wt_r}$ is evaluated using (4.9), which is an implicit function of $f_h^{wt_r}$. ν^{wt_r} is the equilibrium maximum expected profit for trucks from O-D pair w departing at t_r.

4.2.1. DTE assumptions

The DTE problem model and solution relies upon the hyperpath-based routes constructed by HyDOT. The assumptions from the OFEX routing model in Chapter 3 still apply, along with a few more. We shall assume that the following inputs to the DTE problem can be reliably predicted between now ($t = 0$) and $t = T$, based on the historical information, consistent with the assumptions made in the OFEX Routing Problem:

(1) The number of vehicles available to carry loads on each O-D pair w, denoted as B_w, at all times t;

(2) The desired starting and ending times within $t = 0 \cdots T$, and locations within $\mathcal{N}$ for all trucks' tours; and

(3) The load arrival rates for all locations and times within the planning horizon.

4.2.2. Use of the OFEX Routing Problem

The OFEX routing algorithm serves as one of the critical engines for the DTE problem, similar to how the shortest path algorithm serves the traffic assignment and transit assignment problems. However, the Hyperpath DOT (HyDOT) algorithm is modified from the version in Figure 3.4 for the equilibrium application. First, the model no longer needs to estimate the number of competing trucks, b_a for each option using a logit model and an empty truck model that were exogenous to the original model, and the most problematic elements of that model. Instead, the algorithm utilizes the flows generated from the last iteration of the DTE model, an exact number of trucks, endogenous to the model to count the number of competing trucks, b_a. This value of b_a is still an estimate, but as the DTE algorithm iterates, the number of competing trucks becomes increasingly more accurate.

The empty truck model is also completely unnecessary in the equilibrium model, as the model will account for all empty movements and truck arrivals endogenously as well.

4.3. Dynamic Network Loading

This section details the dynamic network loading (DNL) procedure, which produces the actual link choice probability vector $\mathbf{p} = \{p_{ah}^{wt_r}, \forall a, h^{wt_r}, w, t_r\}$ according to the hyperpath flow vector $\mathbf{f}$. At the core of the DNL procedure is the bidding process used when multiple hyperpaths are competing for the same set of loads originating from a city.

4.3.1. Loading Bidding Process

To study the bidding process, recall at each city i, the set of all outgoing links is $\mathcal{O}_i$, and the number of loads on each $a \in \mathcal{O}_i$ is g_a ($g_a > 0$ only if a is a loaded link). Let $\mathcal{H} = \{h_k, k = 1, \cdots, K\}$ be the set of hyperpaths that pass through city i and may compete for loads on $\mathcal{O}_i$. For simplicity, the superscripts related to O-D pairs and time index are suppressed here. At node i, each h_k has its own bidding vector $\phi_k = \{\phi_k^1, \phi_k^2, ..., \phi_k^{m_k}\}$ and biding price vector $\mathbf{x}_k = \{x_k^1, x_k^2, \cdots, x_k^{m_k}\}$, where m_k is the number of links in the biding list. A truck following a hyperpath h_k always bids for loads on link ϕ_k^1 first. Assume that all trucks strictly follow their orders, then we can obtain the total number of trucks bidding for each ϕ_k^1. For every link a, define $\mathcal{H}_a = \{h_l | \phi_l^1 = a\}$. The total number of competing trucks and the demand/supply ratio on each link a are

$$b_a = \sum_{k \in \mathcal{H}_a} f_{h_k}, \vartheta_a = \frac{g_a}{b_a}, \forall a \in \mathcal{O}_i, \tag{4.11}$$

respectively. Each link b_a may consist of flows from multiple hyperpaths, and only the hyperpaths offering the lowest price would win loads. The question is how many loads should be awarded to the winning hyperpaths. One option is to give as many as possible, which implies that loads will be exhausted on any link a where $g_a \leq b_a$. In a competitive market, this method means that trucks are likely to lose the opportunity to get any load if their first choice is too competitive for them to win. In reality, trucks may avoid such an outcome by actively engaging in bidding for multiple links. To capture such behavior, we propose to award loads based on a *minimum assignment ratio* [see 43, for the application of a similar idea in the capacitated traffic assignment problem], defined as

$$\vartheta^* = \min\{\vartheta_a, \forall a \in \mathcal{O}_i\}.$$

Define v_{ah_l} as the flow on link a contributed by hyperpath h_k and rank the hyperpath set $\mathcal{H}_a$ in the ascending order of their bidding price for a. Thus, h_1 in $\mathcal{H}_a$ is the hyperpath that has highest preferences for a and offers the lowest price. Let l^* be the highest-rank hyperpath in $\mathcal{H}_a$ such that $\sum_{l=1}^{l^*} f_{h_l} \geq \vartheta^* b_a$. If l^* does not exist, then we have

$$v_{ah_l} = f_{h_l}, \forall l; \tag{4.12}$$

Otherwise

$$v_{ah_l} = \begin{cases} f_{h_l}, & \forall l = 1, \cdots, l^* - 1 \\ \vartheta^* b_a - \sum_{l=1}^{l^*-1} f_{h_l} & l = l^* \end{cases}. \tag{4.13}$$

Using this method, only the links with the minimum assignment ratio may deplete all loads after the bidding for the first choice is over. If there are still flows on some hyperpaths left unassigned to a link, another round of bidding will be simulated.

The above process will be repeated after (1) removing the links whose loads have become zero from ϕ_k and $\mathbf{x}_k$, and (2) updating g_a, f_{h_k} and b_a. When all flows on all hyperpaths are assigned according to the above bidding process, the link choice probability vector $\mathbf{p}$ can be obtained from v_{ah_l}. A detailed description of the iterative bidding process, as well as the overall network loading procedure, is presented in the next section.

4.3.2. DNL Algorithm

The pseudocode of the dynamic network loading (DNL) algorithm is given in Figure 4.3. We define, for each by hyperpath h between O-D pair w starting at t_r, (1) $v_{ah}^{wt_r}$ as the flow contributed to link a in the STEN, (2) $\check{f}_{ith}^{wt_r}$ as the residual flow that arrive at node (i, t) and (3) $\phi_{ith}^{wt_r}$ as the bidding order at node (i, t).

The algorithm runs forward in time starting at the initial time, $t = 0$. The initial section of the algorithm (lines 4 - 12) totals the truck flow at a node $\hat{b}_r^t$ and initialize the total residual truck flow by hyperpath $\check{f}_{ith}^{wt_r}$. Next, for each node in the STEN (i, t), (lines 14-18) leftover loads are carried over to the next time. Then, while truck flow remains unassigned (lines 19-31), the hyperpaths are first partitioned into groups (lines 20-24) that have the same first choice outgoing node, and the demand for the highest remaining link is totaled as b_a. Next, ϑ^* is set to the proportion of residual flow yet to be assigned, to the minimum ratio of the corresponding load over the residual summed demands (line 24). Note, if $\vartheta^* \geq 1$, then all residual flows are assigned to their current preferred choices, and the algorithm moves to the next time increment. If $\vartheta^* < 1$, the algorithm checks the bid prices (lines 25-31) by ranking the bids from all hyperpaths in ascending order of $x_{ah}^{wt_r}$, and assigning link flow $v_{ah}^{wt_r}$, based on the winning bids by Equations (4.12 - 4.13). The algorithm repeats the assignment by updating the residual loads and trucks available and updating the new lowest bidding option if necessary (lines 27-30). Finally, the probabilities $\mathbf{p}$ and profit vectors $\mathbf{u}$ are determined, (lines 32-35).

In general, a single node is removed from the bidding order sets at each primary while loop structure of the procedure (lines 9-33). However, the algorithm can saturate more than one node or even all nodes (it occurs when all outgoing arcs saturate simultaneously). Therefore, in the worst case this loop is performed at most $|a^+|$ times for each node i, $\sum_{i \in \mathcal{L}} |a^+| = (|\mathcal{A}_f| + 1)$. Inside the main while loop is an additional while loop to account for the bidding competition, which in the worst case checks all hyperpaths from all O-D pairs, $|H|$. The summation of the flows by nodes and by hyperpaths creates at most $|\mathcal{A}| \times |H| \times n + |H|$ calculations for each time. It follows that the entire algorithm structure is executed at most $|H| \times ((|\mathcal{A}_f| + 1) + |\mathcal{A}| \times n + 1) \times (T - 1)$ times. The algorithm without any restrictions is non-polynomial because the number of possible hyperpaths grows exponentially as the network grows. However, in a practical setting, the number of utilized hyperpaths can always be artificially capped.

4.3.3. Example

We now demonstrate how the bidding process in the DNL algorithm works using the example shown in Figure 4.2. Recall that there are three unique hyperpaths with different flows at node i, each with a bidding order and price vectors $\phi_k, \mathbf{x}_k,\ k = 1, 2, 3$ respectively.

Algorithm 4.1 Dynamic Network Loading (DNL) Algorithm

```
**Inputs:** $\mathfrak{G}(\mathcal{N}, \mathcal{E}, \mathcal{T})$, $B_w^{t_r}, g_a^t, f_h^{wt_r}, \tau_a^t, x_{ah}^{wt_r}, c_a, \phi_{ith}^{wt_r}$.
**Initialization**: Set $v_{ah}^{wt_r} = \emptyset, \check{f}_{ith}^{wt_r} = \emptyset, \forall i \in \mathcal{L}, t \in \mathcal{T}, w \in \mathcal{W}, h \in \mathcal{H}, \hat{b}_i^t = \emptyset, \forall i \in \mathcal{L}, t \in \mathcal{T}$.
**for** $t = 0$ to $T - 1$ **do**
    **for all** $w, t_r$ **do**
        **if** $t_r = t$ **then** $\hat{b}_r^t = \hat{b}_r^t + B_w^{t_r}$, $f_{rth}^{wt} = \check{f}_{rth}^{wt} = \check{f}_{rth}^{wt} + B_w^{t_r}$
        **end if**
        **for all** $h \in \mathcal{H}^{wt_r}$ **do**
            **for all** $a$ on $h$ **do**
                if $t + \tau_a^t \leq T$, $\hat{b}_{a^+}^{t+\tau_a^t} = \hat{b}_{a^+}^{t+\tau_a^t} + v_{ah}^{wt_r}$, $f_{a^+(t+\tau_a^t)h}^{wt_r} = \check{f}_{a^+(t+\tau_a^t)h}^{wt_r} = \check{f}_{a^+(t+\tau_a^t)h}^{wt_r} + v_{ah}^{wt_r}$
            **end for**
        **end for**
    **end for**
    **for all** $(i, t)$ **do**
        **for all** $a$ such that $a^- = (i, t)$ **do**
            **if** $t = 0$ **then** $e_a^0 = g_a^0$
            **else** $e_a^t = g_a^t + e_a^{t-1}$
            **end if**
        **end for**
        **while** $\sum_{w \in \mathcal{W}} \sum_{h \in \mathcal{H}_w} \check{f}_{ith}^{wt_r} > 0$ **do**
            Update the bidding order $\phi_{ith}^{wt_r}$ for all $w, t_r, h$.
            **for all** $w, t_r$, $h \in \mathcal{H}_w^{t_r}$ **do**
                Set $a$ as the first in $\phi_{ith}^{wt_r}$, $b_a = b_a + \check{f}_{ith}^{wt_r}$
            **end for**
            Set $\vartheta^* = \min(1, \min_a\{\frac{e_a^t}{b_a}, \forall a\})$
            Construct for each $a$ a temporary hyperpath set $\mathcal{H}_a$ and rank the hyperpaths in $\mathcal{H}_a$ in the
    ascending order of $x_{ah}^{wt_r}$
            Set $l^*$ to the highest-rank hyperpath in $\mathcal{H}_a$ such that $\sum_{l=1}^{l^*} \check{f}_{ith_l}^{wt_r} \geq \vartheta^* b_a$, Set $v_{ah_l}^{wt_r}$ by
    Equations (4.12 - 4.13).
            **for all** $a$ such that $a^- = (i, t)$ **do** $e_a^t = e_a^t - v_{ah_l}^{wt_r}$, $\check{f}_{ith_l}^{wt_r} = \check{f}_{ith_l}^{wt_r} - v_{ah_l}^{wt_r}$
                **if** $e_a^t = 0, \forall a^- = (i, t)$ **then** Remove the first choice from the bidding order
    $\phi_{ith}^{wt_r}, \forall w, t_r, h$
                **end if**
            **end for**
        **end while**
        Set $p_{ah}^{wt_r} = \frac{v_{ah}^{wt_r}}{f_{ith}^{wt_r}}$, $\forall w, t_r, h, a$
    **end for**
**end for**
Set $u_h^{wt_r} = \sum_{a \in h^{wt_r}} p_{ah}^{wt_r}(x_{ah}^{wt_r} - c_a)$, $\forall h^{wt_r} \in \mathcal{H}^{wt_r}$
**Outputs:** $v_{ah}^{wt}, f_{ih}^{wt}, \hat{b}_i^t, e_a^t, p_{ah}^{wt}, u_h^{wt_r}$
```

Figure 4.3. Pseudocode of the dynamic network loading algorithm

Table 4.1. Dynamic network loading results for the example shown in Figure 4.2

Iter.	Arc	$(i, j_1), a = 1$	$(i, j_2), a = 2$	$(i, j_3), a = 3$
1	Residual Loads, e	10	24	∞
	Sum of demand b_a, ratio ϑ	$20, 0.5$	$20, 1.2$	null
	Min Ratio ϑ^*	0.5	0.5	null
	Winning bid demand $\dot{b}_a$, ratio $\hat{\vartheta}^*$	$12, 5/6$	null	null
	Assigned Flow, $\mathbf{v}$	$v_{13} = 10$	$v_{22} = 10$	null
2	Residual Loads, e	0	14	∞
	Sum of demand b_a, ratio ϑ	null	$20, 0.7$	null
	Winning bid demand $\dot{b}_a$, ratio $\hat{\vartheta}^*$	null	8,1	null
	Assigned Flow, $\mathbf{v}$	null	$v_{21} = 8$	null
	Residual Loads, e	0	6	∞
	Winning bid demand $\dot{b}_a$, ratio $\hat{\vartheta}^*$	null	12,0.5	null
	Assigned Flow, $\mathbf{v}$	null	$v_{22} = 5 + 10 = 15, v_{23} = 1$	null
3	Residual Loads, e	0	0	∞
	Sum of demand b_a, ratio ϑ	null	null	6,∞
	Assigned Flow, $\mathbf{v}$	null	null	$v_{32} = 5, v_{33} = 1$
Outputs	Assigned Flow, $\mathbf{v}$	$v_{13} = 10$	$v_{21} = 8, v_{22} = 15, v_{23}$	$v_{32} = 5, v_{33} = 1$
	Probabilities, $\mathbf{p} = \frac{v_{ah}}{f_h}$	$p_{11} = 0, p_{12} = 0, p_{13} = \frac{10}{12}$	$p_{21} = 1, p_{22} = \frac{15}{20}, p_{23} = \frac{1}{12}$	$p_{31} = 0, p_{32} = \frac{5}{20}, p_{33} = \frac{1}{12}$
	Hyperpath profit/truck, $\mathbf{u}$, $\frac{\sum \sigma_a}{f_h}$	$u(f_1) = \frac{32}{8} = 4$	$u(f_2) = \frac{90-25}{20} = 3.25$	$u(f_3) = \frac{40+6-5}{12} = 3.417$

As shown in the figure

$$\phi_1 = \{1, 2, 3\}, \phi_2 = \{2, 1, 3\}, \phi_3 = \{1, 2, 3\}, \mathbf{x}_1 = \{10, 9, -\}, \mathbf{x}_2 = \{11, 10, -\},$$

$$\mathbf{x}_3 = \{9, 11, -\}, f_{h_1} = 8, f_{h_2} = 20, f_{h_3} = 12, g_1 = 10, g_2 = 24, g_3 = 0.$$

For simplicity, set cost $c_a = 5, \forall a$.

In the first round, the trucks are bidding for their first choice. The flows from h_1 and h_3 are both competing for loads to 1 while h_3 is the only hyperpath that prefers 2. Thus, the supply at link 1 is $b_1 = 20$ (8 from h_1 and 12 from h_3) and the supply at link 2 is 20 (all from h_2). At the beginning, the available loads at the two links are $g_1 = 10$ and $g_2 = 24$. Hence, the assignment ratios are $\vartheta_1 = \frac{10}{20}$ and $\vartheta_2 = \frac{24}{20}$. Since $\frac{10}{20} < \frac{24}{20}$, the minimum assignment ratio $\vartheta^* = \frac{10}{20} = 0.5$. For link 1, both h_1 and h_3 compete for its load. Ranking them based on the bidding price leads to a set $\mathcal{H}_1 = \{h_3, h_1\}$. Using Equations (4.12 - 4.13), we find $l^* = 1$, hence $v_{1h_3} = 10, v_{1h_1} = 0$. That is, all loads on link 1 will be awarded to h_3. For link 2, only h_2 is competing for its loads. However, the maximum loads allowed to be awarded to h_2 is capped by $\vartheta^* \times b_2 = 10$. After the first round, thus, loads on link 1 are depleted and removed from ϕ_k, and there are still 24 - 10 = 14 loads left on link 2.

