TRANSPORTATION RESOURCE MANAGEMENT IN LARGE-SCALE FREIGHT CONSOLIDATION NETWORKS

A Thesis
Presented to
The Academic Faculty

by

José Antonio Carbajal Orozco

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Industrial and Systems Engineering

Georgia Institute of Technology
December 2011

TRANSPORTATION RESOURCE MANAGEMENT IN LARGE-SCALE FREIGHT CONSOLIDATION NETWORKS

Approved by:

Alan L. Erera, Advisor
School of Industrial and Systems Engineering
Georgia Institute of Technology

Martin W. P. Savelsbergh, Advisor
Mathematics, Informatics, and Statistics
CSIRO

Ozlem Ergun
School of Industrial and Systems Engineering
Georgia Institute of Technology

Joel Sokol
School of Industrial and Systems Engineering
Georgia Institute of Technology

Richard T. Wong
Lead Operations Research Analyst
UPS

Date Approved: August 24, 2011

ACKNOWLEDGEMENTS

Completing this dissertation has been quite a journey! It has brought me immense professional and personal growth and a plethora of experiences that I will fondly remember.

I am very grateful to have worked with Alan Erera and Martin Savelsbergh. They have been exceptional advisors who guided, motivated, and supported me to develop as a researcher and make this dissertation possible. Their patience and encouragement was constant and unwavering in every stage of my life as a Ph.D. student, especially during bad times. It has been a pleasure having them as my mentors.

I would like to thank Ozlem Ergun, Joel Sokol, and Richard Wong for their time and effort to serve on my dissertation committee. I have greatly benefited from their insightful minds during our discussions about the research contained here as well as other unrelated projects.

I am also grateful to several fellow doctoral students at Georgia Tech. There are too many friends to whom I am indebted for many reasons, too many people to mention here, but any list would have to include Adem and Moin, for the countless hours we spent discussing homework assignments during the early years, Dexin, Feng, and Ralph, for all the fun outings and the honorary membership into the Chinese community, and Kate and Jackie, for being the best officemates ever.

Finally, I would like to thank my family whose support despite the distance has kept me going through personal hardships and has allowed me to complete this journey. To them, I dedicate this dissertation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

This dissertation proposes approaches that enable effective planning and control of mobile transportation resources in large-scale consolidation networks. We develop models, algorithms, and methodologies that are applied to fleet sizing and fleet repositioning. Three specific but interrelated problems are studied. The first two relate to the trade-offs between fleet size and repositioning costs in transportation resource management, while the third involves a dynamic empty repositioning problem with explicit consideration of the uncertainty of future requirements that will be revealed over time.

Chapter 1 provides an overview of freight trucking, including the consolidation trucking systems that will be the focus of this research.

Chapter 2 proposes an optimization modeling approach for analyzing the trade-off between the cost of a larger fleet of tractors and the cost of repositioning tractors for a trucking company operating a consolidation network, such as a less-than-truckload (LTL) company. Specifically, we analyze the value of using extra tractor repositioning moves (in addition to the ones required to balance resources throughout the network) to attain savings in the fixed costs of owning or leasing a tractor fleet during a planning horizon. The primary contributions of the research in this chapter are that (1) we develop the first optimization models that explore the impact of fleet size reductions via repositioning strategies that have regularity and repeatability properties, and (2) we demonstrate that substantial savings in operational costs can be achieved by repositioning tractors in anticipation of regional changes in freight demand.

Chapter 3 studies the optimal Pareto frontiers between the fleet size and repositioning costs of resources required to perform a fixed aperiodic or periodic schedule

of transportation requests. We model resource schedules in two alternative ways: as flows on event-based, time-expanded networks; and as perfect matchings on bipartite networks. The main contributions from this chapter are that (1) we develop an efficient re-optimization procedure to compute adjacent Pareto points that significantly reduces the time to compute the entire Pareto frontier of fleet size versus repositioning costs in aperiodic networks, (2) we show that the natural extension to compute adjacent Pareto points in periodic networks does not work in general as it may increase the fleet size by more than one unit, and (3) we demonstrate that the perfect matching modeling framework is frequently intractable for large-scale instances.

Chapter 4 considers robust models for dynamic empty-trailer repositioning problems in very large-scale consolidation networks. We investigate approaches that deploy two-stage robust optimization models in a rolling horizon framework to address a multistage dynamic empty repositioning problem in which information is revealed over time. Using real data from a national package/parcel express carrier, we develop and use a simulation to evaluate the performance of repositioning plans in terms of unmet loaded requests and execution costs. The main contributions from this chapter are that (1) we develop approaches for embedding two-stage robust optimization models within a rolling horizon framework for dynamic empty repositioning, (2) we demonstrate that such approaches enable the solution of very large-scale instances, and (3) we show that less conservative implementations of robust optimization models are required within rolling horizon frameworks.

Finally, Chapter 5 summarizes the main conclusions from this dissertation and discusses directions for further research.

CHAPTER I

INTRODUCTION

The focus of this dissertation is on mobile resource management problems arising in large-scale freight consolidation systems. Specifically, this study investigates these problems in the context of public consolidation trucking carriers, including both less-than-truckload (LTL) and parcel/express carriers. The problems addressed here are mainly related to fleet sizing and empty repositioning. Although these resource management problems are not new, there are now particular reasons to address them again. First, the size of the problems has significantly increased. Large national companies have shifted their focus from regional optimization of their operations to enterprise-wide optimization, and models developed in the past do not necessarily scale well when applied to these very large instances. Second, advances in computational capabilities have made it possible both to gather large volumes of information, which are now available to be used in models, and to allow the solution of large-scale mathematical programming problems. In particular, improvements in the strength of commercial solvers for integer and mixed-integer programming have made it possible to solve optimization models with hundreds of thousands of variables, such as the ones presented in this study.

This introductory chapter first provides an overview of freight trucking, including the consolidation trucking systems that will be the focus of this research. Then, the resource management problems that will be addressed in this thesis are introduced. Finally, the specific research objectives and the contributions of the dissertation are summarized providing an overview of the primary results of this research work.

1.1 Trucking Operations

Trucking remains the dominant mode of freight transportation in the United States, hauling about 11 billion tons of goods annually. Truck transportation in the U.S. accounts for 70 percent of the total tonnage and 80 percent of the total value of the domestic freight activity. The total revenue of the trucking industry is estimated to be about $650 billion and represents about 5 percent of U.S. Gross Domestic Product [3]. Furthermore, trucking is expected to gain additional ground in relation to other domestic transport over the next several years. By 2018, truck tonnage and revenue are expected to increase, respectively, about 30 percent and 72 percent [3]. Additionally, the industry is greatly fragmented with about 500,000 carriers, 214,000 of which are public and the rest private, and the largest 50 companies account for less than 30 percent of the market [3]. In this highly competitive environment, the profitability of individual carriers depends heavily on developing cost-efficient operations and finding out new ways to reduce costs without compromising service standards.

Trucking transportation systems are organized to provide a certain combination of cost and service to customers in certain geographic markets. *Truckload trucking* handles shipments between 10,000 and 50,000 lbs and operates using direct trailer or domestic rail container services in which no intermediate freight handling activities occur. This thesis will refer to a *container* generically as a trailer or container hauled by a truck providing the motive power. Customers of a truckload carrier are provided with one or more containers at an origin location where they are loaded, and these containers are then transported to a destination location, where they are unloaded. Sometimes, truckload moves include multiple pickups and/or multiple dropoffs, but this is less common than single origin to single destination moves. Importantly, public truckload carriers do not use terminals to transfer freight en route from origin to destination.

On the other hand, *less-than-truckload (LTL) trucking*, which handles customer shipments between 150 and 10,000 lbs, and *package/parcel express*, which typically handles smaller shipments less than 150 lbs, operate consolidated container services. Since the shipment requested by each customer does not fill an entire truck, transporting each such shipment directly from origin to destination is not economically viable. Consolidated container service providers collect and consolidate freight from multiple customers, and route shipments through a terminal network of transfer points to increase trailer utilization and to take advantage of transportation economies of scale.

Unlike truckload carriers providing direct origin to destination service, consolidation service providers operate a fixed network of terminals, which may be owned or leased. This network of terminals, and the transportation lanes connecting them, is often referred to as the *linehaul network* and typically exhibits a hub-and-spoke topology for larger networks. The terminals are used to consolidate outbound freight and de-consolidate inbound freight, where a sorting process is used to transfer shipments from arriving inbound containers to departing outbound containers. LTL terminals are typically operated as cross-dock facilities, where forklifts are used during the sort process to transfer goods. Parcel terminals typically include more sorting infrastructure within the terminal, such as various conveyance systems, to move parcels between container unpacking and repacking.