In the second round all unassigned flow (8 from h_1, 10 from h_2 and 2 from h_3) are competing for the 14 loads on link 2. The minimum assignment ratio is $\vartheta^* = \frac{14}{20} = 0.7$. Ranking the hyperpaths based on the bidding price leads to $\mathcal{H}_2 = \{h_1, h_2, h_3\}$, with h_2 and h_3 tied with the same bidding price. In this case, $l^* = 2$, hence, $v_{2h_1} = 8$. Because h_2 and h_3 have the same price, they share the remaining six loads proportional to their flow, that is, $v_{2h_2} = 6/12 = 5$ and $v_{2h_2} = 6/12 = 1$. After the second round, loads on link 2 are exhausted.

In the third and the final round, the remaining flows (5 from h_2, and 1 from h_3) are all assigned to link 3, which is a fallback option that has no load limitation. The calculations are summarized in Table 4.1.

4.4. DTE Variational Inequality Formulation

The DTE conditions (4.10) can be restated as the following complementarity conditions:

$$u_h^{wt_r} - \nu^{wt_r} \leq 0; \forall h^{wt_r} \in \mathcal{H}^{wt_r}, w \in W, t_r \in \mathcal{T}, \tag{4.14a}$$

$$f_h^{wt_r}(u_h^{wt_r} - \nu^{wt_r}) = 0, \forall h^{wt_r} \in \mathcal{H}^{wt_r}, w \in W, t_r \in \mathcal{T}, \tag{4.14b}$$

$$f_h^{wt_r} \geq 0; \forall h^{wt_r}, w, t_r, \tag{4.14c}$$

$$\sum_{h \in \mathcal{H}^{wt_r}} f_h^{wt_r} = B^{wt_r}, \forall w, t_r. \tag{4.14d}$$

Let **f** and **u** be the vectors for hyperpath flows and profits, respectively. Note that **u** is an implicit function of **f**, in that **u** can be evaluated for any given **f** using the DNL algorithm presented in the previous section. We further define the feasible set for **f** as

$$\mathcal{F} = \{\mathbf{f} = \{f_h^{wt_r}\} \,\big|\, f_h^{wt_r} \quad \text{satisfies Conditions} \quad (4.14c - 4.14d)\}.$$

It is easy to show [see, e.g., 71, 12] that the complementarity system (4.14) is equivalent to the following variational inequality problem (VIP)

$$\text{VIP}(\mathbf{u}, \mathcal{F}) : \text{Find} \quad \mathbf{f}^* \in \mathcal{F} \quad \text{such that}$$

$$\langle \mathbf{u}(\mathbf{f}^*), \mathbf{f} - \mathbf{f}^* \rangle \leqslant 0 \quad \text{holds} \quad \forall \mathbf{f} \in \mathcal{F}, \tag{4.15}$$

where $\langle a, b \rangle = a^T b$. Because the profit function $\mathbf{u}(\mathbf{f}^*)$ is not available in closed form, the VIP (4.15) is not particularly amenable to analysis. In the following, we first show that the Jacobian matrix of $\mathbf{u}(\mathbf{f}^*)$ is asymmetric, making it impossible to reformulate the VIP as a standard optimization problem [see, e.g., 51]. To see this, consider a simple example in which trucks from two distinct hyperpaths, h_1 and h_2, compete for one load on link 1. The flow on each hyperpath is denoted as $f_k, k = 1, 2$. Assume that their bidding price is the same for the load and that the two trucks have a different fallback link, links 2 for the truck on h_1 and 3 for the truck on h_2. Let σ_i be the profit earned on link i. Note that since the trucks bid for link 1 at the same price, the profit σ_1 is the same, and the trucks will receive the loads proportional to the total flow. Therefore, the expected profit for each hyperpath is:

$$\begin{aligned} u_1(f_1, f_2) &= \frac{f_1\sigma_1}{f_1 + f_2} + (1 - \frac{f_1}{f_1 + f_2})\sigma_2, \\ u_2(f_1, f_2) &= \frac{f_2\sigma_1}{f_1 + x_2} + (1 - \frac{f_2}{f_1 + f_2})\sigma_3. \end{aligned}$$

Then, the cross partial derivatives of u_i are given by

$$\begin{aligned} \frac{\partial u_1}{\partial f_2}(f_1, f_2) &= \frac{f_1(\sigma_2 - \sigma_1)}{(f_1 + f_2)^2}, \\ \frac{\partial u_2}{\partial f_1}(f_1, f_2) &= \frac{f_2(\sigma_3 - \sigma_1)}{(f_1 + f_2)^2}. \end{aligned}$$

These are clearly different when $f_1(\sigma_2 - \sigma_1) \neq f_2(\sigma_3 - \sigma_1)$. Therefore, the Jacobian matrix of $\mathbf{u}$ is asymmetric in general.

Not only is the profit function $\mathbf{u}(\mathbf{f}^*)$ asymmetric, it can also fail to be monotone on $\mathcal{F}$. By definition, the profit function $\mathbf{u}$ is monotone on $\mathcal{F}$ if

$$\langle \mathbf{u}(\mathbf{f}_1) - \mathbf{u}(\mathbf{f}_2), \mathbf{f}_1 - \mathbf{f}_2 \rangle \geqslant 0, \forall \mathbf{f}_1, \mathbf{f}_2 \in \mathcal{F}.$$

Intuitively, everything else equal, the profit on a hyperpath cannot increase when more trucks are assigned to it, because more trucks would only intensify competition, which in turn reduces profitability. However, due to interactions among competing hyperpaths, monotonicity is not always held. Consider the example given in Figure 4.2 where there are three hyperpaths. Assume $\mathbf{f}_1 = \{8, 20, 12\}$, and $\mathbf{f}_2 = \{11, 19, 10\}$. Applying the proposed loading algorithm we obtain $\mathbf{u}(\mathbf{f}_1) = \{4, 3.25, 3.417\}$, and $\mathbf{u}(\mathbf{f}_2) = \{4, 2.526, 4\}$. We have

$$\langle \mathbf{u}(\mathbf{f}_1) - \mathbf{u}(\mathbf{f}_2), \mathbf{f}_1 - \mathbf{f}_2 \rangle = \langle (0, 0.724, -0.583), (-3, 1, 2) \rangle = -0.443 < 0.$$

This example shows that the profit function is not monotone in general. The lack of monotonicity means, among other things, that the DTE may not be unique.

Having addressed the issue of uniqueness, we proceed to examine the existence of DTE. To this end, we first define the lower semi-continuous function.

Definition 2 (lower semi-continuous function)**.** *A function $f : R^n \to R$ is lower semi-continuous at point $\bar{x}$, if $\forall \epsilon > 0, \exists \eta > 0$ such that when $||x - \bar{x}|| < \eta, f(\bar{x}) - f(x) \leq \epsilon$. Equivalently, $f(\bar{x}) \leq \liminf_{x \to \bar{x}} f(x)$.*

[22] establishes the following result.

Lemma 2 (Ky Fan Inequality)**.** *Suppose that X is a compact subset of R^n and that $\Phi : X \times X \to R$ is a function satisfying*

$$\forall y \in X, x \to \Phi(x, y) \text{ is lower semi-continuous, and} \tag{4.18}$$

$$\forall x \in X, y \to \Phi(x, y) \text{ is concave.} \tag{4.19}$$

Then there exists $x^* \in X$ *such that*

$$\sup_{y \in X} \Phi(x^*, y) \le \sup_{y \in X} \Phi(y, y). \tag{4.20}$$

We proceed to show that the profit function is a lower semi-continuous function for our setting.

Lemma 3. *The hyperpath profit* $\mathbf{u}$ *is a lower semi-continuous function of the hyperpath flow* $\mathbf{f}$*.*

Proof. Consider the profit function $\mathbf{u}$ from the DTE problem in Equation 4.9 . The profit function is dependent on the probabilities $p_{ah}^{wt_r}$, which in turn depends on $v_{ah}^{wt_r}$ and $f_{ith}^{wt_r}$ (See Line 32 in Figure 4.3). $f_{ith}^{wt_r}$ is the hyperpath flow $f_h^{wt_r}$ passing node (i,t), hence it must be continuous in $f_h^{wt_r}$. $v_{ah}^{wt_r}$ are determined in Equations (4.12 - 4.13), hence also continuous in $f_h^{wt_r}$ in general. Discontinuity may occur in a special case, when $f_{ith}^{wt_r} = 0$, the residual load $e_a^t = 0$ and the hyperpath h^{wt_r} happens to have a as its first choice but with a losing bid.

Let us illustrate this special case with an example. Consider two hyperpaths, both having only two options at a node: a_1: win the load, or a_2: travel empty to the same node. Let the bid price of the two paths satisfy $x_{a_1}^1 < x_{a_1}^2$, i.e., hyperpath 2 will lose since it offers a higher price. Assume the total flow $f = g$, where g is the number of available loads on option a. Without loss of generality, assume the profit of winning option a_1 is: $\sigma_{a_1}^1 = 20, \sigma_{a_1}^2 = 25$ and the cost of option a_2 = -10 for both hyperpaths. For a flow pattern $f_1 = g, f_2 = 0$, the link choice probability $p_{a_1 1} = 1, p_{a_1 2} = 0$ and the profit

$u_1(f_1, f_2) = 20, u_2(f_1, f_2) = -10$. For another flow pattern $f_1 = g - \epsilon, f_2 = \epsilon$, however,

$$\lim_{\epsilon \to 0^+} p_{a_2 2}(f_1 = g - \epsilon, f_2 = \epsilon) = 1.0 > 0, \lim_{\epsilon \to 0^+} u_2(f_1 = g - \epsilon, f_2 = \epsilon) = 25 > -10$$

Therefore, both the link choice probability and the profit are discontinuous at the point $(f_1 = g, f_2 = 0)$, but are lower semi-continuous. □

We are now ready to prove the existence of DTE, using similar arguments made by [32].

Theorem 1. *There exists at least one solution satisfying the variational inequality VIP(*$\mathbf{u}(\mathbf{f})$*,*$\mathcal{F}$*).*

Proof. To show the second condition of the Ky Fan's Inequality, set $\Phi(\mathbf{f}, \mathbf{y}) = \langle \mathbf{u}(\mathbf{f}), \mathbf{f} - \mathbf{y} \rangle$ such that the function Φ is concave in $\mathbf{y}$ and satisfies $\Phi(\mathbf{y}, \mathbf{y}) = 0$. Now the proof can be completed by showing the first condition, i.e. $\mathbf{f} \to \Phi(\mathbf{f}, \mathbf{y})$ is lower semi-continuous for every value of $\mathbf{y}$. Since $\mathbf{u}$ is lower semi-continuous the following results: $\forall \epsilon' = \frac{\epsilon}{2||\bar{\mathbf{f}} - \mathbf{y}||} > 0 \; \exists \eta > 0$ such that $||\mathbf{f} - \bar{\mathbf{f}}|| < \eta \Rightarrow \mathbf{u}(\mathbf{f}) \geq \mathbf{u}(\bar{\mathbf{f}}) - \epsilon' \Leftrightarrow \forall \epsilon' > 0 \; \exists \eta > 0$ such that $||\mathbf{f} - \bar{\mathbf{f}}|| \Rightarrow -\mathbf{u}(\mathbf{f}) \leq -\mathbf{u}(\bar{\mathbf{f}}) + \epsilon'$. Then it follows that:

$$\begin{aligned}
\langle -\mathbf{u}(\mathbf{f}), \mathbf{f} - \mathbf{y} \rangle \leq & \langle -\mathbf{u}(\bar{\mathbf{f}}), \mathbf{f} - \mathbf{y} \rangle + \epsilon' ||\mathbf{f} - \mathbf{y}|| \\
\leq & \langle -\mathbf{u}(\bar{\mathbf{f}}), \bar{\mathbf{f}} - \mathbf{y} \rangle + \langle -\mathbf{u}(\bar{\mathbf{f}}), \mathbf{f} - \bar{\mathbf{f}} \rangle + \epsilon' ||\mathbf{f} - \mathbf{y}|| \\
\leq & \langle -\mathbf{u}(\bar{\mathbf{f}}), \bar{\mathbf{f}} - \mathbf{y} \rangle + ||\mathbf{u}(\bar{\mathbf{f}})|| ||\mathbf{f} - \bar{\mathbf{f}}|| + \epsilon' ||\mathbf{f} - \bar{\mathbf{f}}|| + \epsilon' ||\bar{\mathbf{f}} - \mathbf{y}|| \\
\leq & \langle -\mathbf{u}(\bar{\mathbf{f}}), \bar{\mathbf{f}} - \mathbf{y} \rangle + (\epsilon' + ||\mathbf{u}(\bar{\mathbf{f}})||)\eta + \frac{\epsilon}{2} \\
\leq & \langle -\mathbf{u}(\bar{\mathbf{f}}), \bar{\mathbf{f}} - \mathbf{y} \rangle + \epsilon \text{ if } \eta \leq \frac{\epsilon}{2(\epsilon' + ||\mathbf{u}(\bar{\mathbf{f}})||)},
\end{aligned}$$

where the last term is uniformly bounded, because $\mathbf{u}(\bar{\mathbf{f}})$ is bounded by $M = \sum_{a \in \mathcal{A}_f} \sigma_a$. This proves that the function $\mathbf{f} \rightarrow \Phi(\mathbf{f}, \mathbf{y})$ is lower semi-continuous on the set $\mathcal{F}$, and Ky Fan's Inequality may be applied now that both conditions are met. Setting $\sup_{\mathbf{y} \in \mathcal{F}} \Phi(\mathbf{y}, \mathbf{y}) = 0$, and $\sup_{\mathbf{y} \in \mathcal{F}} \Phi(\mathbf{f}^*, \mathbf{y})$ also reaches its suprema when $\mathbf{f}^* = \mathbf{y}$, so $\sup_{\mathbf{y} \in \mathcal{F}} \Phi(\mathbf{f}^*, \mathbf{y}) = 0$. Therefore, there exists $\mathbf{f}^* \in \mathcal{F} \ni$:

$$\forall \mathbf{y} \in \mathcal{F}, \; \langle \mathbf{u}(\mathbf{f}^*), \mathbf{y} - \mathbf{f}^* \rangle \leq 0.$$

□

4.5. Hyperpath Assignment Algorithms

The idea of MSA is fairly simple. In each iteration k, it first identifies the hyperpath with the highest profit, and shifts flow from other hyperpaths to it according to step size. Let $\mathbf{f}(k) = \{\cdots, f_h^{wt_r}(k), \cdots\}$ be the hyperpath flow vector at iteration k, and $h^{wt_r *}$ be the hyperpath that has the highest profit for O-D pair w at t_r after loading $\mathbf{f}(k)$ to the network. Allocating all truck flows to $h^{wt_r *}$, and zero to all other hyperpaths for each O-D pair and departure time will lead to a target flow pattern $\mathbf{y}^k$. Then, the new hyperpath flow vector is given by

$$\mathbf{f}(k+1) = \mathbf{f}(k) + \beta^k(\mathbf{y}(k) - \mathbf{f}(k)). \tag{4.21}$$

Following the literature, the convergence at a solution $\mathbf{f} = \{\mathbf{f}^{wt_r}, \forall w, t_r\}$ is measured using a relative gap function defined as

$$r = \max_{w \in \mathcal{W}, t_r \in \mathcal{T}} \frac{\langle \mathbf{u}^{wt_r}(\mathbf{f}), \mathbf{y}^{\mathbf{wt_r}} - \mathbf{f}^{wt_r} \rangle}{\langle \mathbf{u}^{wt_r}(\mathbf{f}), \mathbf{f}^{wt_r} \rangle}. \tag{4.22}$$

Note that here the relative gap is computed for each O-D pair and departure time, and the largest value is used to measure convergence. This way, we can ensure that equilibrium according to a given threshold is achieved for each O-D pair and departure time. It is easy to see that the gap function reaches zero when the equilibrium conditions are strictly satisfied.

It is mainly the choice of β^k that define different MSA schemes. In the simplest case [67], we just set the step size by:

$$\beta_k = \frac{1}{k}.$$

The problem with this pre-defined step size is that it does not respond to any changes in the convergence behavior at different solution stages. To address this issue, [42] proposed a self-regulated scheme that sets the step size according to

$$\beta_k = \frac{1}{\alpha_k} \text{ where } \alpha_k = \begin{cases} \alpha_{k-1} + \Gamma, & \text{If } r_k \geq r_{k-1}, \\ \\ \alpha_{k-1} + \gamma, & \text{Otherwise,} \end{cases}, \alpha_0 = 1. \tag{4.23}$$

where $\gamma \in [0.1, 0.5]$ and $\Gamma \in [1.5, 2]$ are parameters. The idea is to reduce the step size faster (hence more conservative move along the current direction) when the current iteration enlarges the relative gap, and to slow down the reduction of the step size (hence more aggressive move along the current direction) when the current iteration closes the gap. In this dissertation, we propose a self-regulated scheme similar to (4.23) but make an additional correction related to the gap computed for each O-D pair. Accordingly, our step size is O-D specific, i.e.,

$$\beta_k^{wt_r} = \frac{1 - \langle \mathbf{u}^{wt_r}(\mathbf{f}), \mathbf{f}^{wt_r} \rangle / \langle \mathbf{u}^{wt_r}(\mathbf{y}), \mathbf{f}^{wt_r} \rangle}{\alpha_k}, \tag{4.24}$$

where α_k is computed as in (4.23) with $\alpha_0 = 0.5, \Gamma = 0.018, \gamma = 0.002$ (the parameters are chosen based on numerical experiments). The second term in the numerator is the ratio between the current total profit and the maximum possible profit, which always ranges between 0 and 1. The idea is that, when the solution approaches DTE, the step size should be smaller. The ratio in the numerator does exactly that: the ratio becomes larger as the solution moves towards DTE, which reduces the step size. If the solution arrives at DTE, the step size will be reduced to zero. For convenience, Liu's scheme and the proposed scheme will be referred to as MSA by self-regulation (MSASR) and MSA by self-regulation of profit (MSASRP), respectively.

Algorithm 4.2 MSA Hyperpath Balancing Algorithm

```
Inputs: $\epsilon, \mathbf{f}$: $f_h^{wt_r}, \forall h \in \mathcal{H}, w \in W, t \in \mathcal{T}$.
Initialization: set $r$ by Equation (4.22), $k = 0$,
while $r > \epsilon$ do
    $k = k + 1$, $\alpha_k = \frac{1}{k}$
    for all $w, t_r$ do
        Set an auxiliary hyperpath flow vector $\mathbf{y}^{wt_r}(k)$ according to an all-or-nothing assignment.
        Set $\beta_k$ by Equation (4.24)
        if $\beta_k^{wt_r} \geq 1$ then $\beta_k = 0.999$
        end if
        $\mathbf{f}^{wt_r}(k+1) = \mathbf{f}^{wt_r}(k) + \beta_k^{wt_r}(\mathbf{y}^{wt_r}(k) - \mathbf{f}^{wt_r}(k))$
    end for
    Run DNL Algorithm to update link flows $\mathbf{v}$ and profits $\mathbf{u}$ with $\mathbf{f}^{k+1}$.
    Update $r$ by Equation (4.22)
end while
Outputs: $\mathbf{f}, \mathbf{v}, \mathbf{u}, \mathbf{p}$.
```

Figure 4.4. MSA Hyperpath Balancing Algorithm.

Algorithm 4.3 MSASR Hyperpath Balancing Algorithm

```
Inputs: $\epsilon, \Gamma = 1.8, \gamma = 0.2, \mathbf{f}$: $f_h^{wt_r}, \forall h \in \mathcal{H}, w \in W, t \in \mathcal{T}$.
Initialization: set $r$ by Equation (4.22), $k = 0$,
while $r > \epsilon$ do
    $k = k + 1$
    if $k = 1$ then $\alpha_k = 1.0$
    else
        if $r_k \geq r_{k-1}$ then $\alpha_k = \alpha_{k-1} + \Gamma$
        else $\alpha_k = \alpha_{k-1} + \gamma$
        end if
    end if
    for all $w, t_r$ do
        Set an auxiliary hyperpath flow vector $\mathbf{y}^{wt_r}(k)$ according to an all-or-nothing assignment.
        Set $\beta_k$ by Equation (4.24)
        if $\beta_k^{wt_r} \geq 1$ then $\beta_k = 0.999$
        end if
        $\mathbf{f}^{wt_r}(k+1) = \mathbf{f}^{wt_r}(k) + \beta_k^{wt_r}(\mathbf{y}^{wt_r}(k) - \mathbf{f}^{wt_r}(k))$
    end for
    Run DNL Algorithm to update link flows $\mathbf{v}$ and profits $\mathbf{u}$ with $\mathbf{f}^{k+1}$.
    Update $r$ by Equation (4.22)
end while
Outputs: $\mathbf{f}, \mathbf{v}, \mathbf{u}, \mathbf{p}$.
```

Figure 4.5. MSASR Hyperpath Balancing Algorithm.

It is well-known that MSA is a convergent algorithm for a broad class of traffic assignment problems [see, e.g., 61, 5]. Nonetheless, the proof of convergence usually requires that the objective function (in the case where the problem can be formulated as a mathematical program) is convex to avoid the trap of local optimal solutions. This fundamental requirement is unfortunately not met in the DTE problem, which has a non-monotone cost function in the VIP formulation. Hence, the proposed MSA algorithm must be considered as no more than a heuristic.

Algorithm 4.4 MSASRP Hyperpath Assignment Algorithm

```
**Inputs:** $\epsilon, \Gamma = 0.018, \gamma = 0.002, \mathbf{f}$: $f_h^{wt_r}, \forall h \in \mathcal{H}, w \in W, t \in \mathcal{T}$.
**Initialization**: set $r$ by Equation (4.22), $k = 0$,
while $r > \epsilon$ do $k = k + 1$
    if $k = 1$ then $\alpha_k = 0.5$
    else
        if $r_k \geq r_{k-1}$ then $\alpha_k = \alpha_{k-1} + \Gamma$
        else $\alpha_k = \alpha_{k-1} + \gamma$
        end if
    end if
    for all $w, t_r$ do
        Set an auxiliary hyperpath flow vector $\mathbf{y}^{wt_r}(k)$ according to an all-or-nothing assignment.
        Set $\beta_k$ by Equation (4.24)
        if $\beta_k^{wt_r} \geq 1$ then $\beta_k = 0.999$
        end if
        $\mathbf{f}^{wt_r}(k+1) = \mathbf{f}^{wt_r}(k) + \beta_k^{wt_r}(\mathbf{y}^{wt_r}(k) - \mathbf{f}^{wt_r}(k))$
    end for
    Run DNL Algorithm to update link flows $\mathbf{v}$ and profits $\mathbf{u}$ with $\mathbf{f}^{k+1}$.
    Update $r$ by Equation (4.22)
end while
**Outputs:** $\mathbf{f}, \mathbf{v}, \mathbf{u}, \mathbf{p}$.
```

Figure 4.6. MSASRP hyperpath assignment algorithm

4.5.1. Example Assignment

Again we use the same example in Figure 4.2 for the illustration. Recall that for the initial hyperpath flow vector ($\mathbf{f} = [8, 20, 12]$), the profits $\mathbf{u} = \{4, 3.25, 3.417\}$. Clearly, this solution does not meet the DTE conditions (4.10) and the relative gap is $r = 0.159$.

Using the current flow as a starting point, Table 4.2 reports the solutions obtained at different MSA iterations. The algorithm achieves a relative gap below 1×10^{-4} after 66 iterations. For the initial solution, the average profit per truck is about 3.65. At DTE, the average profit per truck converges toward a much less value of 2.65. In this case, the solution seems to suggest that the hyperpath h_2 should not receive any flow at DTE.

Interestingly, it is possible but difficult for MSA to locate unless searching with small step size near the solution, a different DTE exists. This other equilibrium turns out to be more profitable with $\mathbf{f} = [2.289, 27.412, 10.299]$ and $\mathbf{u} = \{3.739, 3.739, 3.739\}$. This result confirms that there are indeed multiple DTE, thanks to the non-monotonicity of the profit function.

Table 4.2. MSASRP on 1 node example, 40 trucks, $\gamma = 0.002, \Gamma = 0.018, h = 3, \epsilon = 0.0001$

# Iter.	f_1	f_2	f_3	u_1	u_2	u_3	β_k	r
0	8	20	12	4	3.25	3.417	0.275	0.159
1	16.798	14.501	8.701	3.691	1.449	4	0.509	0.358
2	8.246	7.118	24.636	4	3.655	3.187	0.270	0.163
3	16.830	5.194	17.976	4	1.972	2.380	0.461	0.330
5	10.896	1.110	27.995	4	3.395	3.031	0.311	0.210
10	30.763	0.054	9.182	2.150	-2.246	4	0.566	0.557
20	16.076	1.37×10^{-04}	23.924	4	2.445	2.405	0.319	0.313
50	28.555	1.04×10^{-08}	11.445	2.564	-2.250	2.864	0.072	0.081
66	28.235	4.38×10^{-09}	11.765	2.650	-2.25	2.650	1.37×10^{-05}	1.64×10^{-05}

4.6. DTE Solution Algorithm

We are ready to present the overall solution algorithm based on column generation, as described in Figure 4.8.

The algorithm starts by finding an optimal hyperpath for each O-D pair $w \in \mathcal{W}, t_r \in \mathcal{T}$, while assuming that there are no competitions for any link (i.e., the truck flow vector is set to zero). Then, we assign all flows from each O-D to the corresponding initial hyperpath and call the DNL procedure (see Figure 4.3). The loading process generates the profits for each hyperpath, as well as truck flows on each link. This completes the initialization process.

Entering the main loop, we first attempt to generate a new hyperpath for each O-D pair by calling the HyDOT algorithm again. This time the competitions from other trucks are estimated using truck flows obtained from the previous DNL. The generated path is added to the current hyperpath set only if (1) it is a new hyperpath, and (2) its expected profit is higher than those of all hyperpaths in the current set.

How to verify whether or not a hyperpath is new merits some explanation. Since a hyperpath encapsulates both routing and bidding decisions, comparing two hyperpaths involves checking not only all links but also the bidding order at each node. Strictly speaking, even the bidding price is a property of a hyperpath. However, treating hyperpaths with the same topology and bidding order as "different" only because of a slightly different bid price is not realistic, because it leads to an infinite number of possible hyperpaths. Thus, in this dissertation, we do not differentiate hyperpaths based on bidding price.

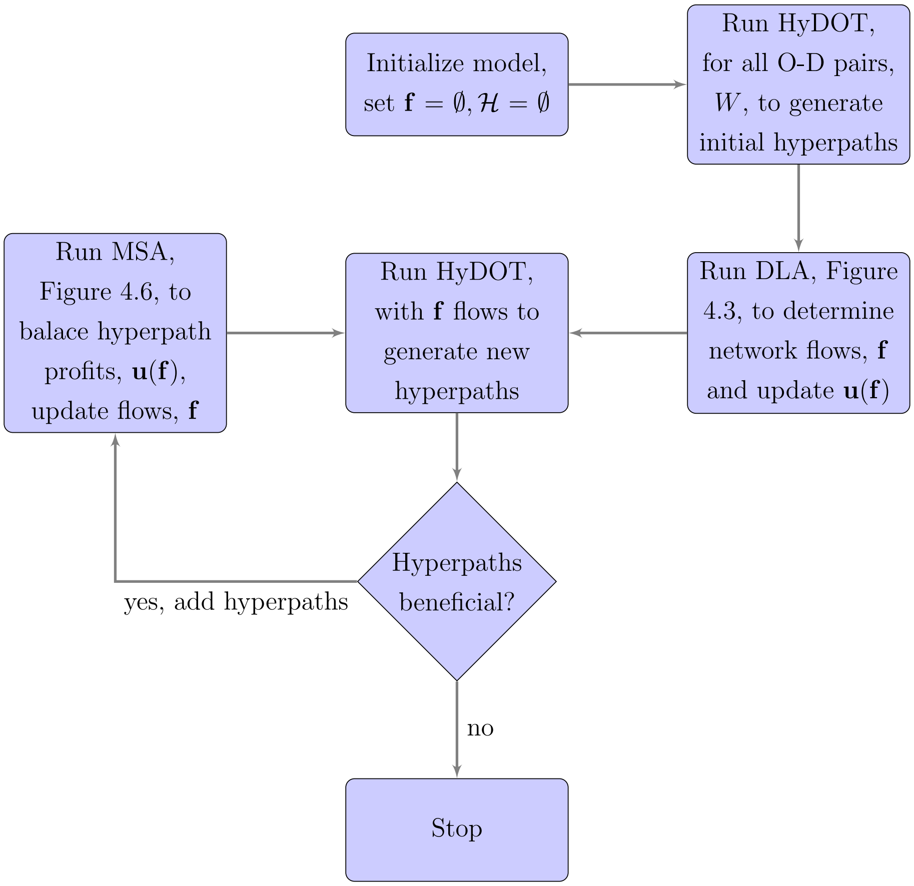

Initialize model, set $\mathbf{f} = \emptyset, \mathcal{H} = \emptyset$
Run HyDOT, for all O-D pairs, W, to generate initial hyperpaths
Run DLA, Figure 4.3, to determine network flows, $\mathbf{f}$ and update $\mathbf{u}(\mathbf{f})$
Run HyDOT, with $\mathbf{f}$ flows to generate new hyperpaths
Run MSA, Figure 4.6, to balace hyperpath profits, $\mathbf{u}(\mathbf{f})$, update flows, $\mathbf{f}$
Hyperpaths beneficial?
yes, add hyperpaths
no
Stop

Figure 4.7. DTE Algorithm

If at least one O-D pair accepts a newly generated hyperpath, the algorithm proceeds to call the MSA algorithm to equilibrate flows on the current hyperpath sets until the convergence criterion is met; otherwise, the algorithm terminates. Once the equilibrium is reached by MSA, the main loop is repeated, started with another dynamic network loading.

Algorithm 4.5 DTE Algorithm

```
Initialization
Set $\mathbf{u}, \mathbf{f}, \mathbf{v}, \mathbf{p}, \mathcal{H} = \emptyset$, NewPath = true, iteration $k = 0$
for all $w, t_r$ do
    Run HyDOT to generate the initial hyperpath $h^{wt_r}$ , the corresponding bidding orders vector $\phi_h^{wt_r}$ and the bidding price vector $\mathbf{x}_h^{wt_r}$.
    Initialize $\mathcal{H}^{wt_r} = \{h^{wt_r}\}$.
end for
Run DNL to update link flow vector $\mathbf{v}$ and profit vector $\mathbf{u}$.
while NewPath = true do
    NewPath = false
    for all $w, t_r$ do
        Set $k = k + 1$
        Run HyDOT to generate new hyperpath $\bar{h}^{wt_r}$, $\phi_{\bar{h}}^{wt_r}$, $\mathbf{x}_{\bar{h}}^{wt_r}$ based on current $\mathbf{v}$.
        if $\bar{h}^{wt_r} \neq h^{wt_r}, \forall h^{wt_r} \in \mathcal{H}^{wt_r}$ and $u_{\bar{h}}^{wt_r} > u_h^{wt_r} \forall h^{wt_r} \in \mathcal{H}^{wt_r}$ then
            Set NewPath = true and $\mathcal{H}^{wt_r} = \bar{h}^{wt_r} \bigcup \mathcal{H}^{wt_r}$
        end if
    end for
    if NewPath = true then
        Run MSA Hyperpath Assignment Algorithm to equilibrate flows for each $\mathcal{H}^{wt_r}$.
    end if
end while
Outputs: $\mathbf{f}, \mathbf{v}, \mathbf{u}, \mathbf{p}$.
```

Figure 4.8. DTE solution algorithm

4.7. Summary

This chapter explored how to model the hyperpath-based DTE problem. Also, the mathematical characteristics were explored and found to be less than ideal for finding unique solutions quickly. While the problem is shown to be non-monotone and has some areas of discontinuity, the solution was shown to exist. Also, the dynamic loading model, while driven to employ rules that favor simplicity, still scales poorly because of the possible exponential number of hyperpaths. While this problem is shown to be challenging to solve with many possible equilibrium solutions to a problem, the possibly substantial significant impact of the model demands numerical exploration.

CHAPTER 5

Data, Numerical Results, and Analysis

In this chapter, we apply the OFEX routing and DTE models to data from a real OFEX in China. We first discuss the nature of the OFEX data used to study the problem for the dissertation. We generate simple and complex numerical results for both the OFEX routing problem and the DTE problem and analyze the results. We examine the computational performance of both algorithms and compare each model's improvement against the baseline solutions.

5.1. Data Framework

The data used for testing, validating, and running numerical analysis of the problem in this dissertation proposal comes from real data collected in China. The initial data was collected from periods in December of 2015 and July 2016. The data collected includes the posted time of the shipment on the website of the third party logistics company, how long it lasted online, the origin and destination province and city of the shipment, the type of freight, the truck type used, the distance, the weight, and the price. The initial data includes information on nearly 113,000 shipments. This data allows for analyzing many aspects of the data. A few examples include the shipment arrival rate for each location, and the shipment imbalances, as seen in figure 5.1.

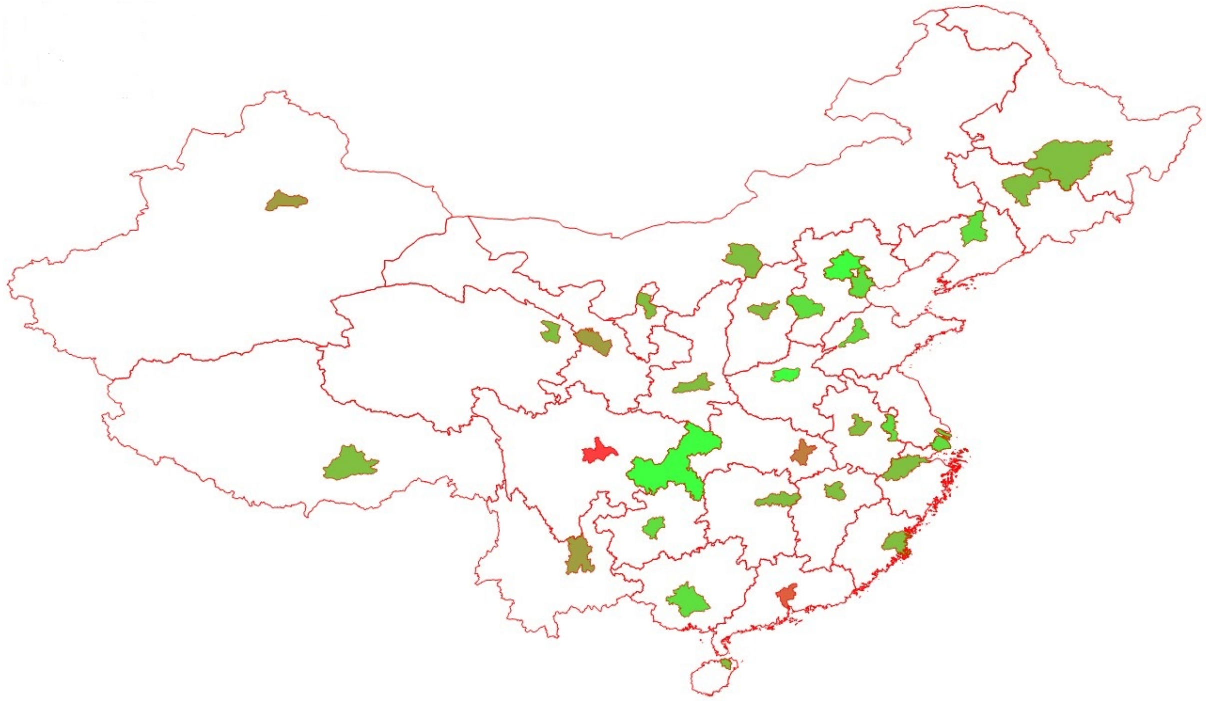


Figure 5.1. Aggregated Shipping Balance

5.2. OFEX Routing Model Numerical Experiments

The hyperpath algorithm was coded in Octave 4.2.1 [20], and all tests were performed on a personal computer with dual-core (2.4 GHz, 2.4 GHz Intel Core i7 3635QM) processors with 8 GB of memory. The computer was equipped with the 64-bit Windows 10 operating system. The algorithm is first tested on a small network to illustrate its properties and then is applied to the real OFEX data from China.