Both LTL and parcel carriers use a so-called pickup-and-delivery system to transport shipments from their origin location to an initial terminal. In parcel systems, pickup-and-delivery tours are typically operated with small delivery vans, while in LTL it is typical to use short trailers (“pups”) behind a city tractor. Each pickup truck tour (or tour segment) will pick up shipments from several customers in a small geographic area and transport them to a terminal serving the area, referred to as a satellite or end-of-line (EOL). End-of-line terminals serve as sorting centers and consolidation facilities for outbound freight. Since there usually is not enough freight collected at an EOL to build full truckloads direct to EOLs serving other areas, additional levels of consolidation frequently take place. Outbound freight from an EOL may be loaded into a container destined to a terminal that consolidates freight from many EOLs, often referred to as a breakbulk. Breakbulks handle enough freight to build and dispatch cost-efficient loads with more nearly full containers. In an LTL system, a typical shipment might travel from an origin EOL to an origin breakbulk, then to a destination breakbulk and finally to a destination EOL. Between each pair of terminals, the shipment travels in a so-called *load* that needs to be transported by the carrier.

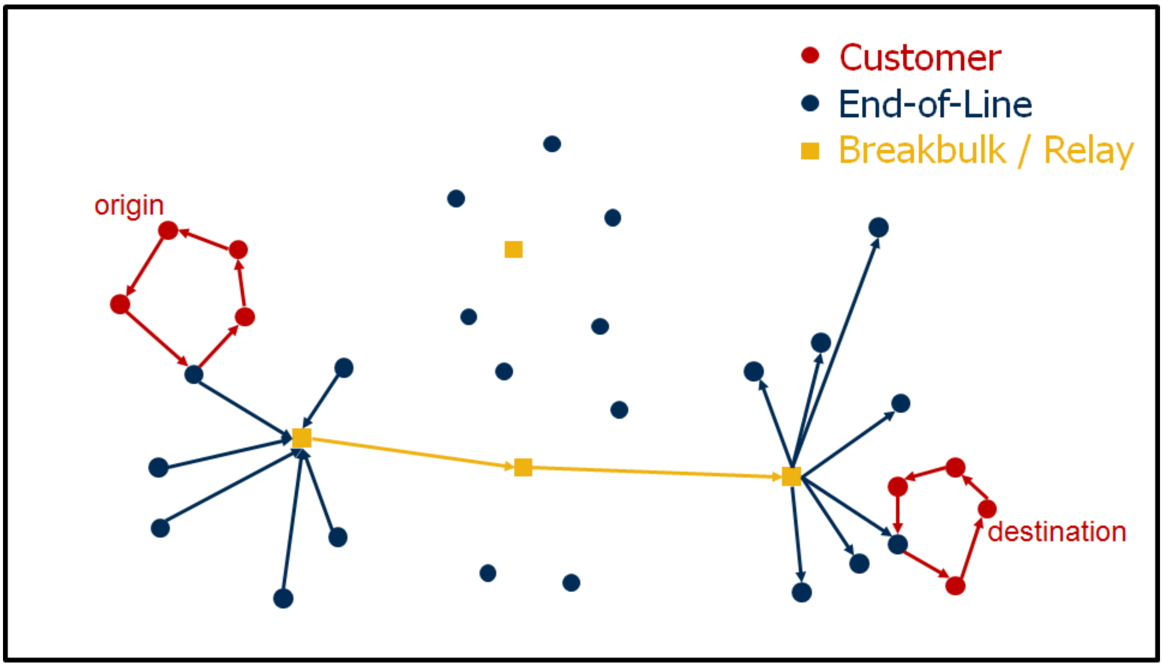


Figure 1: Consolidation system operations

Large national LTL or parcel/express shipping carriers in the United States might operate 300 to over 600 EOL terminals, with approximately one breakbulk for every 20 to 30 EOLs. In addition to their role in consolidation, terminals also serve as temporary storage locations for freight, trailers, or tractors, provide a place for servicing trailers and tractors (maintenance or repair), and provide a base for drivers and dispatchers.

Both types of trucking system configurations use different types of mobile resources to move loads and thus provide freight movement services to their customers, including *containers*, the vessels into which freight is packed for movement, *tractors*, the vehicles or power units used to transport the trailers, and *drivers*, the persons or teams operating the vehicles.

1.2 Transportation Resource Management

Numerous planning and control problems arise in the management of a *fleet* of mobile transportation resources, where a fleet may represent a group of tractors, trailers, or drivers. An important characteristic that differentiates these problems from other industrial planning and control problems is the fact that resources move across geography and across time. Additionally, the dynamics and large-scale nature of the trucking operating environment add to the complexity of resource management problems. This situation is particularly critical for trucking companies operating national networks because many operations research modeling and solution procedures for such planning and control problems do not scale well when used in practically-sized problems.

In this thesis, we study problems arising at the intersection of fleet sizing, fleet repositioning, and dynamic planning under uncertainty for mobile transportation resources, defined as follows:

- *Fleet repositioning.* Almost all trucking carriers serve sets of loaded requests that are imbalanced in both time and space. Some customer regions are typically net resource attractors while others are net resource generators. Due to such imbalances, carriers need to move resources empty (*i.e.*, without serving a loaded request) between terminals. Planning and executing cost-effective empty repositioning moves remains a primary challenge in trucking operations.

- *Fleet sizing.* The size of a fleet is the number of resources available to cover the required loaded transportation movements. In general, small fleet sizes are desired, but larger fleet sizes might be justified to reduce the need for empty repositioning or as a hedge against uncertainty in operational conditions such as future demand patterns and resource breakdowns.

- *Dynamic planning under uncertainty.* Dynamic decision making involves the following key features: a series of actions must be taken at different points in time to control and optimize the performance of a dynamic and stochastic system; the actions are interdependent (i.e., later decisions depend on earlier actions); and information is revealed over time (i.e., only partial information is known when decisions are made). In the particular context of trucking operations, monthly, weekly, or even daily fleet management plans are constructed based on information about customer locations and demand quantities and timing, which are all uncertain to some degree before the actual execution. Thus, effective fleet management planning must appropriately account for future operational uncertainty to develop dynamic plans that hedge against adverse impacts on the plans and avoid operational disruptions.

This dissertation proposes approaches that enable effective planning and control of mobile transportation resources in large-scale consolidation networks. We develop models, algorithms, and methodologies that are applied to fleet sizing and fleet repositioning. Three specific but interrelated problems are studied in this dissertation. The first two relate to the trade-offs between fleet size and repositioning costs in transportation resource management, while the third involves a dynamic empty repositioning problem with explicit consideration of the uncertainty of future requirements that will be revealed over time. Additionally, the first two problems involve tactical decision making and are addressed using static deterministic network-based optimization models, while the third problem involves dynamic operational decision making in which information is revealed over time and is addressed with a framework that explores how to properly deploy robust optimization models for dynamic planning problems. An important emphasis of this dissertation is in understanding the limitations of existing and proposed models to solve these problems, as well as the refinements that solution techniques need to address these shortcomings.

At the intersection of fleet sizing and fleet repositioning, we study the trade-offs between tractor fleet sizing and repositioning costs. In trucking operations, tractors are costly resources; therefore, reducing the required tractor fleet size can have an important impact on profits. Since consolidation carriers often face demand with different patterns over the course of a week or during the weeks of a month, and because tractors have relatively few operating constraints, an interesting question is what savings are possible from adding extra repositioning moves that deploy tractors to different parts of the network at different times based on need. We developed two sets of variants of tactical network flow optimization models using event-based, time-expanded networks to investigate the tradeoffs between a larger fleet of tractors and the cost of tractor repositioning under different strategies. The first variant set includes models that combine fleet costs and repositioning costs into a single objective function, including some with nonlinear objectives. The second variant set uses a bi-criteria optimization framework and includes models that efficiently compute all the points on the Pareto frontier of fleet size versus repositioning costs.

At the intersection of fleet repositioning and dynamic planning under uncertainty, we study dynamic empty-trailer repositioning. Dynamic empty repositioning plans involve the following characteristics: (1) repositioning decisions are updated over time (daily, weekly, etc.), and move mobile resources from terminals which are net loaded resource attractors to terminals that are net loaded resource generators; (2) at each decision epoch, the number of empty resources available for repositioning and in transit depends on prior repositioning decisions as well as uncertain demand for loaded resources; and (3) uncertain demands for loaded resources, and thus net resource requirements at different terminals, are revealed over time. Developing dynamic empty repositioning plans remains a major challenge for trucking transportation providers operating very-large-scale consolidation networks because only partial future trailer requirements are known. Although prior research has proposed robust optimization models for empty repositioning problems under uncertainty, the existing models do not scale well as the terminal network size grows. Furthermore, prior research has not thoroughly addressed how to properly deploy robust optimization models in multi-stage optimization problem settings such as those encountered in dynamic repositioning. This study proposes approaches for embedding two-stage robust optimization models within a rolling horizon framework for dynamic empty repositioning, and demonstrates that such an approach can be used to create cost-effective, deployable repositioning plans in very large-scale freight transport networks.

1.3 Dissertation Outline and Contributions

We conclude this introductory chapter by outlining the thesis, and describing its specific primary contributions.