Unless otherwise specified, the following default parameter values are used throughout this section:

$$\delta = 1h, \alpha_w = ¥10/h, \alpha_e = ¥175/h, \alpha_f = ¥210/h, \gamma = ¥125/h, \bar{p}_0 = 0.9, \epsilon = 0.35.$$

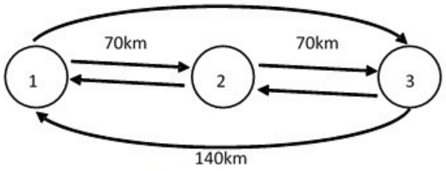


Figure 5.2. A three-city shipping network

5.2.1. Illustrative example

In this illustrative example, a three-city shipping network is considered, as shown in Figure 5.2. The distance between each city pair is given a link label in the plot. The travel speed between a given city pair is assumed to be a constant of 70 km per hour. For simplicity, the handling time is set to 0 in this example. Table 5.1 reports the loads, travel time τ_a and price range for each city pair. Note that the data in the table are the same for all time intervals.

Table 5.1. Three-city example data

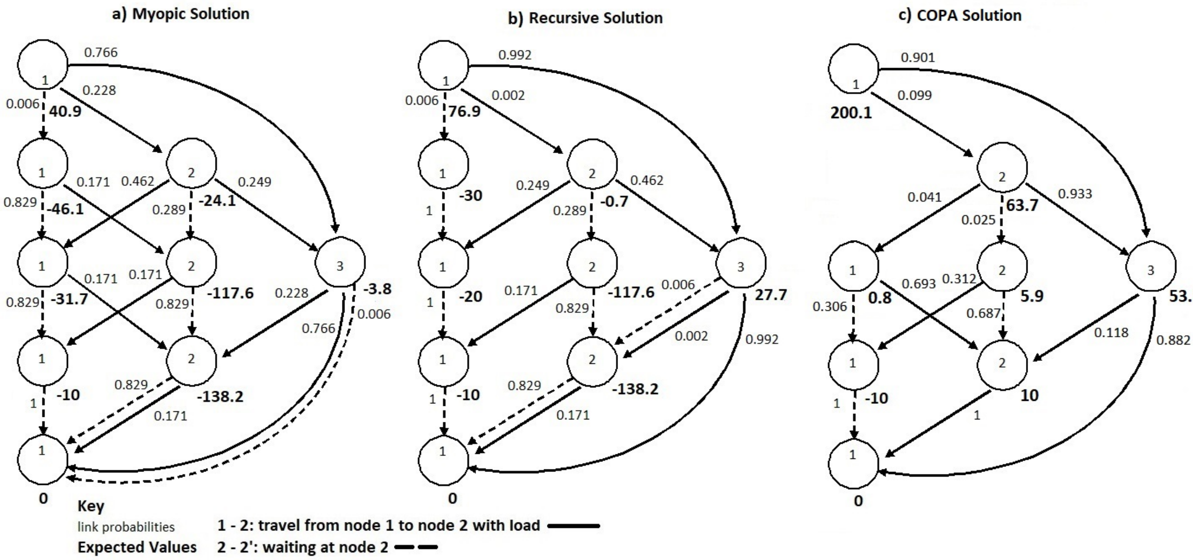
City pair	τ_a (hr)	g_a	l_a	u_a
1 - 2	1	1	200	320
1 - 3	2	1	340	600
2 - 1	1	1	220	280
2 - 3	1	1	220	280
3 - 1	2	1	420	480
3 - 2	1	1	230	250

Two other trucks are assumed to be competing for loads at every node for all times, and another truck is assumed to arrive empty at each node every two periods. This setting allows for the example to have meaningful competitions. This setting gives $H_i^t = 2.5\ \forall i \in \mathcal{L}, t = 0, \cdots, T$.

The COPA-HyDOT algorithm solves the routing problem recursively, starting at the boundary stage $t = 4$, where $V_1^4 = 0$. For stage $t = 3$ at node 1, STEN shows that the only feasible decision is to stay and wait at node 1, which makes $V_1^3 = -10 + (V_1^4 = 0) = -10$ (cf. Figure 5.3-(c)). For node 2 at $t = 3$, two options exist: a loaded move to node 1 (the feasible option) and an empty move to node 1 (the fallback option). Since there is only one feasible option, all available trucks will compete for it, so $H_2^3 = 2.5$ and $b_{21}^3 = 3.5$ (includes the the routing truck). With $b_{21}^3 = 3.5$ and $g_{21}^3 = 1$, $p_a^0 = 0.171$ as given by Equation (3.9). Solving the optimal bidding problem sets $x_{21}^{3*} = 220$, which is the lowest bid in the price range (since the recourse option is a significant loss) and sets $p_{21}^3 = 1$. This results in $V_2^3 = 1 \times 10 + 0 \times -175 + V_1^4 = 10$. No bidding order is concerned until there is more than one option available, which first occurs at $t = 2$ at node 3, where the more profitable order is to bid for (3,1) first and then bid for (3,2). The expected profits are 53.5 with this order, or 42.0 if a truck bids for (3,2) first.

Completing the COPA-HyDOT algorithm in this fashion yields a hyperpath shown in Figure 5.3-(c), where each link on the hyperpath is labeled with the probability of being chosen by a truck at its tail in the STEN. The final expected profit $V_1^1 =$ ¥200.1. Figure 5.3-(d) shows that the hyperpath obtained by the algorithm contains eight simple paths (each has a non-zero choice probability and profit).

The hyperpaths obtained by the benchmark BAPA-HyDOT algorithm are reported in Figure 5.3-(a) and (b), where (a) corresponds to the `Myopic` model and (b) the `Recursive` model. The `Myopic` and `Recursive` models generate the expected value of 40.9 and 76.9, respectively. As expected, because the `Recursive` model allows the truck to anticipate the future profits, it helps to improve the overall profits (in this case by nearly 100%). For example, with the anticipatory information, a truck could avoid taking a load that could



d) Enumerated Hyperpaths

	Myopic		Recursive		COPA			Myopic		Recursive		COPA	
Hyperpath	Profit	Probability	Profit	Probability	Profit	Probability	Hyperpath	Profit	Probability	Profit	Probability	Profit	Probability
1 - 1' - 1' - 1' - 1'	-40	0.0041	-40	0.006	-	-	1 - 2 - 3 - 1	120	0.0435	120	0.0008	60.1	0.0815
1 - 1' - 1' - 2 - 1	70	0.0001	-	-	-	-	1 - 2 - 3 - 1'	-260	0.0003	-	-	-	-
1 - 1' - 1' - 2 - 1'	-145	0.0007	-	-	-	-	1 - 2 - 3 - 2 - 1	160	0.0022	160	3E-07	33.5	0.0109
1 - 1' - 2 - 1 - 1'	70	0.0002	-	-	-	-	1 - 2 - 3 - 2 - 1'	-55	0.0107	-55	1E-06	-	-
1 - 1' - 2 - 2' - 1	70	0.0001	-	-	-	-	1 - 2 - 3 - 2' - 1	-	-	-45	9E-07	-	-
1 - 1' - 2 - 2' - 1'	-145	0.0007	-	-	-	-	1 - 2 - 3 - 2' - 1'	-	-	-260	4E-06	-	-
1 - 2 - 1 - 1' - 1'	70	0.0873	70	0.0004	0.7	0.0013	1 - 3 - 1	80	0.5868	80	0.9845	219.4	0.7947
1 - 2 - 1 - 2 - 1	180	0.0031	-	-	30.7	0.0028	1 - 3 - 1'	-300	0.0046	-	-	-	-
1 - 2 - 1 - 2 - 1'	-35	0.0149	-	-	-	-	1 - 3 - 2 - 1	120	0.0299	120	0.0003	192.7	0.1063
1 - 2 - 2' - 1 - 1'	70	0.0113	70	0.0001	-1.2	0.0008	1 - 3 - 2 - 1'	-95	0.1448	-95	0.0015	-	-
1 - 2 - 2' - 2' - 1	70	0.0094	70	0.0001	-20	0.0017	1 - 3 - 2' - 1	-	-	-85	0.001	-	-
1 - 2 - 2' - 2' - 1'	-145	0.0453	-145	0.0004	-	-	1 - 3 - 2' - 1'	-	-	-300	0.0049	-	-

Figure 5.3. COPA, BAPA-Myopic and BAPA-recursive models results for three-city example

result in a higher chance of future loss than simply waiting or moving empty to another location.

Compared to the `Recursive` model, COPA further increases the expected profit by over 150%. Among other things, optimal pricing helps (1) reduce the probability of being forced to move empty, and (2) increases profits by setting the price higher than the expected value, when the probability of finding a load is high.

Figure 5.3-(d) shows that the hyperpaths corresponding to the `Myopic` and `Recursive` models contain 20 and 15 simple paths, respectively, compared to 8 for the COPA model. This shows that both the forward-looking strategy and the optimal pricing help eliminate suboptimal choices that should not be included in the hyperpath. The finding is consistent with the previous observation that COPA yields the highest expected profit, followed by the `Recursive` model.

5.2.2. Real-world OFEX Data

As mentioned in the previous section, the OFEX data set used in this dissertation consists of detailed information about freight loads between 31 major cities in China within four weeks (two weeks in December of 2015, and another two in July of 2016). Because the loads are posted online continuously in the raw data, they first have to be aggregated into "operating intervals." Figure 5.4 plots the distribution of the number of loads posted by the time of day. It shows that the number of loads posted on the platform begins to rise sharply around 6:00 AM, peaks around 9:30 AM and then quickly declines. Overall, most

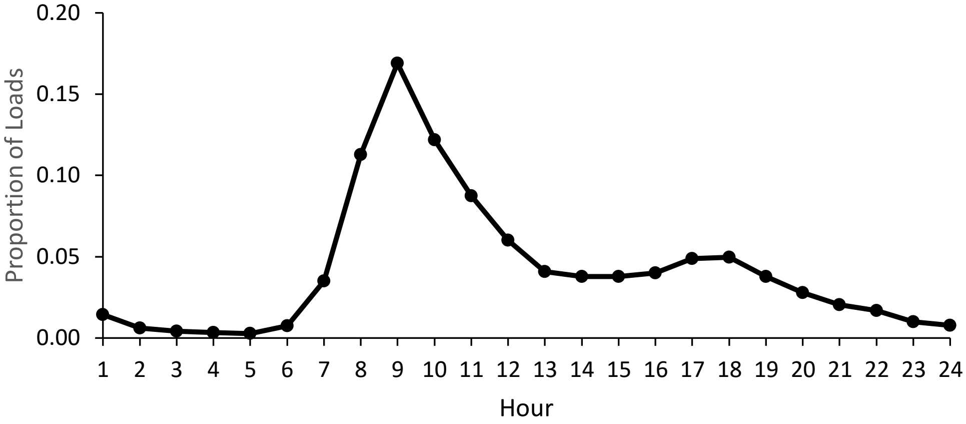


Figure 5.4. Hour of day load distribution

loads were posted in the morning hours, with a very small percentage appearing between 6:00 PM and 6:00 AM.

The length of the operating interval (δ) is selected as one hour in this dissertation. That is to say, a truck can make a decision every hour within the 12 working hours (from 6:00 AM to 6:00 PM) in a day. The constant travel speed between cities is estimated as 70 km/h, and the inter-city travel time is calculated based on the distance and the speed, and then rounded up to the next hour.

A close look at the data shows that hourly loads fluctuate significantly and that a posted load stays on the platform for about six hours on average. In light of these observations, we estimate for each city pair the hourly loads at hour t by taking an average of loads recorded in hours $t-5, \cdots, t-1, t$. The only exception is hour 1 (6:00 AM), for which averaging goes back to 6:00 PM the previous day. This hour is treated differently because there are little activities overnight, as shown in Figure 5.4. In the same manner, the ranges of prices of the loads are aggregated.

Using the above method, a dataset is created to represent freight activities within the 31-city network in an "average week" (by taking the average over the four weeks covered by the raw data). This results in a STEN that has 84 (12×7) time intervals and 158,844 ($(31 \times 2 - 1) \times 31 \times 84$) links. If a tour schedule longer than a week is needed for a test, the dataset is simply replicated as needed to represent as many weeks as required.

5.2.3. Experiment Results Based on OFEX data

In this experiment, each run is specified by start and end locations and times. The locations can be the same as in a selective TSP, or different as in the orienteering problem.

This allows for maximum flexibility, especially for a long planning horizon where the truck must stop to rest for more than just an overnight to follow trucking regulations. To use this model to maximize a specified time horizon, the model could be run repeatedly over all potential end locations to find the optimal expected value ending point. It is also possible to run the model repeatedly over a set of possible ending times to find the optimal time to end a tour.

5.2.3.1. Bidding Order Impact. The first set of experiments was designed to examine how different bidding orders (see Section 3.2.3) may affect the solutions obtained by the COPA-HyDOT algorithm.

Table 5.2 reports the average profit generated from all tours starting and ending at the capital city of Hubei province, chosen for being centrally located within China. The first column reports the test ID, the second shows the number of time intervals used in the test, columned 3 - 10 report the average profits when $\bar{\Phi}$ includes only one of the eight bidding orders; and column 11 shows the average profit when $\bar{\Phi}$ includes all eight bidding orders. Clearly, the last column gives the highest expected profit because it compares all eight orders at each node. The results show that overall the bidding order seems to have a negligible impact on the expected profits. The worst order, $\bar{\phi}_0$', is still able to yield 99.56% of the maximum average profit achieved by the strategy that considers all eight orders.

Looking closer at the data shows that for the hyperpaths generated, the number of feasible options under consideration is often quite small (sometimes only one), making the order less relevant. In many other cases, where as many as ten options are considered, the best load was priced to achieve 100% winning probability, which again wipes out the

Table 5.2. Impact of bidding orders on average profits from tours based at the capital city of Hubei Province

Test	T (hrs)	$\bar{\phi}_0$ (¥)	$\bar{\phi}'_0$ (¥)	$\bar{\phi}_1$ (¥)	$\bar{\phi}'_1$ (¥)	$\bar{\phi}_2$ (¥)	$\bar{\phi}'_2$ (¥)	$\bar{\phi}_3$ (¥)	$\bar{\phi}'_3$ (¥)	$\bar{\Phi}$ (¥)
1	50	2343.1	2337.0	2336.9	2343.3	2342.3	2337.8	2342.2	2337.7	2344.2
2	100	6908.0	6884.8	6883.0	6908.8	6885.5	6882.4	6916.6	6875.4	6920.4
3	150	12291	12230	12244	12285	12285	12243	12289	12237	12302
4	200	17354	17295	17297	17350	17354	17295	17346	17305	17354
5	250	22367	22309	22316	22369	22367	22313	22380	22309	22407
Rank (% of $\bar{\Phi}$)		2(99.90)	8(99.56)	5(99.59)	3(99.89)	4(99.82)	6(99.59)	1(99.92)	7(99.56)	-

Chinese Yuan (¥) $\approx$ 0.15 US Dollar ($)

effect of bidding order because other bids could not change the overall expected profit. There were cases when the bidding order did make a significant impact. In those cases, we found that the magnitude of the impacts varies by location and time, and no order seems to have a clear advantage over others concerning profit maximization. Because the overall impact of bidding orders is quite small, the following experiments choose only to use $\bar{\phi}_3$ to reduce the computational overhead. Table 5.2 shows that this order yields the highest average expected profits among the eight, although the difference is small.

5.2.3.2. Comparison with Benchmark Models. In this section, the performance of the COPA model is compared with the `Myopic`-BAPA and `Recursive`-BAPA benchmark models.

Table 5.3 shows the comparison results for two base cities: the capital cities of Anhui and Hubei provinces, respectively. The capital city of Anhui is located in eastern China, and according to the data collected, is a sink node, i.e., more loads enter the city than leave. In contrast, the capital city of Hubei, located in the center of the country, is a source node (more loads leave than enter) and has much higher load activities than Anhui. These two cities were selected to examine how such localities might affect the performance of the three models.