Chapter 2 proposes an optimization modeling approach for analyzing the trade-off between the cost of a larger fleet of tractors and the cost of repositioning tractors for a trucking company operating a consolidation network, such as a less-than-truckload (LTL) company. Specifically, we analyze the value of using extra tractor repositioning moves (in addition to the ones required to balance resources throughout the network) to attain savings in the fixed costs of owning or leasing a tractor fleet during a planning horizon. We develop network flow optimization models, some with side constraints and nonlinear objective functions, using event-based, time-expanded networks to determine appropriate fleet sizes and extra repositioning moves under different repositioning strategies with different degrees of implementation flexibility, and we compare the optimal costs of the strategies. For repositioning costs, two different cost schemes are explored: one linear and one nonlinear. Computational experiments using real data from a national LTL carrier compare the total system costs obtained with four different strategies and show that extra repositioning may indeed enable fleet size reductions and concomitant cost savings up to 5%. The primary contributions of the research in this chapter are that (1) we develop the first optimization models that explore the impact of fleet size reductions via repositioning strategies that have regularity and repeatability properties, and (2) we demonstrate that substantial savings in operational costs can be achieved by repositioning tractors in anticipation of regional changes in freight demand.

Chapter 3 studies the optimal Pareto frontiers between the fleet size and repositioning costs of resources required to perform a fixed aperiodic or periodic schedule of transportation requests. We model resource schedules in two alternative ways: as flows on event-based, time-expanded networks; and as perfect matchings on bipartite networks. For aperiodic schedules, all of the Pareto points can be computed in polynomial time solving linear programming formulations of flows on the time-expanded networks or solving minimum weight perfect matching problems on the bipartite networks. Furthermore, adjacent Pareto points can be computed efficiently by solving a single shortest path problem in either type of network. Aperiodic schedules are more difficult. The end points on the frontier can still be computed in polynomial time by solving a sequence of two linear programs and the rest of the points on the frontier can be computed using either integer programming flow formulations or perfect matchings with additional side constraints. Computational experiments using real data from a national less-than-truckload (LTL) carrier compare both the practical applicability of the two proposed modeling frameworks and the computation time to find all the points on the frontier. The main contributions from this chapter are that (1) we develop an efficient re-optimization procedure to compute adjacent

Pareto points that significantly reduces the time to compute the entire Pareto frontier of fleet size versus repositioning costs in aperiodic networks, (2) we show that the natural extension to compute adjacent Pareto points in periodic networks does not work in general as it may increase the fleet size by more than one unit, and (3) we demonstrate that the perfect matching modeling framework is frequently intractable for large-scale instances.

Finally, Chapter 4 considers robust models for dynamic empty-trailer repositioning problems in very large-scale consolidation networks. We investigate approaches that deploy two-stage robust optimization models in a rolling horizon framework to address a multistage dynamic empty repositioning problem in which information is revealed over time. Using real data from a national package/parcel express carrier, we develop and use a simulation to evaluate the performance of repositioning plans in terms of unmet loaded requests and execution costs, and show that the plans generated with our proposed approaches can reduce the unmet loaded requests up to 80% with a modest increase of 8% in execution costs compared to plans generated by deterministic optimization models. Additionally, we provide computational evidence supporting that (1) robust optimization models can use shorter planning horizons to obtain the same or better quality decisions than those obtained with pure deterministic models, and (2) robust optimization models designed explicitly to be embedded within rolling horizon implementations can use less conservative uncertainty estimates than robust optimization models which ignore this key implementation idea. These conclusions are important because robust optimization models are more difficult to solve than deterministic models, and they often do not scale well for large-scale systems over the same planning horizon as that of a deterministic model. Therefore, reducing the size of robust models (via a reduction of the planning horizon or via a simplification of the uncertainty sets against which protection is sought) provides a mechanism to increase the number of problem settings for which these approaches are tractable.

The main contributions from this chapter are that (1) we develop approaches for embedding two-stage robust optimization models within a rolling horizon framework for dynamic empty repositioning, (2) we demonstrate that such approaches enable the solution of very large-scale instances, and (3) we show that less conservative implementations of robust optimization models are required within rolling horizon frameworks.

CHAPTER II

BALANCING FLEET SIZE AND REPOSITIONING COSTS IN LTL TRUCKING

Trucking companies operating consolidation networks, such as less-than-truckload (LTL) carriers or parcel carriers, use tractors to move individual trailers or short trailer trains between pairs of terminals in the so-called linehaul network. These tractor dispatches not only move loaded trailers packed with customer shipments through the network, but also may move empty trailers or no trailers at all. Although a carrier is likely to serve customer demand that is not balanced over geography and time, tractor dispatches throughout the network are balanced over time such that tractors, trailers, and drivers are returned from locations that are net attractors of freight shipments to locations that are net generators.

Operations research techniques such as mathematical programming and dynamic programming have long been used to help determine tactical and operational resource repositioning plans that correct resource imbalances that naturally arise due to demand imbalance in such systems. This is not the focus of this chapter. Instead, we assume that good resource repositioning plans have been developed to correct demand-related resource imbalances, and focus instead on the required tractor fleet size (and its associated fixed costs) required to execute the operations.

In trucking operations, tractors are costly resources and therefore reducing the required owned or leased fleet size can have an important impact on the bottom line. Furthermore, unlike driver resources that are subject to various government (and sometimes union) work rules, tractors have relatively few operating constraints. Therefore, in this chapter we explore the potential tractor fleet size savings that may arise by adding extra tractor repositioning moves to deploy tractors to different parts of the network at different times based on need. These additional tractor repositioning moves may be executed, for example, by one-way drivers simply driving tractors in deadhead moves, or may be larger groups of tractors moved together with a single tractor pulling a flatbed trailer loaded with additional tractors. Since consolidation carriers often face demand with different patterns over the course of a week, or during the weeks of the month, such additional repositioning moves may be beneficial.

Tractor fleet sizing is a tactical decision problem for trucking carriers. Monthly or quarterly adjustments to the fleet size are appropriate in practice. Therefore, in this chapter we adopt an approach that determines an appropriate fleet size using actual historical dispatch data (loaded and empty) for a recent month (for example, an average-demand or peak-demand month during the time period since the previous fleet size adjustment). Note again that we assume that this historical data already includes necessary empty dispatches to correct imbalances in the demand for loaded resources. We then develop a deterministic optimization model using the historical dispatch data that explicitly models both the costs of carrying additional tractors in the fleet and the costs associated with adding extra repositioning moves that may enable a smaller fleet size.

We use the developed models to investigate the value of executing extra repositioning moves as a means to attain savings in the system-wide costs associated with owning or leasing a tractor fleet during a planning horizon. Savings are realized if the decrease in fleet costs, which results from the reduction in the fleet size required to cover all the scheduled dispatches, offsets the costs of the extra repositioning moves. We study a number of different repositioning strategies and compare the total costs of each to the total system cost incurred when no extra repositioning moves are allowed.

The primary contributions of this chapter are the following:

- This chapter develops a modeling framework to explore the value of using a repositioning strategy to attain savings in tractor fleet costs during a planning horizon;

- This chapter investigates different repositioning strategies with different degrees of implementation flexibility, and determines the total system cost savings that may result from each; and
- This chapter presents computational results comparing repositioning strategies using real data from a national LTL carrier under two different costing schemes, and shows that total system cost savings of up to 5% are achievable.

The rest of the chapter is organized as follows. Section 2.1 presents a review of additional literature related to this problem; Section 2.2 describes the modeling framework and discusses the specific characteristics of our models; in particular, Section 2.2.1 describes the construction of the time-expanded networks used in our models, Section 2.2.2 presents the different repositioning strategies which are analyzed, and Section 2.2.3 discusses the two different costing schemes used to evaluate repositioning strategies, a linear one and a nonlinear one. Finally Section 2.3 presents the results of computational experiments performed using data from a national LTL carrier.

2.1 Related Literature

The subject of equipment fleet sizing has been extensively analyzed in the literature and a large variety of problems have been reported. [56] presents an extensive survey of the fleet sizing problems that have been studied in the literature. It integrates several previously developed classifications for these problems , such as the one introduced by [51] in terms of traffic patterns (one-to-one, one-to-many, or many-to-many) and shipment size (full vehicle loads or partial loads), and the one presented by [15] in terms of type of flows (empty or combined empty and loaded), transportation mode (unimodal or multimodal), fleet homogeneity (homogeneous or heterogeneous), and type of company (freight carriers or industrial firms). Additionally, it also classifies the problems by modeling approach (static or dynamic, deterministic or stochastic, using mathematical programming, simulation, or a combined approach), application environments (container terminals, manufacturing systems, railroad networks, urban/passenger transportation systems, freight transportation systems, or maritime transportation), solution procedure (exact algorithms or approximate algorithms), and number of objectives (single or multiple). In contrast, [37] presents a simpler classification of fleet sizing problems, dividing them only into queueing models and time-space models. The former are typically used for long-term decision making, such as planning fleet sizes for several years; they are analytically tractable and rely on aggregate data. The latter are used for short- and medium-term decision making; they include a detailed representation of the underlying system but are also more complex to solve.

The topic of operational repositioning of empty equipment has also received extensive attention by the research community. [21] provides a review of the existing literature in this area, including approaches based on both deterministic and stochastic models. Deterministic models are typically linear network flow models [31, 34, 54] or else relatively easy-to-solve integer programming extensions that result from the addition of various side constraints [1, 20]. Stochastic models are typically large scale dynamic programming models [39, 40, 11], and are solved by a variety of approximation techniques.

A third branch of research has analyzed simultaneous fleet sizing and repositioning decisions. [24, 25] report models and solution methods to minimize the total fleet size and deadheading costs for a large trucking company and a large bus company operating in a metropolitan area. Both cases assume linear costs and incorporate side operational constraints. [7] considers a combined fleet sizing and vehicle allocation problem under dynamic and uncertain conditions. [17] focuses its study of fleet sizing and empty equipment redistribution on hub-and-spoke networks with a single hub, and apply inventory control models and queueing theory to develop decentralized stock control policies for empty equipment based on stockout probabilities.

Closely related to our work in this chapter, fleet sizing and repositioning problems have been addressed for freight trucking operations. In particular, [7] considers a combined fleet sizing and vehicle allocation problem under dynamic and uncertain conditions. They propose a stochastic programming model and a network approximation, and develop a solution procedure which is illustrated on hypothetical problems. Since the repositioning opportunities are not restricted and the uncertainty in travel times is explicitly modeled, their models are intractable and approximate solutions are sought. In contrast, we restrict the repositioning opportunities so that the resulting models can be solved, and the resulting repositioning moves can be implemented in practice. In another related work, [17] also presents a model to simultaneously study fleet sizing and empty equipment redistribution. They focus the analysis on hub-and-spoke networks with a single hub, and apply inventory control models and queueing theory to develop decentralized stock control policies for empty equipment based on stockout probabilities. In contrast, in this chapter we do not restrict the analysis to simple network structures since many real world companies operate complex networks; however, since we use detailed models we do not provide closed form results or simple control policies.

Also related to our research, but in a different transportation environment, [49] investigates and compares static and dynamic models for determining rail-car fleet sizes under different repositioning scenarios. Their proposed dynamic model is based on a time-space network representation and is solved by a decomposition algorithm that exploits the acyclic nature of the network. There are a few notable differences between their time-space network and our time-space network; theirs has a regular discretization with a granularity of a day, whereas ours has a non-regular discretization (as it is event-based with nodes only at the start and end times of scheduled loaded moves and potential repositioning moves) with a time accuracy of a minute. Furthermore, our time-space network wraps around as we assume repeating loaded demands. In addition, the resulting networks differ substantially in size; [49] handles networks of up to 27,000 nodes and 330,000 arcs whereas we deal with networks of up to 442,126 nodes and 824,890 arcs.

Additionally, [9] also studies fleet sizing and repositioning but for a multi-terminal urban bus transportation system. They develop a two-phase approach based on a so-called "deficit function" to reduce the total bus fleet required to meet a fixed scheduled of trips by inserting deadheading trips. They also prove a lower bound on the minimum fleet size that can be attained by exploiting all possible deadheading opportunities. This lower bound was later improved by [50]. [8] extended this work further by permitting variable departure times along with the possible insertions of deadheading trips.

Our research differs from all of this work because the repositioning strategies that we consider in this chapter are not the same as those typically pursued in the research literature that seek to redistribute resources to correct imbalances in loaded demand. We consider repositioning strategies that seek to exploit differences in the timing of loaded demand to reduce the fleet size. Additionally, we address a different application environment with a different repositioning cost structure, we allow only a restricted set of repositioning opportunities to facilitate the actual implementation of such moves in practice, and we explicitly model the trade-off between the reduction of fleet costs and the increase in repositioning costs.

2.2 Modeling Framework

As mentioned before, tractor fleet sizing is a tactical decision problem for trucking carriers concerned with determining the number of tractors required to ensure that trailer dispatches can be executed; both loaded trailer dispatches and empty trailer dispatches which correct imbalances in freight flows. Our focus in this chapter is exploring whether tractor fleet size savings may arise when extra tractor repositioning moves are added to deploy tractors to different parts of the network at different times based on need. Our analysis uses historical trailer dispatch data.

This leads to the following problem: a trucking company operating a consolidation network must serve a number of scheduled trailer dispatches among its terminals throughout a planning horizon; such dispatches must occur at specific times, which are known with certainty. The company wishes to determine the system-wide number of tractors required to serve all of these requests as well as a plan for extra tractor repositioning moves with the objective of minimizing the fleet sizing costs plus the repositioning costs incurred during the planning horizon. Given that the costs associated with serving the scheduled requests are the same for any feasible solution to this problem, we ignore them throughout our analysis.

2.2.1 Time-expanded Networks

We model this situation using time-expanded networks in which each node represents a specific terminal at a particular point in time and each arc represents the movement or possible movement of tractors between different terminals at different times. We consider different cases for our models. In the base case, in which no repositioning moves are considered, only two types of arcs are involved: demand arcs, representing scheduled trailer dispatches that require a tractor, and inventory arcs, representing the option for tractors to remain idle at a terminal. Figure 2 exemplifies the time expansion of a network consisting of three terminals, and a planning horizon that covers ten periods and involves four scheduled dispatches. Given such a time-expanded network, the deficit-function techniques outlined by [9] can be used to determine the minimum fleet size required to serve all the scheduled requests. Since no repositioning moves are considered, such a solution will also minimize the total costs throughout the planning horizon.

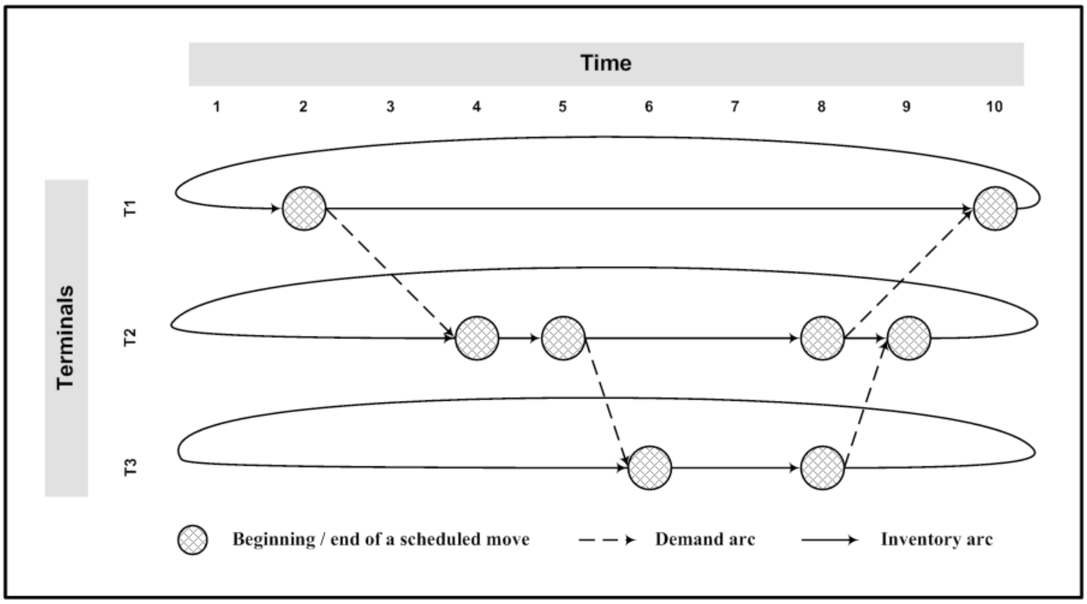


Figure 2: Event-based, time-expanded network without repositioning opportunities

In settings where extra tractor repositioning moves are considered, additional repositioning arcs (and possibly additional nodes) have to be included into the network. This situation is depicted in Figure 3 which shows the same scheduled dispatches as Figure 2 but also includes twelve potential repositioning options (six at time 1 and six at time 6) among the three terminals; the traveling times between terminals T1 and T2, T2 and T3, and T1 and T3 are equal to 1, 2, and 3, respectively. In these settings, deficit-function techniques cannot be used to determine the actions that will minimize the fleet size plus repositioning costs; in fact, these techniques would not even be able to compute the minimum fleet size to cover all the scheduled requests. We address this more complex setting using mathematical programming formulations of flows on the time-expanded networks. The decision variables are the amount of flow on each of the arcs in the network. The flow on the demand arcs corresponds to tractors serving the scheduled trailer movements, the flow on the repositioning arcs characterizes tractors being transported to a different terminal without moving any loads, and the flow on the inventory arcs denotes tractors remaining idle at the corresponding terminal. The constraints for our models are (C1) to satisfy the flow balance equations at each of the nodes, (C2) to meet the demand requirements (i.e., to cover all scheduled requests), and (C3) to honor the flow integrality requirements. Additionally, some repositioning strategies impose certain repeatability patterns on the repositioning moves, such as weekly or bi-weekly repeatability; for such strategies, the flows also have to satisfy the desired repeatability patterns (C4). Finally, the objective function has two components: (1) a component capturing the cost of operating the tractor fleet during the planning horizon (assumed to be a linear function of the number of tractors required in the network), and (2) a component capturing the cost of the extra tractor repositioning moves (which will be discussed in detail in Section 2.2.3).