Table 5.3. Average profit and computing time for tours based at capitals of Anhui and Hubei

Anhui (Less Connected City)					
Test	T (hr)	`Myopic`-BAPA (¥)	`Recursive`-BAPA (¥)	COPA (¥)	COPA runtime (s)
1	40	-838.0	-400	1699.3	3.24
2	50	-1219.4	-426.4	2269.9	5.37
3	60	-1295.0	-174.8	3005.8	7.83
4	70	-1413.5	353.5	3684.0	10.50
5	80	-1396.6	1106.4	5104.2	12.54
Hubei (More Connected City)					
1	40	-907.0	-273.7	1879.2	3.44
2	50	-645.6	262.6	2191.4	5.62
3	60	-1063.2	683.9	2808.7	8.25
4	70	-1397.8	1882.4	3858.9	10.77
5	80	-1203.5	2303.2	4702.8	13.62

For both cities, the results show that COPA could improve profitability dramatically, compared to the two benchmark cases. In many instances, COPA increases the profit generated by `Recursive`-BAPA by 200% to 400%, and in some instances the profit changes from negative to a positive profit. The improvement mostly comes from nearly 200% increase in revenue, and a 100% cost savings on average in these instances. Notably, this finding is consistent with the observation by [38]. Also worth noting is that accurately anticipating future profits alone could generate substantial benefits profit. In most cases, `Recursive`-BAPA delivers significantly better results than `Myopic`-BAPA, often turning a considerable loss into a positive profit. In a nutshell, the above findings confirm the value of both optimal bidding and forward-looking strategies.

Comparing the results from the two cities yields somewhat surprising findings. The tours based at the capital city of Hubei (the centrally located source city) are only doing better than those based at the capital city of Anhui (a less connected sink city with much lower load activities) in the benchmark cases. When COPA is applied, the average profits of the tours based at Anhui outperform those associated with Hubei by a nontrivial margin. This result indicates that COPA is much less sensitive to the base city of a truck than the benchmark cases.

The last column in Table 5.3 reports the time required by the COPA model to solve each test case. Even with a MATLAB script, the problem can be solved within seconds in most cases. The case with the most extended time horizon took less than 14 seconds. The algorithm also scales well as the number of time intervals increase. Thus, we expect that applying the proposed algorithm to solve real-world problems is practical.

5.2.3.3. Sensitivity Analysis. In this section, experiments are designed to test the sensitivity of the COPA routing results to the starting time, the base city and the average ratio of empty truck flow (ϵ defined in Section 3.4.1).

Table 5.4 shows the expected profits of 60-hour work tours based on all 31 cities, with all tours starting and ending in the same city. The best 60-hour tour originates and ends in the capital city of Heilongjiang, while the minimum is based at the neighbor province of Jilin. While the location does affect profits, the standard deviation related to the choice of the base city is only ¥267, less than 10% of the average profits. Again, this suggests that COPA reduces the importance of the base city because it is able to *take advantage of network-wide visibility.*

While the starting location and proximity to other locations may influence the results, the departure time can also play a role. Figure 5.5 shows how a 60-hour tour's profit changes depending on the departure time. The plot reveals a periodic pattern for this data set, with a departure in the middle of the working days generally producing higher expected profits than starting early or late in the working day. This result seems correlated

Table 5.4. Expected profits for tours based at all cities ($T = 60$)

City	COPA profit (¥)	City	COPA profit (¥)	City	COPA profit (¥)
Anhui	3005.8	He'nan	3078.2	Shaanxi	2976.3
Beijing	3148.3	Hubei	2808.7	Shandong	3504.3
Chongqing	3030.4	Hunan	3271.5	Shanghai	2700.8
Fujian	2934.1	Jiangsu	2919.7	Shanxi	2904.4
Gansu	3215.1	Jiangxi	3162.0	Sichuan	3066.2
Guangdong	3030.4	Jilin	2449.3	Tianjin	2812.4
Guangxi	3066.9	Liaoning	3502.3	Xinjiang	2613.9
Guizhou	2873.4	Neimenggu	2882.3	Xizang	3247.9
Hainan	2819.7	Ningxia	3185.0	Yunnan	3035.3
Hebei	3255.0	Qinghai	3489.8	Zhejiang	3424.0
Heilongjiang	3537.7	Mean	3062.9	St. Dev.	266.8

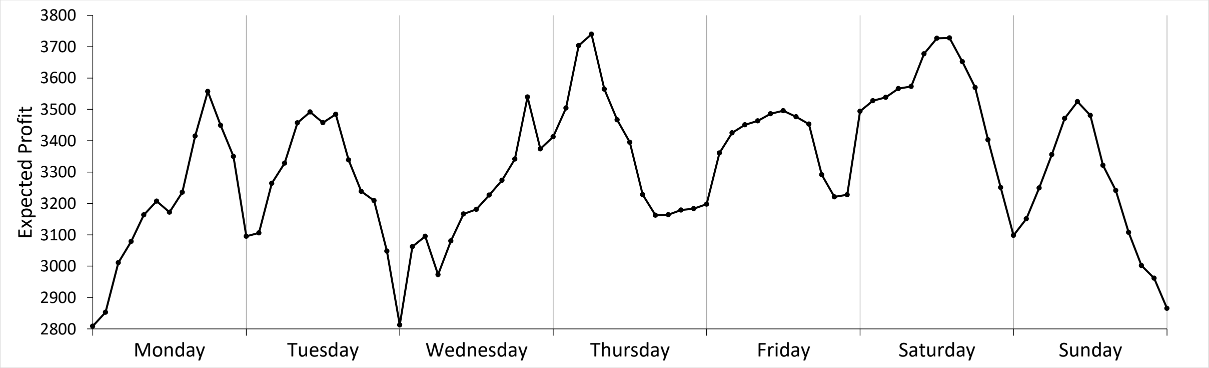


Figure 5.5. Expected profits at all possible departure times in a week ($T = 60$, base city = Hubei)

well with the fact that the morning hours have the highest load activities (see Figure 5.4). The exception to the pattern happens between late Tuesday to late Wednesday. In fact, the minimum expected profit is generated by leaving at 6:00 PM on Tuesday (or 6:00 AM on Wednesday, the same time point in the model). This result may be explained by the fact that departing at this hour means that the truck has to look for loads during the entire weekend and must return to the base at the end of Sunday when load activities are significantly decreased.

We finally test the sensitivity to ϵ, the ratio of empty truck flow. Increasing ϵ is expected to negatively affect profitability because it means more empty trucks become available at each market, hence greater competition. Figure 5.6 shows how the value of ϵ changes the average expected profit for 80-hour tours for tours from all locations ending in the capital city of Hubei. The range of ϵ values tested in the experiments is selected such that $\rho_1 \geq 0$. The reason for imposing this requirement is that $\rho_1 \leq 0$ typically creates too few trucks available to move loads.

As anticipated, the expected profits decrease as ϵ increases, in all three routing models. However, the magnitude of the impact varies. The COPA model is the least sensitive. When ϵ doubles, the average profit drops by merely 10% or ¥495. In contrast, the profits decrease by nearly 50% for the `Recursive` model and the losses incurred for the `Myopic` hit 237%, over four times worse. The `Recursive` model has the highest absolute negative change, as the difference in the range of ϵ is ¥1411, versus ¥1167 for the `Myopic`. This finding shows that the optimal bidding strategies adopted in the COPA model are better equipped to cope with the variable competitions in the market.

5.3. DTE Model Numerical Experiments

The DTE algorithm was coded in Octave 4.2.1 [20], and all tests were performed on a personal computer with a 2.0 GHz, AMD Ryzen5 2500U processor with 8 GB of memory. The computer was equipped with the 64-bit Windows 10 operating system. We first examine the properties of the DTE problem and the performance of the proposed algorithm using an illustrative example. We then apply the algorithm to a larger example created using real data collected by a Chinese OFEX platform.

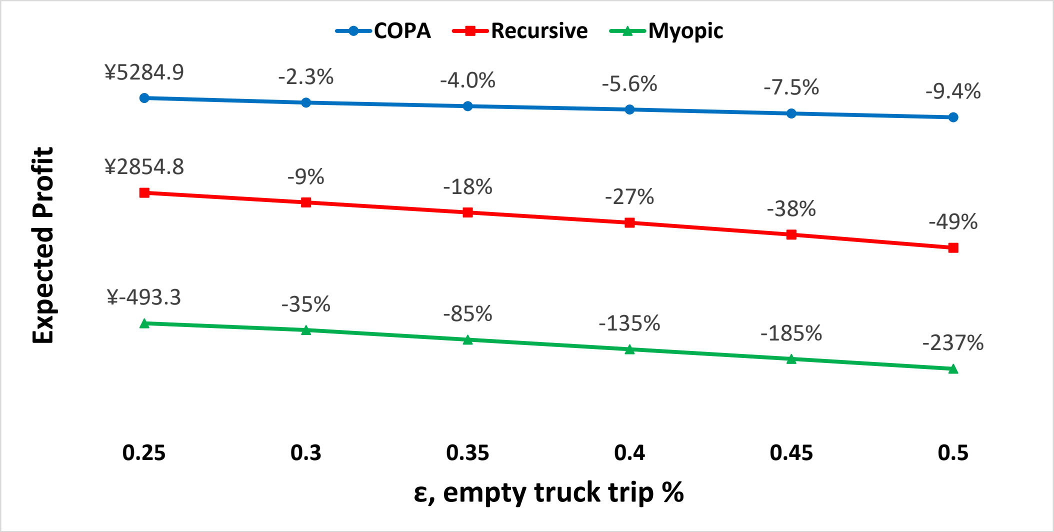


Figure 5.6. Average expected profit for $\epsilon = 0.2$ to 0.5 ($T = 80$, base city = Hubei)

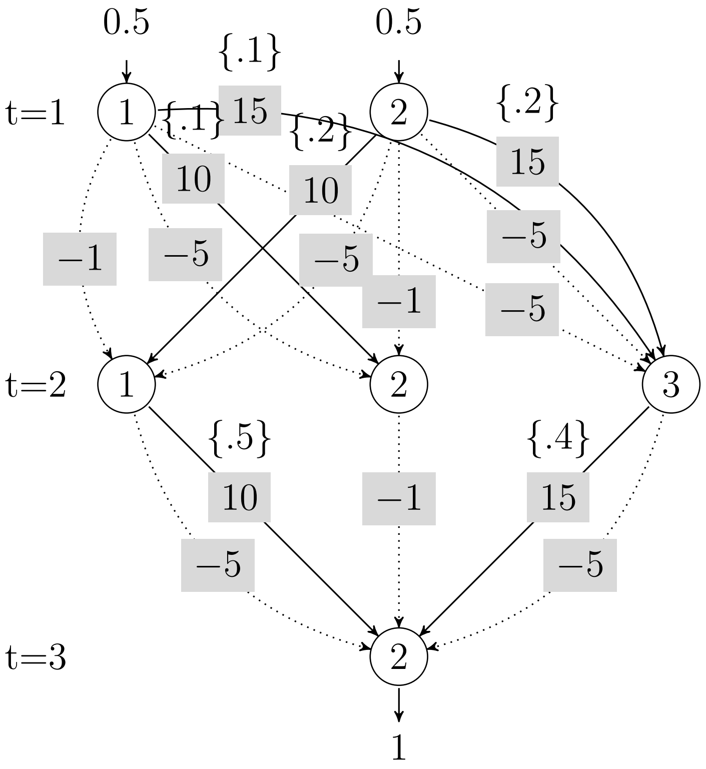


Figure 5.7. An illustrative example network

5.3.1. Illustrative example without bidding

The topology of the STEN is shown in Figure 5.7. The number of available loads, g_a^t, is placed in $\{\}$. For example, there are 0.2 units loads going from city 2 to 3 starting at $t = 1$. We prohibit bidding in this example to simplify the problem further. Accordingly, the profit $\sigma_a^t = x_a^t - c_a^t$ is assumed to be the same for each load. These are shaded in gray. The fixed profit is 15 from city 2 to 3 starting at $t = 1$. The dotted lines in the plot represent links in $\mathcal{A}_e$.

There are two O-D pairs for this problem: 0.5 units of trucks start at node 1, and the other half start at node 2, and all trucks plan to end at node 2 at $t = 3$. We simply identify the O-D pair starting at city 1 and 2 as O-D pair 1 and 2, respectively. From inspection, trucks are better off by going through city 3 because of the higher profits of connecting loads. However, there are not many loads available to or from city 3: $g_{13}^1 = 0.1, g_{23}^1 = 0.2$, and g_{32}^2 is limited to 0.4. Thus, trucks will first compete to fill the most profitable jobs, and those that are left behind will choose routes that maximize future profits.

To solve this problem, we first generate the initial hyperpaths. For O-D pair 1, $h_1^1(\phi) =$ ([3,3'],[2,2']), where the first and second brackets [] present the choice order at $t = 1$ and 2, respectively. In the first bracket, for example, the first number 3 represents moving to city 3 loaded whereas 3' means moving empty to city 3. Similarly, $h_1^2(\phi) =$ ([3,1,3'],[2,2']) for O-D pair 2. Then the trucks are loaded according to the DNL procedure. Once the loading is completed, the profits for each hyperpath can be evaluated as $u_1^1 = 4, u_1^2 = 16$. This gives us the initial solution.

We then attempt to generate for each O-D pair a second hyperpath based on the loading results. The second hyperpath generated for O-D pair 1 is h_2^1 =([3,2,1'],[2,2']),

Table 5.5. Truck flows for equilibrium solution

w_1 Arc x_{ah}^{tw}	$x_{13,1}^{1,1}$	$x_{13',1}^{1,1}$	$x_{32,1}^{2,1}$	$x_{32',1}^{2,1}$	$x_{11',2}^{1,1}$	$x_{12,2}^{1,1}$	$x_{13,2}^{1,1}$	$x_{12,2}^{2,1}$	$x_{22',2}^{2,1}$	$x_{32,2}^{2,1}$	$x_{32',2}^{2,1}$
w_1 Flow	0.005	0.021	0.025	0.001	0.279	0.095	0.1	0.279	0.1	0.09	0.005
w_2 Arc x_{ah}^{tw}	$x_{21,1}^{1,2}$	$x_{23,1}^{1,2}$	$x_{23',1}^{1,2}$	$x_{12,1}^{2,2}$	$x_{32,1}^{2,2}$	$x_{32',1}^{2,2}$					
w_2 Flow	0.2	0.2	0.1	0.2	0.285	0.015					
Node^t	1^1	2^1	3^1	1^2	2^2	3^2	1^3	2^3	3^3		
Flow	0.5	0.5	0	0.479	0.1	0.421	0	1	0		

which has an expected profit of 11.2, higher than 4. The new strategy suggests that it is now more profitable to go to city 2 at $t = 1$ from node 1 than to 3', due to overcrowding at city 3. The second hyperpath generated for O-D pair 2, however, is not new, so the hyperpath set for O-D pair 2 remains unchanged. With two active hyperpaths for O-D pair 1, the DTE conditions dictate that $u_1^1 = u_2^1$ if both hyperpaths are utilized. For this simple example, it is possible to equilibrate flows on these two hyperpaths analytically. Note that the profits on the two hyperpaths can be expressed in closed-form functions of hyperpath flow as follows:

$$\text{(5.1a)} \quad u_1^1 = \frac{15f_1^1(\frac{0.1}{0.5}) - 5f_1^1(1 - \frac{0.1}{0.5}) + 15f_1^1(\frac{0.4}{0.3+f_1^1+\frac{0.1f_2^1}{0.5}}) - 5f_1^1(1 - \frac{0.4}{0.3+f_1^1+\frac{0.1f_2^1}{0.5}})}{f_1^1}$$

$$\text{(5.1b)} \quad u_2^1 = \frac{15f_2^1(\frac{0.1}{0.5}) + 9f_2^1(1 - \frac{0.1}{0.5}) + 15\frac{0.1f_2^1}{0.5}(\frac{0.4}{0.3+f_1^1+\frac{0.1f_2^1}{0.5}}) - 5\frac{0.1f_2^1}{0.5}(1 - \frac{0.4}{0.3+f_1^1+\frac{0.1f_2^1}{0.5}})}{f_2^1}$$

$$\text{(5.1c)} \quad f_1^1 + f_2^1 = 0.5$$

Setting $u_1^1 = u_2^1$ and recalling the flow conservation conditions $f_1^1 + f_2^1 = 0.5$, we found at DTE, $f_1^1 = 0.0263$ and $f_2^1 = 0.4737$, which produces the equilibrium profit of $u_1^1 = u_2^1 = 13$. The solution also produces a profit $u_1^2 = 21.4$ for h_1^2. This turns out to be the final solution because running the HyDOT algorithm again does not generate any new hyperpaths.

A DTE solution not only tells how truck flows might distribute in the network, but it also enables an analyst to discover possible strategies for improving network efficiency. For example, the DTE solution obtained above indicates that 0.021 loads are unserved at node 1 at $t = 2$. A possible remedy to this problem is for the OFEX platform to apply a simple zero-sum price adjustment, similar to an optimal toll in the traffic assignment problem. Excess demand needs to shift away from city 3, toward city 1. Therefore, the platform can give a bonus of 0.5 for loads from city 1 to 2 at $t = 2$, which increases the profit to 10.5. Adding this amount to all 0.5 loads results in a total increase of 0.25. Then, the platform reduces the price from city 3 to 2 at $t = 2$, by 0.625, to 14.375, which over 0.4 loads results in an equal total adjustment of 0.25. Applying the new prices to the network enables h_2^1 to be the dominant hyperpath, i.e., all trucks will choose h_2^1, and the resulting flow serves all loads.