Figure 3 reveals several other relevant aspects of our models and their input data. The input data have the following characteristics:

- The scheduled trailer moves represent complete tours: The total number of scheduled trailer moves out of a terminal equals the total number of scheduled trailer moves into that terminal. This ensures that the model is feasible even without any extra tractor repositioning moves, which is desirable because this constitutes the base case against which tractor repositioning strategies are compared.

- There is a limited set of extra tractor repositioning moves: Only a subset of potential extra tractor repositioning opportunities are included in the model. This is necessary to ensure computational tractability, but also ensures practical

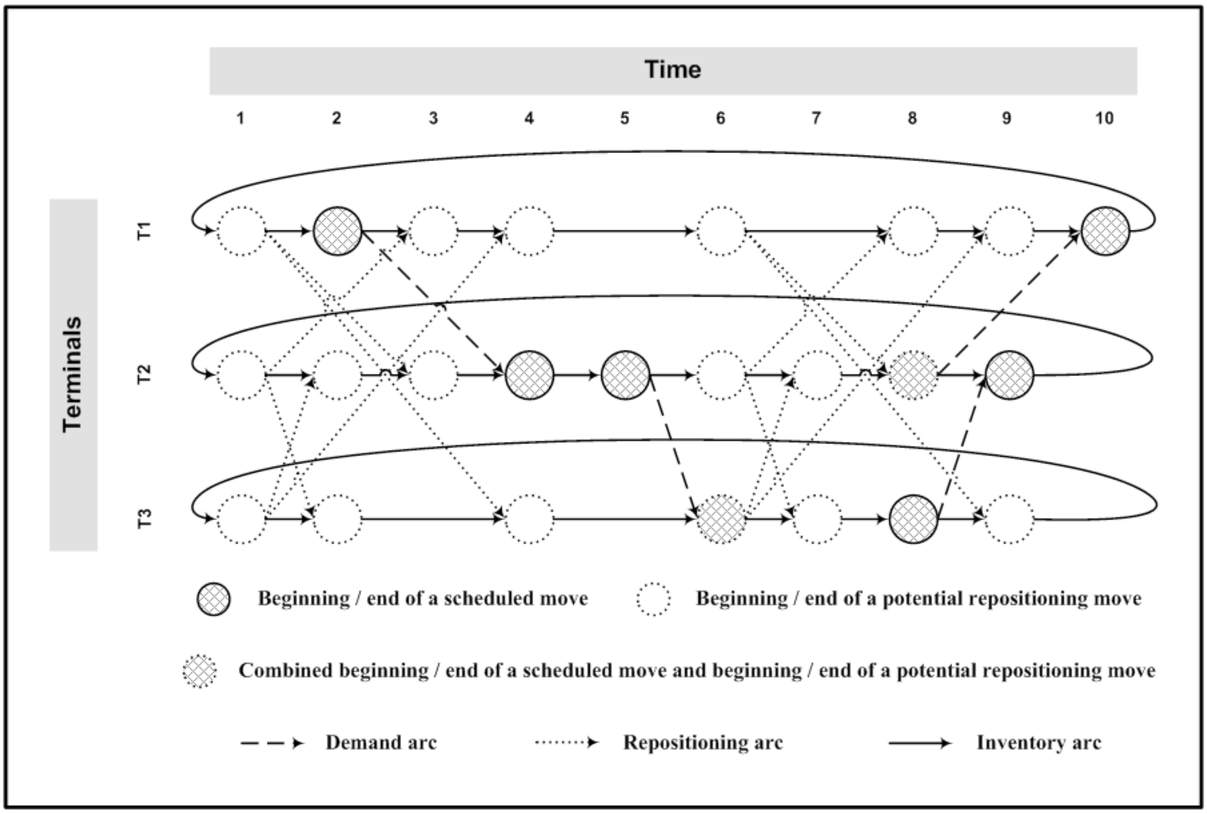

Time
1
2
3
4
5
6
7
8
9
10
Terminals
T1
T2
T3
Beginning / end of a scheduled move
Beginning / end of a potential repositioning move
Combined beginning / end of a scheduled move and beginning / end of a potential repositioning move
Demand arc
Repositioning arc
Inventory arc

Figure 3: Event-based, time-expanded network with repositioning opportunities

solutions, (i.e., repositioning plans that can actually be implemented).

The models have the following characteristics:

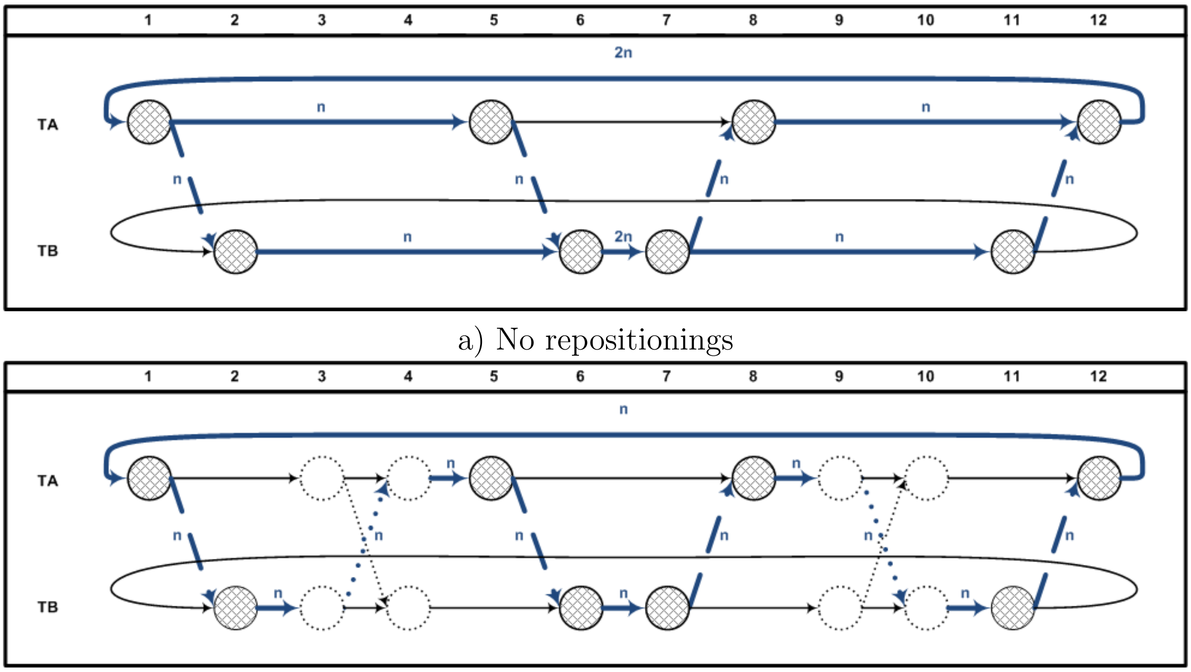
- They are wrap-around models: The arcs corresponding to moves that begin during the planning horizon but are completed after the end of the horizon are wrapped around and connected to earlier periods of the horizon (Figure 1 shows only wrap-around inventory arcs, but demand arcs and repositioning arcs can also wrap around). This characteristic is desirable to prevent warm-up or cool-down effects at the beginning or end of the planning horizon.

- They are circulation models: The networks do not have external supplies or demands of tractors. This property implies that the total number of tractors in the time-expanded network at any point in time remains constant throughout the entire planning horizon.

- They are event-based models: The only events that need to be represented by nodes are the beginning or end of scheduled trailer moves and potential extra tractor repositioning moves. This ensures that the resulting networks do not get too big.

This modeling framework is intended to identify time patterns in the scheduled moves that could be exploited by extra repositioning moves to reduce the required fleet size of a company. For example, consider the following idealized situation: A company operates only two terminals, A and B, and during a planning horizon of twelve periods, n scheduled trailer dispatches from A to B take place at periods 1 and 5, and n scheduled trailer dispatches from B to A take place at periods 7 and 11 (it takes one period to move between the terminals in either direction). Figure 4 illustrates this situation. The top part shows the base case where no repositioning opportunities are available and the bottom part shows the case where tractors can be repositioned between the terminals at periods 3 and 9. For both cases, we show the time-expanded network with the minimum required tractor fleet size.