In the above adjustment, the cost to the shipper remains unchanged, but the amount that the trucks receive changed - it, of course, implies that the platform must be making the transactions so that they can manipulate the prices, in this case for the benefits of the system. This example highlights the potential of the equilibrium analysis for guiding the operations of OFEX platforms.

5.3.2. Example with Bidding

In the previous example, bidding decisions were excluded from making the assignment problem simple enough for an analytical solution. The example also avoids carrying loads undelivered in the current time interval to later intervals, and limits interdependencies

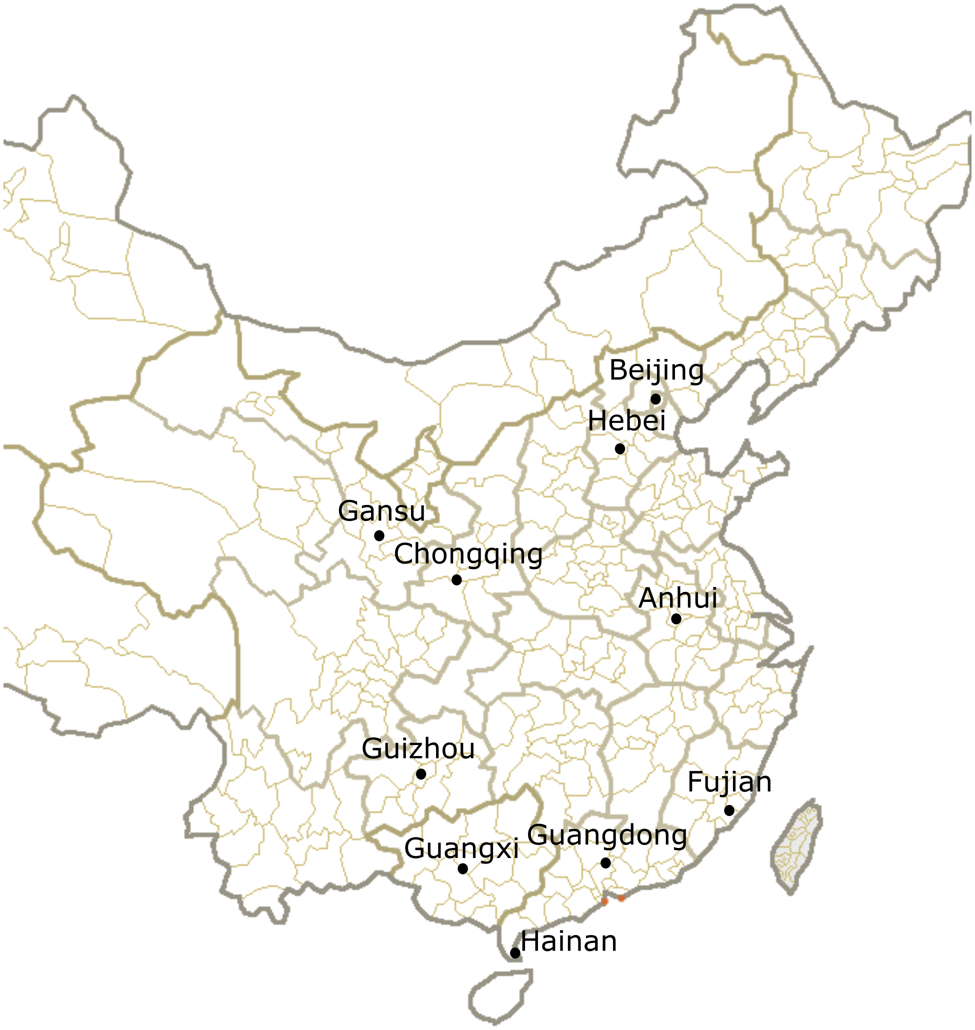


Figure 5.8. 10-city China network

between hyperpaths to only one O-D pair. We shall add these complexities back now, using a network constructed from a real OFEX data set collected in China.

Our goal here is to test the performance of the three MSA schemes: the simplest scheme (MSA), the self-regulated scheme (MSASR) and the proposed self-regulated scheme considering profit (MSASRP). The example network consists of 10 cites, as shown in Figure 5.8 and the planning horizon has 50 periods. The network creates 190 links that include loaded and empty moves for each period. In this example, we only consider one O-D pair that has 30 trucks, starting and ending at the capital city of Anhui, beginning at $t = 0$, ending at $t = 50$.

Table 5.6. MSA on one O-D pair 25 truck example, $h = 4, \epsilon = 0.001$

# Iter.	f_1	f_2	f_3	f_4	u_1	u_2	u_3	u_4	r
0	6.25	6.25	6.25	6.25	-500	1436.5	2143.3	1762.2	0.7706
1	0.0625	0.0625	24.8125	0.0625	-500	-1491	1152.4	2616.1	1.284
2	0.03125	0.03125	12.40625	12.53125	-500	-1459.74	1190.68	1075.3	0.0561
3	0.02083	0.02083	16.60417	8.35417	-500	-1461.54	995.65	1416.24	0.2501
5	0.0125	0.0125	14.9625	10.0125	-500	-1458.96	1060.57	1243.16	0.0985
10	0.00625	0.00625	12.48125	12.50625	-500	-1459.92	1176.87	1080.51	0.0437
41	0.00152	0.00152	13.41006	11.58689	-500	-1457.91	1130.99	1130.48	0.0004

Table 5.7. MSASR on one O-D pair 25 truck example, $h = 4, \epsilon = 0.001$

# Iter.	f_1	f_2	f_3	f_4	u_1	u_2	u_3	u_4	r
0	6.25	6.25	6.25	6.25	-500	1436.5	2143.3	1762.2	0.7706
1	0.0625	0.0625	24.8125	0.0625	-500	-1491	1152.4	2616.1	1.284
2	0.04018	0.04018	15.95089	8.96875	-500	-1460.21	1023.92	1344.15	0.187
3	0.02679	0.02679	10.63393	14.3125	-500	-1461.04	1326.44	971.54	0.1861
5	0.013	0.013	10.67529	14.29871	-500	-1461.12	1320.81	970.86	0.1811
10	0.00568	0.00568	12.95165	12.03699	-500	-1457.80	1155.52	1104.82	0.0225
21	0.00249	0.00249	13.40566	11.58937	-500	-1457.90	1131.39	1130.36	0.0008

Table 5.8. MSASRP on one O-D pair 25 truck example, $h = 4, \epsilon = 0.001$

# Iter.	f_1	f_2	f_3	f_4	u_1	u_2	u_3	u_4	r
0	6.25	6.25	6.25	6.25	-500	1436.5	2143.3	1762.2	0.7706
1	0.00236	0.00236	24.99291	0.00236	-500	-1491.79	1145.84	2604.48	1.2735
2	0.00152	0.00152	16.06905	8.92791	-500	-1460.64	1013.36	1347.74	0.19
3	0.00101	0.00101	10.71461	14.28337	-500	-1461.18	1315.51	970.53	0.1764
5	0.00049	0.00049	10.71495	14.28407	-500	-1461.18	1315.4	970.46	0.1763
10	0.00021	0.00021	12.96857	12.03100	-500	-1457.85	1153.60	1104.98	0.0207
21	0.00009	0.00009	13.41297	11.58684	-500	-1457.92	1130.59	1130.44	0.000076

Table 5.9. Iterations to reach gap value of 10^{-k}, one O-D pair example, $h = 4$

Method	10^1	10^0	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}
MSA	0	0	2	13	41	164	2674
MSASR	0	0	6	6	21	105	953
MSASRP	0	0	6	6	21	21	871

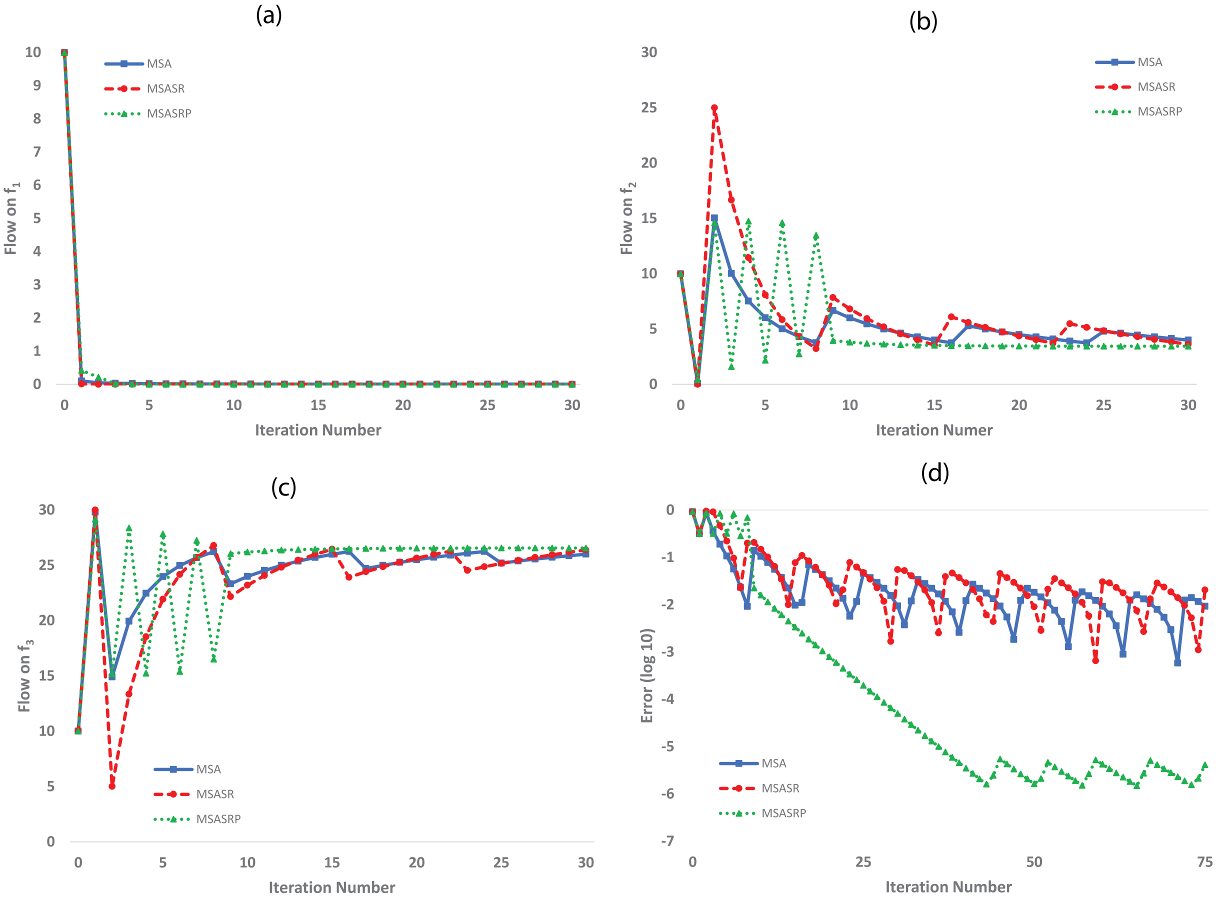


Figure 5.9. Convergence comparison for one O-D pair example, $h = 3$

In this example, the DTE algorithm performs three main iterations, and a new hyperpath is added in each iteration. When there are only 2 hyperpaths, all three MSA schemes converge quickly, and there is little difference in their overall performance. Hence, we only show the results for the last round equilibration, when there are three hyperpaths. For illustration, we start from an initial solution that assigns an equal amount of flows to each hyperpath.

Figure 5.9 compares the convergence behavior of the three MSA algorithm variants. Figure 5.9(a-c) shows how the flows on the three hyperpaths converge to their DTE value and Figure 5.9(d) reports the change of the relative gap as the iteration proceeds. It is evident from the plots that the proposed algorithm is able to converge to DTE much faster than the other two methods. Figure 5.9(d) indicates that MSASRP can reduce the relative gap down to the level of 10^{-6}, whereas the other two algorithms begin to experience difficulty around $r = 10^{-3}$. Figure 5.9(b-c) clearly shows that the flows on the hyperpaths 2 and 3 stabilize at the equilibrium value much earlier in MSASRP than in the other two.

It is worth noting that, in the early stage, the MSASRP algorithm seems to converge slower, possibly because it retains a relatively larger step size at the beginning. However, it quickly surpassed the two competitors, and eventually achieved a more precise solution. The performance of the algorithm depends on three parameters: α_0, γ, and Γ. The optimal values of these parameters may be problem specific and finding them usually requires careful tweaks.

5.3.3. Large Example with Multiple O-D pairs

In this section, we continue to use variants of the ten-city example to test the DTE algorithm. We first examine how sensitive the performance of the algorithm is to the number of O-D pairs.

Table 5.10 reports the results of the DTE when the number of O-D pair increases from 1 to 5. The second column shows the number of total trucks in the network. The third column shows how many hyperpaths were added to each O-D pair. The fourth column shows how many of the hyperpaths in the final iteration had significant positive flow at DTE. As the number of O-D pairs increases, the level of effort for the solution algorithm increases exponentially. When there are more than two O-D pairs, converging to a reasonably precise solution becomes very difficult with the current implementation of the algorithm. When there are five O-D pairs, each call to the MSA hyperpath assignment algorithm requires more than five hundred seconds to complete. This complexity can be prohibitively expensive considering that the number of hyperpaths could easily reach hundreds (if not thousands) even for average-sized problems.

Table 5.10. DTE algorithm hyperpaths produced by number of O-D pairs

# of O-D Pairs	Trucks	Hyperpaths by O-D pair	Used Hyperpaths	DTE Iterations	Rel. Gap after 50 iter.	Time (s)
1	15	4	2	4	0.00005	30
2	25	{4,5}	{3,3}	5	0.02	303
3	35	{6,9,5}	{5,4,5}	9	0.32	2645
4	45	{5,9,5,3}	{4,4,5,3}	9	0.35	4805
5	55	{5,16,5,3,4}	{4,5,5,3,4}	16	0.18	8608

We finally apply the algorithm to solve the ten-city network at full scale, where trucks are assumed to start and end in the same city from all cities. Thus, there are ten O-D pairs in total. The numbers of loads available at each city are obtained from the real data. Three scenarios are tested. The average competition scenario represents an average level of competition in the network, in which the number of trucks initially available in a city is obtained as follows. For each city, we run the algorithm multiple times, each time with a varying number of initial trucks, until it generates roughly ¥2500 for each truck. This method helped balance cities with high demands, by placing more trucks at those locations. The second scenario increases the number of trucks in each city by 50% to reflect a higher level of competition. Also, a lower level of competition decreases the number of trucks in each city by 50% to reflect a lower level of competition. It took our algorithm 5, 16, and 36 hours to solve the low, average, and high competition cases (at a convergence criterion of 0.35). Given the slow convergence of the MSA algorithm, we do not expect a much better solution could be obtained within a reasonable amount of time.

The DTE solutions are compared against three benchmarks: (1) a myopic routing solution where trucks try to maximize only the next decision, and always bid at the average price; (2) a recursive routing solution where trucks anticipate the expected profits in the future, but still always bid at the average price; and (3) the initial solution obtained by the DTE algorithm, i.e., assigning all flows to the first set of hyperpaths that optimizes both routing and bidding decisions (interactions between these decisions are ignored, however).

Table 5.11. Low competition scenario results, $\epsilon = 0.35$

O-D Pair	Trucks	Initial Profit	Added Hyperpaths	Used Hyperpaths	Final Profit	% Change
Anhui	10	3455	1	1	3455	0
Beijing	5	3390	8	4	3402	0.3
Chongqing	81	2422	5	3	3198	32.1
Fujian	18	2982	6	4	2812	-5.7
Gansu	8	3733	1	1	3733	0
Guangdong	118	3223	4	2	3256	1.0
Guangxi	80	3402	4	3	3378	0.7
Guizhou	85	2586	9	5	2386	-6.7
Hainan	28	3181	4	3	3219	1.2
Hebei	4	2822	7	4	2815	-0.2

	Avg Profit ¥	Loads Served	Empty kms	Loaded kms	% Waiting
Myopic Baseline	1701	1158	73.4K	1.03M	22.0
Recursive Baseline	1481	1120	1400	838K	34.9
Baseline	2978	1068	700	978K	26.3
Equilibrium	3089	1110	11148	1.05M	21.0
% Change	3.7	4.9	1492	7.4	-20.1

Table 5.12. Avg competition scenario results, $\epsilon = 0.35$

O-D Pair	Trucks	Initial Profit	Added Hyperpaths	Used Hyperpaths	Final Profit	% Change
Anhui	20	1690	10	5	2486	47.1
Beijing	9	2558	12	7	2639	3.2
Chongqing	161	1608	10	4	1313	-18.3
Fujian	36	1783	5	3	2427	36.1
Gansu	15	2603	8	3	1386	-46.8
Guangdong	235	1573	17	7	1989	26.4
Guangxi	160	2822	17	8	2961	4.9
Guizhou	170	1944	9	3	2732	40.5
Hainan	55	2714	17	3	2994	10.3
Hebei	7	908	17	10	1428	57.3

	Avg Profit ¥	Loads Served	Empty kms	Loaded kms	% Waiting
Myopic Baseline	1173	2327	224K	1.86M	27.9
Recursive Baseline	853	1931	84.8K	1.36M	46.3
Baseline	1989	1958	38220	1.53M	39.4
Equilibrium	2273	2049	82040	1.68M	23.3
% Change	14.3	4.6	114.7	9.7	-69.1

Table 5.13. High competition scenario results, $\epsilon = 0.35$

O-D Pair	Trucks	Initial Profit	Added Hyperpaths	Used Hyperpaths	Final Profit	% Change
Anhui	30	960	19	7	1951	103.3
Beijing	14	1385	22	5	765	-44.8
Chongqing	242	1332	8	5	1276	-4.2
Fujian	54	760	20	4	2573	238.6
Gansu	23	1183	4	1	226	-80.9
Guangdong	353	455	15	8	1144	151.6
Guangxi	240	2554	16	4	2647	3.6
Guizhou	255	1038	8	3	166	-84.0
Hainan	83	2550	20	5	2762	8.3
Hebei	11	291	13	2	-217	-174.5

	Avg Profit ¥	Loads Served	Empty kms	Loaded kms	% Waiting
Myopic Baseline	689	3184	400K	2.42M	37.3
Recursive Baseline	455	2524	187K	1.71M	54.9
Baseline	1296	2680	240K	2.01M	47.7
Equilibrium	1403	2196	177K	1.76M	38.4
% Change	8.2	-26.3	-26.3	-12.4	-19.6

The top half of Tables 5.11 - 5.13 show the results for each O-D pair in both the average (left side) and high competition (right side) cases. On the left, the first column reports the name of the base city; the second shows the number of trucks; the third shows the average profit after the DTE algorithm completed initialization; the fourth and fifth columns report the total number of hyperpaths generated and used, respectively; the sixth column shows the average profit when the DTE algorithm was terminated; and the last column shows the relative difference between Column three and Column six. Two observations are noteworthy from the top half of the table. First, the DTE solution is not always better than the initial solution (i.e., the third benchmark mentioned above), although several more O-D pairs benefit. Some city, such as Gansu, see their profit drop by almost 50% after implementing the DTE solution. Second, the average number of used hyperpaths is quite small: about 5 per O-D pair for the average competition case and a little more than 4 for the high competition case.

The bottom half of Tables 5.11 - 5.13 compares the DTE solutions with the three benchmark solutions. The last row shows how much the DTE solution improves over the third benchmark (i.e., the initial solution) for different metrics. On the left side, the second column reports the average profit per O-D pair; the third shows the total number of loads served; the fourth and fifth columns show the miles traveled without and with loads, respectively; and the sixth column reports the waiting time as the percentage of the planning horizon. With the average competition, the DTE solution generated 15% more profits, served about 5% more loads, traveled 10% more distance with loads, and cut the waiting time by 17 percentage points, compared to the initial solution benchmark. Interestingly, it also more than doubled the empty travel distance in the initial solution. As expected, the higher competition lowered the improvement in profitability delivered by the DTE solution. More interestingly, the DTE solution achieves this improvement in a completely different manner, by driving less and shipping fewer loads. Clearly, the intensified competition forces the trucks to be more selective on what loads to deliver and where to get them.

The myopic solution serves the most loads but also has low profitability, even compared to the initial solution benchmark. Its empty travel distance is exceedingly high compared to all other cases, evidently one of the worst consequences for being “myopic”. The performance of the recursive solution is worst because it only uses a single sub-optimal hyperpath per O-D pair that may be quickly saturated. This result highlights the importance of the optimal bidding. The initial solution benchmark also uses a single hyperpath for each O-D pair, but thanks to optimal bidding, it generates a much higher profit compared to the recursive solution.

5.4. Summary

The OFEX routing model is a practical approach for developing routing plans for trucks in large logistic networks. We have demonstrated that, for data from a real OFEX network in China, the OFEX routing model generates routing hyperpaths that significantly improve truck expected profits. Furthermore, it achieves this with fast run-times, making it possible to run repeatedly and efficiently for many trucks, which could be a useful tool for any OFEX platform.

The DTE model allows for a network of trucks, all utilizing the OFEX routing model, to generate an iterative series of hyperpaths to work toward a network equilibrium that equalizes expected profits for trucks identified by an O-D pair. Compared to the baseline hyperpaths, the equilibrium solution generates significantly improved profits, increased numbers of goods transported, and less downtime, improving profits for both the shipping companies and the truckers.

CHAPTER 6

Conclusions and Future Research

Online freight exchange (OFEX) firms are rapidly rising and promising to improve efficiency in the trucking industry by matching trucks to loads more intelligently. Their magic wand, which only became available in the past decade, is a massive mobile computing platform that allows direct interactions with the agents (shippers and truckers) and collects and processes data generated from their activities. This dissertation proposed a truck routing model that leverages the immense power of such a platform. We finish the dissertation in this chapter with a summary of the contributions, main conclusion, and potential areas of future work.

6.1. Summary of contributions

(1) Developed a hyperpath-based truck routing model that is based on optimized bidding within an OFEX platform. We demonstrated that the OFEX routing model formulation provides a way to generate robust routing plans with full recourse for truckers in addition to being very fast for realistically sized networks. This model is a capable engine for generating hyperpath-based routing and bidding strategies for trucks in the DTE problem.

(2) Demonstrated the practical significance of the methodology on specific instances using real OFEX data from China. The OFEX routing model generated expected optimal profit routing and bidding strategies for a truck utilizing an online freight

exchange platform that can increase profits by over 200% from the status quo. Furthermore, the DTE problem shows the network could increase profits for all trucks by 10% or more, potentially increasing profits by hundreds of millions of dollars for in China's highly fragmented trucking network.

(3) Established a mathematical foundation for a new research area ripe for application and analysis with potential for significant economic and environmental impacts. By laying the foundational methods, the research community can build upon this foundation to analyze many other networks: from a localized Uber driver network to analyze efficient and fair time-dependent pricing in a city, to a national shipping network to find the optimal place for a new distribution center to alleviate shipping imbalance, many possibilities exist.

6.2. Conclusions

The proposed OFEX routing problem seeks to determine a hyperpath in a space-time expanded network (STEN) that maximizes the expected profits for a given origin-destination pair and a tour duration. At the core of the OFEX routing problem is a combined pricing and bidding (COPA) model that simultaneously (1) considers the probability of winning a load at a given bid price; (2) anticipate the future profits corresponding to the current decision; (3) accounts for the competition in the market; and (4) prioritize the bidding order among possible load options. To the best of our knowledge, such a problem has not been studied in the context of freight logistics.

We show that, on a STEN, the OFEX routing problem can be formulated as a dynamic program and solved in polynomial time. Results from numerical experiments, constructed using real-world data from a Chinese OFEX platform, indicate that the proposed COPA model could (1) improve truck's expected profits up to 400%, compared to the myopic and recursive benchmark solutions built to represent the state of the practice; and (2) enhance the robustness of the overall profitability against the impact of market competition and spatial variations. Thus, the algorithm could potentially be used to help individual truckers discover more profitable shipping tours as they navigate through the OFEX platform.

This research showed how emerging OFEX platforms could help consolidate the highly fragmented industry and improve its efficiency by delivering automated "optimal" guidance to all trucks. The proposed DTE model helps predict system-wide truck flows (including empty truck flows) and identify efficiency improvements gained by network-wide visibility.

The proposed DTE model is formulated as a variational inequality problem and solved using a specialized heuristic algorithm. The algorithm consists of three modules: a dynamic network loading procedure for mapping hyperpath flows onto the freight network, a column generation scheme for creating hyperpaths as needed, and a method of successive average for equilibrating profits on existing hyperpaths. The results of numerical experiments show that (1) the proposed algorithm offers a viable tool to find approximate DTE solutions, but its convergence is rather slow, especially when the effect of cross O-D interactions are significant; (2) the DTE solutions improves the profitability of the system with wide margin compared to the benchmark solutions that either ignore interactions between trucks' decisions or operate trucks according to sub-optimal routing decisions.

In conclusion, the DTE research lays a foundation for analyzing new equilibrium problems that deal with profit maximizing endeavors, such as the OFEX trucking equilibrium demonstrated in this dissertation. The techniques developed in this dissertation could be extended and refined to analyze other for-profit transportation networks such as an Uber driver network equilibrium in a city, or a crowd-sourced shipping network in a region of interest. Although the model presented does not embody the theoretical properties that would guarantee unique equilibrium solutions, the algorithms implemented show promising convergence characteristics, on simple examples and real-world data networks.

6.3. Future Research

This dissertation study opens up many possible directions for future research. The first area of interest is building upon the user-equilibrium solution to explore the system optimal solution. Once the system optimal solution is identified, then optimal pricing of the network can be explored similar to optimal toll pricing research.

Improving the computational efficiency of the DTE assignment algorithm is high on the immediate future to-do list. Part of the reason why the computation time reported in the numerical results seems quite high has to do with the programming tool (Octave, a free version of MATLAB) selected to develop the prototype code. It is expected that switching to a more efficient language (e.g., C++) would dramatically reduce the computation time. In addition, all three components of the algorithm can be tuned for better efficiency.

Another possible direction is to explore a system optimal solution for a fleet of trucks, which considers intra-fleet competitions. Such an investigation could also help the OFEX platform develop strategies to improve the efficiency of the entire system, not just individual trucks. Finally, the methodologies developed in this dissertation could be extended and refined to analyze other for-profit transportation systems such as ride-sourcing platforms (e.g., Uber) or crowd-sourced shipping platforms (e.g., Postmates).

Another area of research could be to explore modeling the user-defined interaction with the OFEX platform. So far, we have assumed that trucks will follow the OFEX plans generated. However, a more plausible scenario is that the platform may try to "control" the tours of trucks, by allowing truckers to define their constraints. The routing model could be revised such that a system objective - such as minimizing the total empty truck flows or waiting time - can be achieved. This revised system would create a problem similar to the vehicle allocation problem, although an OFEX platform would never have - nor would they desire - the kind of control that a traditional trucking company has over its fleet, it allows for different levels of control, and one could measure how much control is worth.

A future study could also examine how the several essential inputs used in the OFEX routing model, such as empty truck flow and the number of competing trucks, can be reliably estimated from mining the diverse sources of data collected on OFEX platforms. Improving these inputs improves the output of the model.

Future work includes studying larger networks for the DTE problem, and how to apply optimal price adjustments to the network to shift the equilibrium to improve network efficiency. Finally, to try to mitigate the effect of the complexity of the DLA, pursuing other methods and models to generate the profit functions would be very beneficial research as the problem is scaled higher.

Another possible methodology to research this problem is to use simulation. The advantage of simulation is the realism that can be implemented and modeled and one can leverage the law of large numbers in the experimental design. The main realism elements that simulation would augment for this problem framework is the bidding process for jobs, and another way to analyze the influence of time on the recourse options. The bidding process is generally ignored in the OFEX routing model and accounted for by utilizing an expected profit number for each time period and location. This process enables the analytical models' profit functions to be monotonically decreasing functions and tractable. However, simulation can enable this realistic tradeoff between time and bid order a reality. However, the data that is available and collected so far is insufficient for a realistic implementation because it does not include information about bids for jobs that did not win, and jobs that were never delivered because they were not bid for. Getting this necessary data could prove difficult. That being said, in the literature, most research utilizes simulation-based dynamic assignment models, and at a minimum, this is interesting future work.

Finally, the methods here can be extended to other research areas. It would be easy to apply this framework to other profit-maximizing networks, like the Uber driver network in Chicago. Similar to the OFEX platform, the Uber platform matches drivers and passengers, and Uber has the data that could enable calculations of the likelihood of arriving passengers in a zone by time, and average profitability of each zone. One could use this to help an Uber driver plan a more profitable route chain with recourse for the amount of time s/he wants to drive. This analysis could help improve profits significantly against how most people interact with the application as an Uber driver, surfing the smartphone application to find the next passenger nearest their current location.

References

[1] American Trucking Associations. Reports, trends, and statistics, 2015.

[2] Angelelli, E., Archetti, C., Filippi, C., and Vindigni, M. The probabilistic orienteering problem. *Comput. Oper. Res. 81*, C (May 2017), 269–281.

[3] Beckmann, M. J., McGuire, C. B., and Winsten, C. B. Studies in the economics of transportation. *RAND Corporation, RM-1488-PR* (1955).

[4] Bellman, R. *On a Routing Problem.* P (Rand Corporation). Rand Corporation, 1956.

[5] Cantarella, G. E. A general fixed-point approach to multimode multi-user equilibrium assignment with elastic demand. *Transportation Science 31*, 2 (1997), 107–128.

[6] Chabini, I. Discrete dynamic shortest path problems in transportation applications: Complexity and algorithms with optimal run time. *Transportation Research Record: Journal of the Transportation Research Board 1645*, 1 (1998), 170–175.

[7] Chen, P. W., and Nie, Y. M. Optimal transit routing with partial online information. *Transportation Research Part B: Methodological 72* (2015), 40–58.

[8] Chi Institute Consulting Group. 2016 china's highway logistics industry status analysis and forecast of development trends, 2016.

[9] Chriqui, C., and Robillard, P. Common bus lines. *Transportation Science 9*, 2 (1975), 115–121.

[10] Cominetti, R., and Correa, J. Common-lines and passenger assignment in congested transit networks. *Transportation Science 35*, 3 (2001), 250–267.

[11] Cooke, K., and Halsey, E. The shortest route through a network with time-dependent intermodal transit times. *Journal of Mathematical Analysis and Applications 14* (1966), 492–498.

[12] Dafermos, S. Traffic equilibrium and variational inequalities. *Transportation Science 14*, 1 (1980), 42–54.

[13] Dantzig, G., Fulkerson, R., and Johnson, S. Solution of a large-scale traveling-salesman problem. *Operations Research 2* (1954), 393–410.

[14] De Cea, J., and Fernández, E. Transit assignment for congested public transport systems: an equilibrium model. *Transportation science 27*, 2 (1993), 133–147.

[15] Dejax, P. J., and Crainic, T. G. Survey papera review of empty flows and fleet management models in freight transportation. *Transportation Science 21*, 4 (1987), 227–248.

[16] Desaulniers, G., Desrosiers, J., Erdmann, A., Solomon, M. M., and Soumis, F. *VRP with Pickup and Delivery.* Society of Industrial and Applied Mathematics, 2002, ch. 9, pp. 225–242.

[17] Dijkstra, E. W. A note on two problems in connexion with graphs. *NUMERISCHE MATHEMATIK 1*, 1 (1959), 269–271.

[18] Domencich, T. A., and McFadden, D. L. *Urban Travel Demand: A Behavioral Analysis.* North-Holland, Amsterdam, The Netherlands, 1975.

[19] Dreyfus, S. E. An appraisal of some shortest-path algorithms. *Oper. Res. 17*, 3 (June 1969), 395–412.

[20] Eaton, J. W., Bateman, D., Hauberg, S., and Wehbring, R. *GNU Octave version 4.2.1 manual: a high-level interactive language for numerical computations.* CreateSpace Independent Publishing Platform, 2016. ISBN 1441413006.

[21] Emura, T., and Liao, Y.-T. Critical review and comparison of continuity correction methods: The normal approximation to the binomial distribution. *Communications in Statistics - Simulation and Computation 0*, 0 (2017), 1–20.

[22] Fan, K. A minimax inequality and applications, inequalities iii (o. *Inequalities 3* (1972), 103–113.

[23] Fernández, J. L., de Cea, J. C., and Soto, A. O. A multi-modal supplydemand equilibrium model for predicting intercity freight flows. *Transportation Research Part B: Methodological 37*, 7 (2003), 615 – 640.

[24] Figliozzi, M. A., Mahmassani, H. S., and Jaillet, P. Pricing in dynamic vehicle routing problems. *Transportation Science 41*, 3 (2007), 302 – 318.

[25] Ford, L. R. *Network flow theory.* Santa Monica, Calif: Rand Corp., 1956.

[26] Friesz, T. L., Gottfried, J., and Morlok, E. K. *A freight network equilibrium model.* Department of Civil and Urban Engineering, University of Pennsylvania, 1981.

[27] Friesz, T. L., Tobin, R. L., and Harker, P. T. Predictive intercity freight network models: the state of the art. *Transportation Research Part A: General 17*, 6 (1983), 409–417.

[28] Friesz, T. L., Viton, P. A., and Tobin, R. L. Economic and computational aspects of freight network equilibrium models: A synthesis. *Journal of Regional Science 25*, 1 (1985), 29–49.

[29] Gansterer, M., Küçüktepe, M., and Hartl, R. F. The multi-vehicle profitable pickup and delivery problem. *OR Spectrum 39*, 1 (Jan 2017), 303–319.

[30] Golden, B. L., Levy, L., and Vohra, R. The orienteering problem. *Naval Research Logistics 34*, 3 (1987), 307–318.

[31] Hamdouch, Y., and Lawphongpanich, S. Schedule-based transit assignment model with travel strategies and capacity constraints. *Transportation Research Part B: Methodological 42*, 7 (2008), 663 – 684.

[32] Hamdouch, Y., Marcotte, P., and Nguyen, S. Capacitated transit assignment with loading priorities. *Mathematical Programming 101*, 1 (Aug 2004), 205–230.

[33] Hamdouch, Y., Szeto, W., and Jiang, Y. A new schedule-based transit assignment model with travel strategies and supply uncertainties. *Transportation Research Part B: Methodological 67* (2014), 35 – 67.

[34] Harker, P. T., and Friesz, T. L. Prediction of intercity freight flows, i: theory. *Transportation Research Part B: Methodological 20*, 2 (1986), 139–153.