In this specific situation the fleet size is reduced by 50% (from $\mathbf{2n}$ to $\mathbf{n}$) by exploiting extra tractor repositioning opportunities. The specific time patterns that make this reduction possible are described next: At terminal B, between the arrival of n tractors at period 2 and the departure of n tractors at period 7, there is enough time to move the tractors to terminal A and back. Likewise at terminal A, between the arrival of n tractors at period 8 and the departure of n tractors at period 1 of the next cycle, there is enough time to move the tractors to terminal B and back. In general, due to the interactions between the scheduled moves among all the terminals, spotting these types of time patterns in a large network is not trivial. Moreover, note that if repeatability conditions were imposed on the repositioning, for example if the same repositioning moves had to be performed in period 9 as in period 3, then the fleet size could not be reduced.



a) No repositionings

b) Repositioning moves are allowed

Figure 4: Example of fleet-size reduction due to repositioning

2.2.2 Tractor Repositioning Strategies

Clearly, there exists a trade-off between the flexibility of a repositioning strategy and the reduction in fleet size it can attain. In general, a flexible repositioning strategy tends to result in large fleet size reductions, but the actual implementation of such a strategy may be challenging. On the other hand, a structured repositioning strategy, in which the repositioning moves obey certain regularity and repeatability patterns, tends to be easier to implement and monitor, but may only attain small fleet size reductions. To explore these trade-offs, we analyze four different repositioning strategies:

1. **2Rep**: Performing at most two repositioning moves per day, with fixed departure times.

2. **BiWkly**: Performing at most two repositioning moves per day, with fixed departure times and biweekly repeatability.

3. **Wknd**: Performing at most two repositioning moves per day, with fixed departure times, only on weekends.

4. **Wkly**: Performing at most two repositioning moves per day, with fixed departure times, and weekly repeatability.

These repositioning strategies will result in different fleet size reductions. However, if the cost savings from the decrease in fleet size offset the costs of the extra tractor repositioning moves, savings will be attained. The next section describes how these costs are incorporated into our models to analyze this trade-off.

2.2.3 Tractor Repositioning Costs

The simplest cost structure to trade off tractor repositioning costs and fleet sizing costs uses linear functions for both types of costs:

- Operating and maintaining a tractor throughout the planning horizon costs C^m.

- Repositioning a tractor costs C^ℓ per mile traveled.

By assigning each extra tractor repositioning arc a cost of C^ℓ times the length in miles from the terminal of origin to the terminal of destination of the corresponding tractor repositioning move, the repositioning costs can be retrieved by multiplying the flow along the tractor repositioning arcs with the cost assigned to these arcs. Because we have a circulation model, the costs of operating and maintaining the tractor fleet is equal to C^m times the total number of tractors circulating in the time-expanded network at some point in time. To quantify the number of tractors circulating in the network, we use the concept of a temporal cut.

A *temporal cut at* t is defined as the subset of arcs in the time-expanded network corresponding to events (i.e., scheduled moves, repositioning moves, and idle tractors) that start on or before time t and are completed after time t, where t can take any value within the planning horizon. Figure 4 illustrates this concept; the arcs that belong to the temporal cut at $t = 3.5$ are shown in bold. A similar concept called a *count time-line* is used for counting aircraft in airline fleet assignment models defined over time-space networks. In those models, the flows associated with a specific type of aircraft on all the arcs that cross a chosen count time-line are summed to enforce that the total number of assigned aircraft does not exceed the number of available aircraft of that particular type (cf., [48] for an extensive review of concepts, models, and algorithms of airline fleet assignment models). In our case, by assigning a cost of C^m to the arcs belonging to a temporal cut we can quantify the total fleet size costs. Finally, the total cost of a repositioning strategy is simply the sum of its repositioning costs plus its fleet size costs.

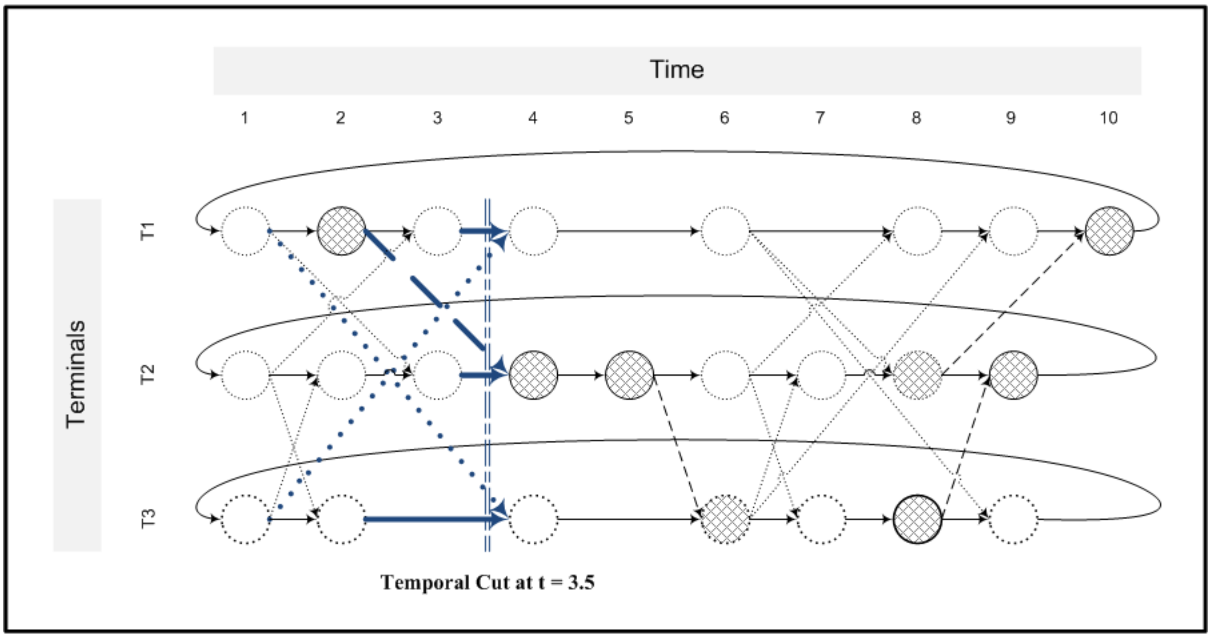


Figure 5: Arcs belonging to a temporal cut

With this cost structure, the models for repositioning strategies that do not have repeatability requirements correspond to minimum cost network flow models and are thus easily solved. On the other hand, the models for repositioning strategies that have repeatability requirements involve additional side constraints and are thus more general and have to be solved using integer programming solvers, which may lead to larger solution times. Nevertheless, in all of our computational experiments, optimal solutions were found quickly.

Next, we present a more realistic costing scheme to evaluate the trade-off between fleet size costs and repositioning costs. For fleet size costs, we maintain the same set up, that is, it costs C^m to operate and maintain a tractor throughout the planning horizon. However, for the tractor repositioning moves we introduce a more complex, nonlinear cost structure: Repositioned tractors are sent in batches (up to a maximum batch size). They exhibit a nonlinear cost structure in which the first tractor of a batch incurs a cost of C_1^n per mile traveled, and each of the remaining tractors on the batch accrues a cost of C_2^n per mile traveled (we assume $C_2^n < C_1^n$ to prevent cost unboundedness). The motivation behind this nonlinear cost structure has to do with the way in which tractors would actually be transported between terminals. Tractors are either repositioned individually by one-way drivers driving tractors in deadhead moves, or repositioned in groups with a single tractor pulling a flatbed trailer loaded with additional tractors. This nonlinear repositioning function is neither concave nor convex, but it can be bounded below by the linear function corresponding to the average per-mile repositioning cost of a tractor on a full batch ($\frac{C_1^n + C_2^n(S_B^{max}-1)}{S_B^{max}}$, where S_B^{max} is the maximum batch size). Figure 6 presents a graph that exemplifies the nonlinear repositioning costs per mile corresponding to the parameters $C_1^n = 2, C_2^n = 0.6$, and $S_B^{max} = 4$. In this case, the lower bound corresponds to a per-mile linear repositioning cost function with parameter $C^\ell = 0.95$. Finally, the total repositioning costs can be compactly stated as $\sum_{a \in \mathcal{A}^R} D_a \left(C_1^n \left\lceil \frac{x_a}{S_B^{max}} \right\rceil + C_2^n \left(x_a - \left\lceil \frac{x_a}{S_B^{max}} \right\rceil \right) \right)$, where $\mathcal{A}^R$ is the set of repositioning arcs, D_a is the distance (in miles) associated with arc a, x_a is the flow of tractors on arc a, S_B^{max} is the maximum batch size, and $\lceil \cdot \rceil$ is the ceiling function.