[35] Hurley, W. J., and Petersen, E. R. Nonlinear tariffs and freight network equilibrium. *Transportation Science 28*, 3 (1994), 236.

[36] Jaillet, P. A priori solution of a traveling salesman problem in which a random subset of the customers are visited. *Operations Research 36*, 6 (1988), 929–936.

[37] Janson, B. N. Dynamic traffic assignment for urban road networks. *Transportation Research Part B: Methodological 25*, 2 (1991), 143 – 161.

[38] Kuyzu, G., Akyol, C. G., Ergun, Ö., and Savelsbergh, M. Bid price optimization for truckload carriers in simultaneous transportation procurement auctions. *Transportation Research Part B: Methodological 73*, Supplement C (2015), 34 – 58.

[39] Laporte, G., Louveaux, F. V., and Mercure, H. A priori optimization of the probabilistic traveling salesman problem. *Operations Research 42*, 3 (1994), 543–549.

[40] Li, Yong, . Investment prospect report for chinese logistics industry (in chinese). Technical report, Deloitte Research, 2016.

[41] Li, Q., Chen, P. W., and Nie, Y. M. Finding optimal hyperpaths in large transit networks with realistic headway distributions. *European Journal of Operational Research 240*, 1 (2015), 98 – 108.

[42] Liu, H., He, X. S., and He, B.-S. *Method of Successive Weighted Averages (MSWA) and Self-Regulated Averaging Schemes for Solving Stochastic User Equilibrium Problem*, vol. 9. 12 2009, pp. 485–503.

[43] Marcotte, P., Nguyen, S., and Schoeb, A. A strategic flow model of traffic assignment in static capacitated networks. *Operations Research 52*, 2 (2004), 191 – 212.

[44] Merchant, D. K., and Nemhauser, G. L. A model and an algorithm for the dynamic traffic assignment problems. *Transportation Science 12*, 3 (1978), 183–199.

[45] Merchant, D. K., and Nemhauser, G. L. Optimality conditions for a dynamic traffic assignment model. *Transportation Science 12*, 3 (1978), 200–207.

[46] Mes, M., van der Heijden, M., and Schuur, P. Look-ahead strategies for dynamic pickup and delivery problems. *OR Spectrum 32*, 2 (Apr 2010), 395–421.

[47] Mes, M., van der Heijden, M., and van Harten, A. Comparison of agent-based scheduling to look-ahead heuristics for real-time transportation problems. *European Journal of Operational Research 181*, 1 (2007), 59 – 75.

[48] Misra, S. Linear programming of empty wagon disposition. *Rail International 3*

(03 1972), 151–158.

[49] Mitrović-Minić, S., and Laporte, G. Waiting strategies for the dynamic pickup and delivery problem with time windows. *Transportation Research Part B: Methodological 38*, 7 (2004), 635 – 655.

[50] Mullaney, D., and Wang, Z. Improving efficiency in chinese trucking and logistics. *Rocky Mountain Institute* (Aug 2016).

[51] Nagurney, A. *Network Economics : A Variational Inequality Approach.* Kluwer Academic, 1993.

[52] Nguyen, S., and Pallottino, S. Equilibrium traffic assignment for large scale transit networks. *European Journal of Operational Research 37*, 2 (1988), 176 – 186.

[53] Orda, A., and Rom, R. Shortest-path and minimum-delay algorithms in networks with time-dependent edge-length. *J. ACM 37*, 3 (July 1990), 607–625.

[54] Patriksson, M. Algorithms for computing traffic equilibria. *Networks and Spatial Economics 4*, 1 (Mar 2004), 23–38.

[55] Peeta, S., and Mahmassani, H. S. System optimal and user equilibrium time-dependent traffic assignment in congested networks. *Annals of Operations Research 60*, 1 (Dec 1995), 81–113.

[56] Peeta, S., and Ziliaskopoulos, A. K. Foundations of dynamic traffic assignment: The past, the present and the future. *Networks and Spatial Economics 1*, 3 (Sep 2001), 233–265.

[57] Powell, W. B. A stochastic model of the dynamic vehicle allocation problem. *Transportation science 20*, 2 (1986), 117–129.

[58] Powell, W. B. An operational planning model for the dynamic vehicle allocation

problem with uncertain demands. *Transportation Research Part B: Methodological 21*, 3 (1987), 217 – 232.

[59] Powell, W. B. A stochastic formulation of the dynamic assignment problem, with an application to truckload motor carriers. *Transportation Science 30*, 3 (1996), 195–219.

[60] Powell, W. B. *Approximate Dynamic Programming - Solving the Curses of Dimensionality.* Wiley, 2007.

[61] Powell, W. B., and Sheffi, Y. The convergence of equilibrium with predetermined step sizes. *Transportation Science 16* (1982), 45–55.

[62] Powell, W. B., Sheffi, Y., Nickerson, K. S., Butterbaugh, K., and Atherton, S. Maximizing profits for north american van lines' truckload division: A new framework for pricing and operations. *Interfaces 18*, 1 (1988), 21–41.

[63] Powell, W. B., Sheffi, Y. Y., and Thiriez, S. The dynamic vehicle allocation problem with random demands. In *Proceedings of the Ninth International Symposium on Transportation and Traffic Theory* (The Netherlands, 1984), J. Volmuller and R. Hamerslag, Eds., VNU Science Press, pp. 357–374.

[64] Powell, W. B., Towns, M. T., and Marar, A. On the value of optimal myopic solutions for dynamic routing and scheduling problems in the presence of user noncompliance. *Transportation Science 34*, 1 (2000), 67–85.

[65] Qiu, X., Feuerriegel, S., and Neumann, D. Making the most of fleets: A profit-maximizing multi-vehicle pickup and delivery selection problem. *European Journal of Operational Research 259*, 1 (2017), 155 – 168.

[66] Ritzinger, U., Puchinger, J., and Hartl, R. F. A survey on dynamic and

stochastic vehicle routing problems. *International Journal of Production Research 54*, 1 (2016), 215–231.

[67] Robbins, H., and Monro, S. A stochastic approximation method. *The Annals of Mathematical Statistics 22*, 3 (1951), 400–407.

[68] Schrijver, A. *Combinatorial Optimization, Polyhedra and Efficiency*, vol. 24. Springer-Verlag Berlin Heidelberg, 2003.

[69] Sheffi, Y. *Urban Transportation Networks: Equilibrium Analysis With Mathematical Programming Methods.* Prentice Hall, Jan 1985.

[70] Shmoys, D., Lenstra, J., Kan, A., and Lawler, E. *The Traveling Salesman Problem.* Wiley Interscience Series in Discrete Mathematics. John Wiley & Sons, 1985.

[71] Smith, M. The existence, uniqueness and stability of traffic equilibria. *Transportation Research Part B: Methodological 13*, 4 (1979), 295 – 304.

[72] Song, S. China drives toward sustainable freight policies. *GreenBiz* (2017).

[73] Spiess, H., and Florian, M. Optimal strategies: A new assignment model for transit networks. *Transportation Research Part B: Methodological 23*, 2 (1989), 83 – 102.

[74] Tjokroamidjojo, D., Kutanoglu, E., and Taylor, G. D. Quantifying the value of advance load information in truckload trucking. *Transportation Research Part E: Logistics and Transportation Review 42*, 4 (2006), 340 – 357.

[75] Topaloglu, H., and Powell, W. Incorporating pricing decisions into the stochastic dynamic fleet management problem. *Transportation Science 41*, 3 (2007), 281 – 301.

[76] Trozzi, V., Gentile, G., Bell, M., and Kaparias, I. Dynamic user equilibrium in public transport networks with passenger congestion and hyperpaths. *Transportation Research Part B 57* (2013), 266–285.

[77] Ulmer, M. *Approximate Dynamic Programming for Dynamic Vehicle Routing.* Springer International Publishing, 2017.

[78] USDOT. Beyond traffic. Tech. rep., United States Department of Transportation, 2016.

[79] Wardrop, J. G. Some theoretical aspects of road traffic research. *Proceedings, Institute of Civil Engineers II*, 1 (1952), 325–378.

[80] White, W. W. Dynamic transshipment networks: An algorithm and its application to the distribution of empty containers. *Networks 2*, 3 (1972), 211–236.

[81] White, W. W., and Bomberault, A. M. A network algorithm for empty freight car allocation. *IBM Systems Journal 8*, 2 (1969), 147–169.

[82] Wu, J. H., Florian, M., and Marcotte, P. Transit equilibrium assignment: a model and solution algorithms. *Transportation Science 28*, 3 (1994), 193–203.

[83] Wu, S., Swerdlow, F., Chakravarty, A., Witcher, B., and Kumar, S. Online retail forecast, 2017 to 2022 (us). Tech. rep., Forrester Inc., 2017.

[84] Yang, J., Jaillet, P., and Mahmassani, H. On-line algorithms for truck fleet assignment and scheduling under real-time information. *Transportation Research Record: Journal of the Transportation Research Board*, 1667 (1999), 107–113.

[85] Yang, J., Jaillet, P., and Mahmassani, H. Real-time multivehicle truckload pickup and delivery problems. *Transportation Science 38*, 2 (2004), 135–148.

[86] Yates, F. Contingency tables involving small numbers, and the χ^2 test. *Journal of*

the Royal Statistical Society 1, 2 (1934), 217–35.

[87] Ziliaskopoulos, A., and Mahmassani, H. Time-dependent shortest path algorithm for real-time intelligent vehicle highway system applications. *Transportation Research Record 1408* (1993), 94–100.

[88] Zolfagharinia, H., and Haughton, M. The benefit of advance load information for truckload carriers. *Transportation Research Part E: Logistics and Transportation Review 70*, Supplement C (2014), 34 – 54.

[89] Zolfagharinia, H., and Haughton, M. Effective truckload dispatch decision methods with incomplete advance load information. *European Journal of Operational Research 252*, 1 (2016), 103 – 121.

[90] Zolfagharinia, H., and Haughton, M. A. Operational flexibility in the truckload trucking industry. *Transportation Research Part B: Methodological 104*, Supplement C (2017), 437 – 460.

APPENDIX A

Notation

Symbol	Definition
Sets	
$\mathcal{A}$	= set of all options for a given city at a given time interval $\|\mathcal{A}\| = 2n - 1$
$\mathcal{A}_d$	= set of dummy connectors in the STEN
$\mathcal{A}_e$	= set of links corresponding to empty moves in the STEN
$\mathcal{A}_f$	= set of links corresponding to loaded moves in the STEN
$\mathcal{A}_w$	= set of links corresponding to waiting in the STEN
$\mathcal{E}$	$= \mathcal{A}_f \bigcup \mathcal{A}_e \bigcup \mathcal{A}_w \bigcup \mathcal{A}_d$
$\mathfrak{G}(\mathcal{N}, \mathcal{E}, T)$	= space-time expanded network (STEN)
$\mathcal{H}_w$	= set of all generated hyperpaths h_w that connect origin i_o and destination j_T of hyperpath w
$\mathcal{K}$	= set of all simple paths of $\mathfrak{G}$
$\mathcal{L}$	= set of nodes in the STEN corresponding to real cities, $\|L\| = Tn$, $\mathcal{L}'$, corresponding to dummy cities
M	= set of information for general MDP model
$\mathcal{N}$	$= \mathcal{L} \bigcup \mathcal{L}'$

$\mathcal{Q}$ = set of all feasible loaded options for a given city at a given time interval. $\|\mathcal{Q}\| = m$

Φ = set of all bid orders

S = general state variable for MDP model, $S_t^{mathbfx}$ = state post decision

T = set of all times, $t \in 0 \cdots T$ in planning horizon

$\mathcal{W}$ = set of all unique O-D pairs, each w represented by tour start time t_s, location r, and tour end time t_e, and end location s, $|W| = \eta$

$\bar{W}$ = information set that becomes known in general MDP model

Data

α_e = cost per Δ for empty travel, α_f = loaded, α_w = waiting

c_a^t = cost incurred on arc a at t

Δ = length of operating interval (1 hour in this dissertation)

g_a^t = number of loads on arc a at t

γ = cost per Δ for loading\unloading

$\hat{i}$ = origin city at start of planning horizon for a w

$\hat{j}$ = destination city at end of planning horizon for a w

l_a^t = lower limit of a winning bid on arc a at t

n = number of cities in the region

p = probability of state transition in P in general MPD model

τ_a^t = travel time on arc a at t

u_a^t = upper limit of a winning bid on arc a at t

Indicator Variables

$\delta^w_{ha_t}$ = hyperpath-link incidence: $\delta^w_{ha_t} = 1$ if $a_t \in h_w$ and 0 otherwise)

Decision Variables

e^t_a = number of residual (unmatched) loads ready for delivery at time t on option a, $\hat{e}^t_a$ = dummy version

$\mathbf{f}$ = hyperpath flow vector

$f^{wt_r}_h$ = flows of trucks in $B^{t_r}_w$ assigned to hyperpath h^{wt_r}

ϕ^{tw}_{ah} = bid orders for option a, on O-D pair w on hyperpath h at time t

π = policy for general MDP model

x^t_a = revenue (equal to bid price) earned on arc a at t

$\mathbf{x}$ = set of decision for general MDP model representing bid prices x and bid order ϕ

ζ^{tw}_{ah} = bid prices for option a, on O-D pair w on hyperpath h at time t

General Variables

a = a movement option, of ij

b_a = number of trucks bidding on option a

$B^{t_r}_w$ = number of trucks that start at city r at time t_r and wish to return to city s at T

d^t_a = number of loads ready for delivery at time t on option a

F_{ij} = cumulative distribution function of unit load price between i and j

$\bar{f}$ = fallback option

$\check{f}_{ith}^{wt_r}$ = residual flow that arrives at node (i, t)

H_i^t = number of trucks available at city i at t

λ_k = probability of traversing simple path k

ω_k = profit associated with path k

p_a^{0t} = probability of winning with expected value bid on load a at t

p_a^t = $F_a(y_a^t)$, probability of winning a load on arc a at t with bid price y_a^t

$p_{ha_t}^w$ = percentage of truck flows that pass through a_t on hyperpath h_w

$\bar{p}_0$ = probability of winning with expected value bid, averaged over time and bids.

$\bar{p}_{ah}^{wt_r}$ = expected probability of choosing link a on $h_w^{t_r}$

$\mathbf{p}$ = actual link choice probability vector

π_a^t = probability of traveling on arc a departing at t

ψ = multiplier associated with load to demand ratios

r = relative gap

ϱ = minimum (winning) bid value

$\bar{s}_a$ = sum of the demand for option a

$\dot{s}_a$ = sum of the winning bid demand for option a

σ_a^t $= y_a^t - c_a^t$, profit earned on arc a at t

t = time

ϑ_a = load to demand ratio for option a

$\mathbf{u}$ = hyperpath profits vector

u_h^w = expected profit associated with hyperpath h_w

v_{ah}^{tw} = truck flow on link a on O-D pair w on hyperpath h at time t, full vector is also denoted as X

V_i^t = maximum expected profit starting from city i at time t to end of process

w = shortcut to the O-D pair $r - s$, with $w^- = r$ and $w^+ = s$

$\acute{W}_h$ = total profit associated with hyperpath h

$z_{\bar{f}}$ = expected profit of the fallback option

Auxiliary Variables

ν_h^w = maximum expected profit for the truck drivers in class w

APPENDIX B

Estimating the number of competing trucks for each option (b_a)

To apply the logit model, we first define the utility of each option a as

$$U_a = \eta_0 + \eta_1 \frac{\bar{\sigma}_a}{\tau_a + 2} + \eta_2 g_a + \epsilon_a, \tag{B.1}$$

where $\eta_i, i = 0, 1, 2$ are coefficients and ϵ_a represents unobserved attributes (error term). Ideally, η_i should be calibrated from the real observations of truckers' choice behavior. As we neither have the data nor consider the estimation of b_a the focus of the present study, the utility function (B.2) is further simplified by scaling $\bar{\sigma}_a$ and g_a into a given range $[\kappa, \theta]$, i.e.,

$$U_a = S\left(\frac{\bar{\sigma}_a}{\tau_a + 2}\right) + S(g_a) + \epsilon_a, \forall a \in \mathcal{A}' \tag{B.2}$$

$$S(y'_a) = \frac{y'_a - \min(y_a, \forall a \in \mathcal{A}')}{\max(y_a, \forall a \in \mathcal{A}') - \min(y_a, \forall a \in \mathcal{A}')}(\theta - \kappa) + \kappa. \tag{B.3}$$

where κ and θ are the desired range, and $\mathcal{A}'$ is defined as the set of all loaded moves such that the estimated unit profit $\sigma_a/(\tau_a + 2\Delta) > \alpha_w$ (the cost of waiting). Then, assuming ϵ_a is an identically and independently distributed (IID) Gumbel variable, the choice probability of option a is given by

$$P_a = \frac{e^{U_a}}{\sum_b e^{U_b}}, \forall a. \tag{B.4}$$

Accordingly, the number of competing trucks is

(B.5a) $$\bar{b}_a = H_i \beta P_a, \ \forall a \in \mathcal{A}, \ \text{where}$$

(B.5b) $$\beta = \min\{1 + 0.2(||A|| - 1), \, 3\}.$$

The parameter β is a scalar multiplier that reflects the fact that trucks can bid on more than one load. The setting of β assumes that the number of options that each truck bids is proportional to the number of options, $||A||$, but is limited by an upper bound (3 is selected in this dissertation).