Although the resulting models are nonlinear programming problems, we can reformulate them as mixed integer linear programs using additional integer variables for

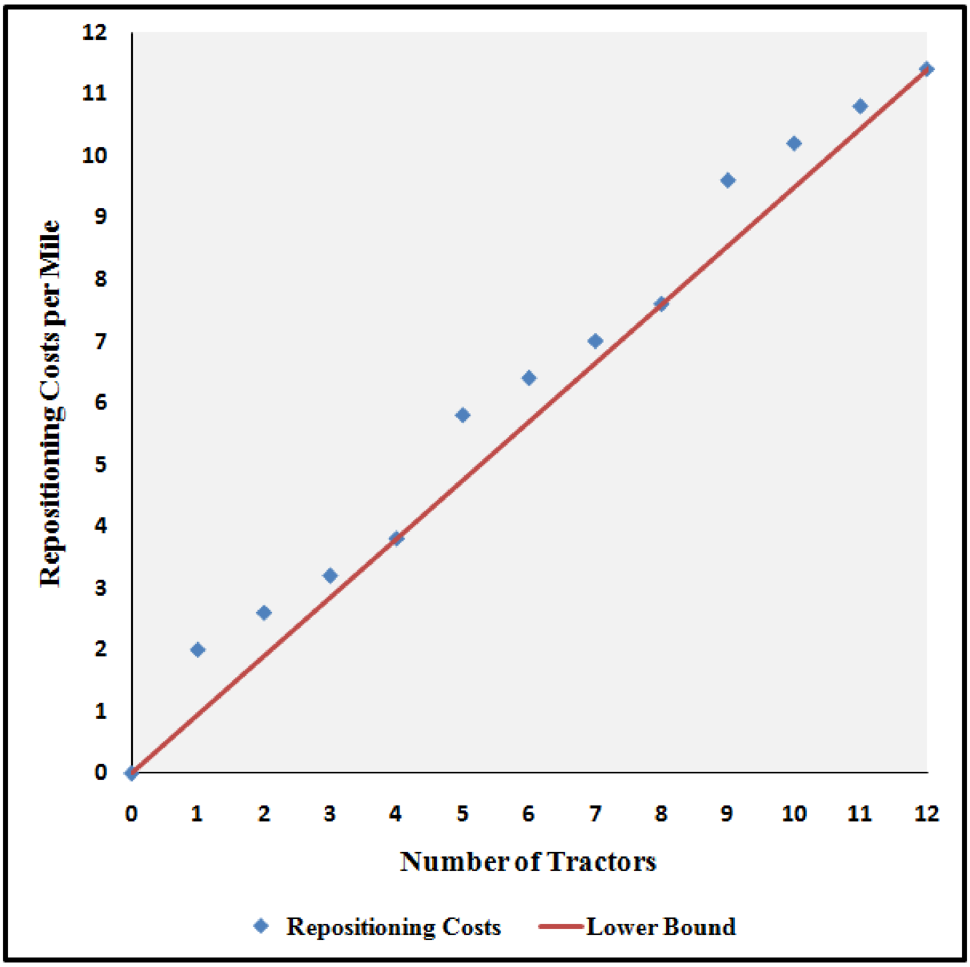


Figure 6: Nonlinear cost structure for repositioning moves

the number of batches sent on each of the repositioning arcs and additional constraints to relate the number of batches sent to the number of tractors sent on each of the repositioning arcs. Using the additional variables, the total repositioning costs can be expressed as $\sum_{a \in \mathcal{A}^R} D_a \left(C_1^n y_a + C_2^n \left(x_a - y_a\right)\right) = \sum_{a \in \mathcal{A}^R} D_a \left(C_2^n x_a + \left(C_1^n - C_2^n\right) y_a\right)$, where y_a is the number of batches sent on arc a. Furthermore, the additional constraints state that the number of batches on a given repositioning arc must be greater than or equal to the number of tractors sent on that arc divided by the maximum batch size (i.e., $y_a \geq \frac{x_a}{S_B^{max}}, \forall a \in \mathcal{A}^R$). The complete mathematical formulations of all the models introduced in this section are shown in the chapter appendix.

2.3 Computational Results

In order to evaluate the cost savings that can result from different repositioning strategies, we performed computational experiments using historical dispatch data from a national LTL carrier that operates 346 terminals. The historical dispatches span four weeks of operations, in which 115,140 scheduled trailer dispatches were performed. The carrier imposed the following conditions on tractor repositioning moves:

1. Tractor repositioning moves can only occur between terminals that act as domicile for tours (in the historical data 135 out of the 350 terminals meet this requirement).

2. No tractor repositioning move can take longer than 11 hours. This condition was imposed to meet Hours-of-Service regulations in case tractor repositioning moves are performed by the carrier itself. This restriction might be unnecessary when tractor repositioning moves are outsourced.

3. Tractor repositioning moves are restricted to at most 2 per day with fixed departure times.

Table 1 summarizes the characteristics of the potential repositioning moves associated with the four tractor repositioning strategies being evaluated when the above conditions are taken into account.

Table 1: Characteristics of the tractor repositioning strategies

Strategy	Max duration	# Moves per day	Start times	Frequency	Repeatability	# Potential moves
2Rep	11 hr	2	7:00 , 19:00	Daily	None	267,624
BiWkly	11 hr	2	7:00 , 19:00	Daily	Bi-weekly	267,624
Wknd	11 hr	2	7:00 , 19:00	On weekends	None	76,464
Wkly	11 hr	2	7:00 , 19:00	Daily	Weekly	267,624

The carrier provided us with accurate estimates of the cost of operating and maintaining a tractor as well as the cost of repositioning a tractor for both costing schemes. For confidentiality reasons these cost estimates are scaled when presenting results.

All of our models were implemented using ILOG OPLStudio 6.010 which calls CPLEX 11.110 as solver. The models for strategies **2Rep** and **Wknd** correspond to minimum cost network flow problems and can therefore be easily solved via linear programming. On the other hand, the models for strategies **BiWkly** and **Wkly** require side constraints to enforce the repeatability patterns; nevertheless, in our computational experiments the resulting integer programs all solved at the root node of the branch and bound tree without adding any cutting planes to the formulation, i.e., the linear programming relaxations happen to yield integral flows. A similar situation was observed in [20], where it is reported that instances of a multi-commodity network flow problem on a time-expanded network modeling the repositioning of empty containers solve quickly. Table 2 shows the results obtained under the linear costing scheme for each of the repositioning strategies. It also includes information about the size of the models and the time required to solve them. The total costs reported correspond to the costs associated with the optimal solution for a given strategy. The numbers in brackets represent the changes of fleet size costs and total costs of the given strategy with respect to those costs when there is no tractor repositioning. As expected, the results show that the more restricted tractor repositioning options yield smaller cost savings. Repeatability of the repositioning moves is desirable because it facilitates the planning and execution of such moves, but it is also costly; in fact, the cost savings are reduced by approximately $\frac{2}{3}$ when biweekly repeatability is imposed and the cost savings are reduced to practically zero when weekly repeatability is imposed. Limiting tractor repositioning moves to weekends (without repeatability restrictions) provides an appealing compromise because tractor repositioning moves

Table 2: Computational results using linear repositioning costs

Strategy	Metrics	Results	Additional Info	
2Rep	Fleet size (tractors)	2,305	# Variables	709,750
	# Repositionings	423	# Constraints	442,126
	Weighted repositionings (tractor-minutes)	68,628	Solution time	63 seconds
	Fleet costs (\$)	2,714,195 [-7.32%]		
	Repositioning costs (\$)	102,625		
	Total costs (\$)	2,816,847 [-3.81%]		
BiWkly	Fleet size (tractors)	2,413	# Variables	709,750
	# Repositionings	270	# Constraints	575,938
	Weighted repositionings (tractor-minutes)	32,952	Solution time	64 seconds
	Fleet costs (\$)	2,841,368 [-2.98%]		
	Repositioning costs (\$)	49,289		
	Total costs (\$)	2,890,657 [-1.29%]		
Wknd	Fleet size (tractors)	2,397	# Variables	518,590
	# Repositionings	181	# Constraints	442,126
	Weighted repositionings (tractor-minutes)	33,084	Solution time	59 seconds
	Fleet costs (\$)	2,822,527 [-3.62%]		
	Repositioning costs (\$)	49,486		
	Total costs (\$)	2,872,014 [-1.93%]		
Wkly	Fleet size (tractors)	2,470	# Variables	709,750
	# Repositionings	132	# Constraints	642,844
	Weighted repositionings (tractor-minutes)	8,372	Solution time	69 seconds
	Fleet costs (\$)	2,908,487 [-0.68%]		
	Repositioning costs (\$)	12,523		
	Total costs (\$)	2,921,439 [-0.26%]		
No Rep	Fleet size (tractors)	2,487	# Variables	187,589
	# Repositionings	-	# Constraints	187,589
	Weighted repositionings (tractor-minutes)	-	Solution time	24 seconds
	Fleet costs (\$)	2,928,505		
	Repositioning costs (\$)	-		
	Total costs (\$)	2,928,505		

take place only on the least busy days of the week, and substantial cost savings remain.

With the nonlinear costing scheme, the resulting mixed integer programs cannot be solved to proven optimality in 24 hours of computation time for any of the strategies evaluated. However, the optimality gap after 24 hours is quite small, less than 0.52% in all cases. Table 3 presents the results for each of the tractor repositioning strategies. We report two solutions. The first solution reported (*Adjusted LC Solution*) results from solving the model with linear repositioning costs with cost parameter equal to the average cost of a tractor in a full batch (i.e., $C^{\ell} = \frac{C_1^n + (S_B^{max}-1)C_2^n}{S_B^{max}}$), and setting $y_a = \left\lceil \frac{x_a}{S_B^{max}} \right\rceil$, $\forall a \in \mathcal{A}^R$. The second solution reported (*MIP Solution*) is the best solution to the mixed integer program found in 24 hours of computation time. For information purposes, we also report the size of the models solved as well as statistics related to the solution process. *LP relax value* corresponds to the objective function value of the LP relaxation, *Time to best* is the time elapsed until the solver found the best solution, and *Optimality gap* is the difference between the values of the best solution and the best lower bound (after 24 hours of computation) as a percentage of the value of the best solution. The results do not differ substantially from our previous experiment. Requiring repeatability of the tractor repositioning moves is costly, and restricting tractor repositioning may provide an acceptable compromise. It is worth observing that the Adjusted LC solutions are quite close to the best solutions found when minimizing the nonlinear costing scheme. Hence, they seem to provide a computationally efficient approach for obtaining high-quality solutions very quickly.

Next, we present a few figures that provide more detail and further insights into the characteristics of the different tractor repositioning strategies. Figure 7 shows for each tractor repositioning strategy the system-wide number of idle tractors over time for the planning horizon compared against the idle tractors when no repositioning moves are performed. The left column shows the entire planning horizon, whereas the right column expands the results for the second week. Note that once again the largest benefits (maximum reduction in idle tractors) are attained when no regularity conditions are imposed on the repositioning moves. In addition, note that the system-wide number of idle tractors never equals zero. The reason for this is as follows. In an optimal solution to any of the tractor repositioning models, the number of idle tractors at each terminal must equal zero at one point in time, but these time points may be different for the different terminals. In fact, it is highly unlikely that these time points would be perfectly aligned, which explains why the system-wide number of idle tractors never reaches zero.

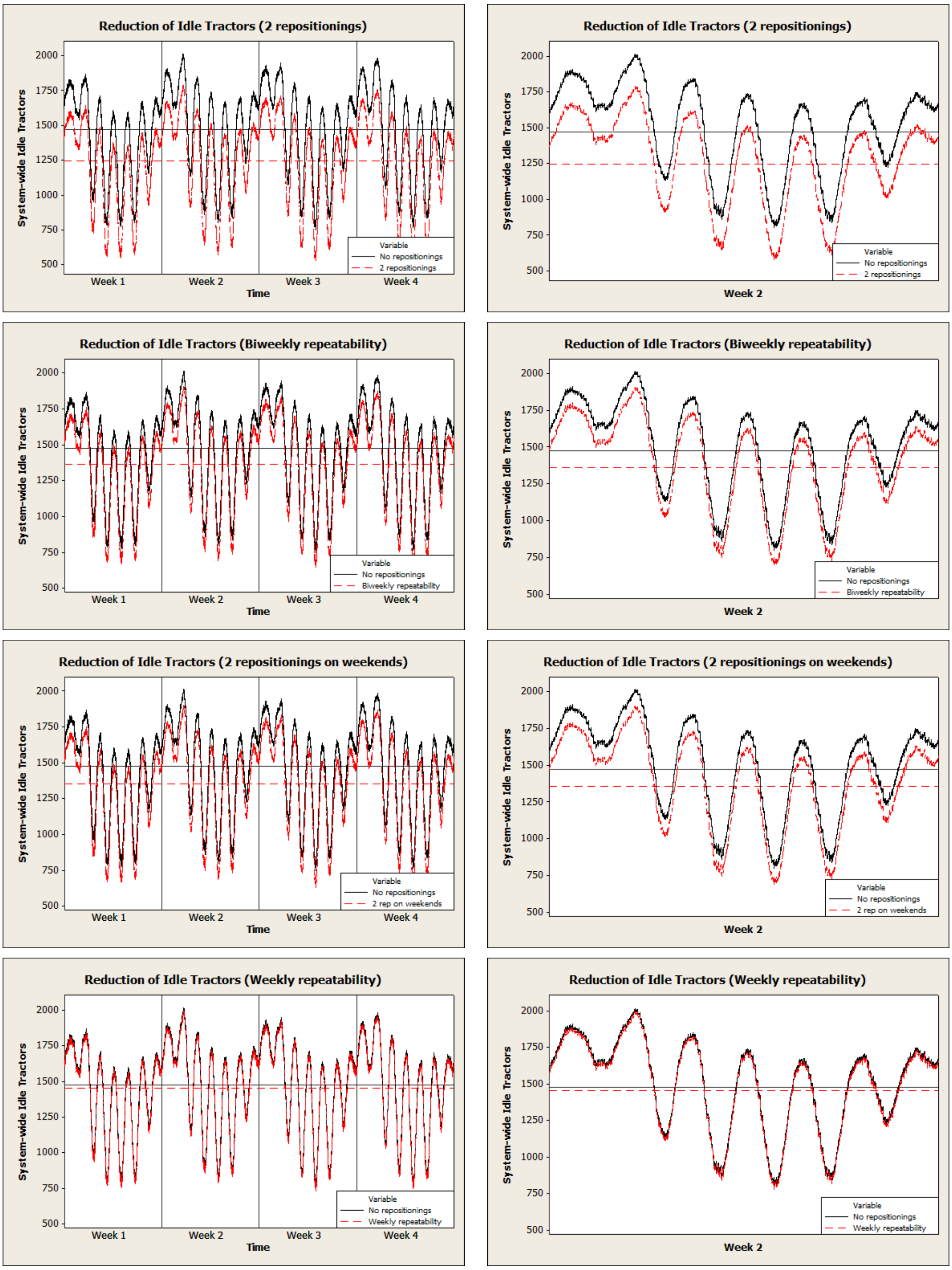


Figure 7: Effect of repositioning strategies on idle tractors

Table 3: Computational results using nonlinear repositioning costs

Strategy	Metrics	Adjusted LC Solution	MIP Solution		
			Solution	Additional Information	
2Rep	Fleet size	2,237	2,262	# Variables	977,374
	# Repositionings	685	675	# Constraints	709,750
	Weighted repositionings	179,186	147,632	LP relax value	2,749,923
	Fleet costs	2,634,123 [-10.05%]	2,663,562 [-9.05%]	Time to best	10 hr
	Repositioning costs	169,643	116,616	Optimality gap	0.52%
	Total costs	2,803,766 [-4.26%]	2,780,177 [-5.06%]		
BiWkly	Fleet size	2,359	2,378	# Variables	977,374
	# Repositionings	522	470	# Constraints	843,562
	Weighted repositionings	115,713	92,697	LP relax value	2,852,561
	Fleet costs	2,777,781 [-5.15%]	2,800,154 [-4.38%]	Time to best	4.5 hr
	Repositioning costs	112,035	74,728	Optimality gap	0.36%
	Total costs	2,889,816 [-1.32%]	2,874,882 [-1.83%]		
Wknd	Fleet size	2,364	2,371	# Variables	595,054
	# Repositionings	295	318	# Constraints	518,590
	Weighted repositionings	88,232	80,540	LP relax value	2,840,689
	Fleet costs	2,783,669 [-4.95%]	2,791,912 [-4.66%]	Time to best	15 hr
	Repositioning costs	80,143	63,365	Optimality gap	0.27%
	Total costs	2,863,812 [-2.21%]	2,855,277 [-2.50%]		
Wkly	Fleet size	2,445	2,467	# Variables	977,374
	# Repositionings	320	172	# Constraints	910,468
	Weighted repositionings	44,479	14,853	LP relax value	2,907,793
	Fleet costs	2,879,049 [-1.69%]	2,904,954 [-0.80%]	Time to best	40 min
	Repositioning costs	50,661	14,671	Optimality gap	0.11%
	Total costs	2,929,710 [-0.04%]	2,919,625 [-0.30%]		
No Rep	Fleet size	2,487			
	# Repositionings	-			
	Weighted repositionings	-			
	Fleet costs	2,928,505			
	Repositioning costs	-			
	Total costs	2,928,505			

The different tractor repositioning strategies affect terminals in the system differently. Figure 8 shows the details of tractor repositioning moves for the different repositioning strategies performed at two terminals in the network (positive numbers represent incoming tractors and negative numbers represent outgoing tractors). We see that Terminal 1 exploits repositioning opportunities in each of the different tractor repositioning strategies whereas Terminal 2 does not. When repeatability requirements are imposed limited repositioning takes place; in fact, when weekly repeatability is required, there is no repositioning at all.

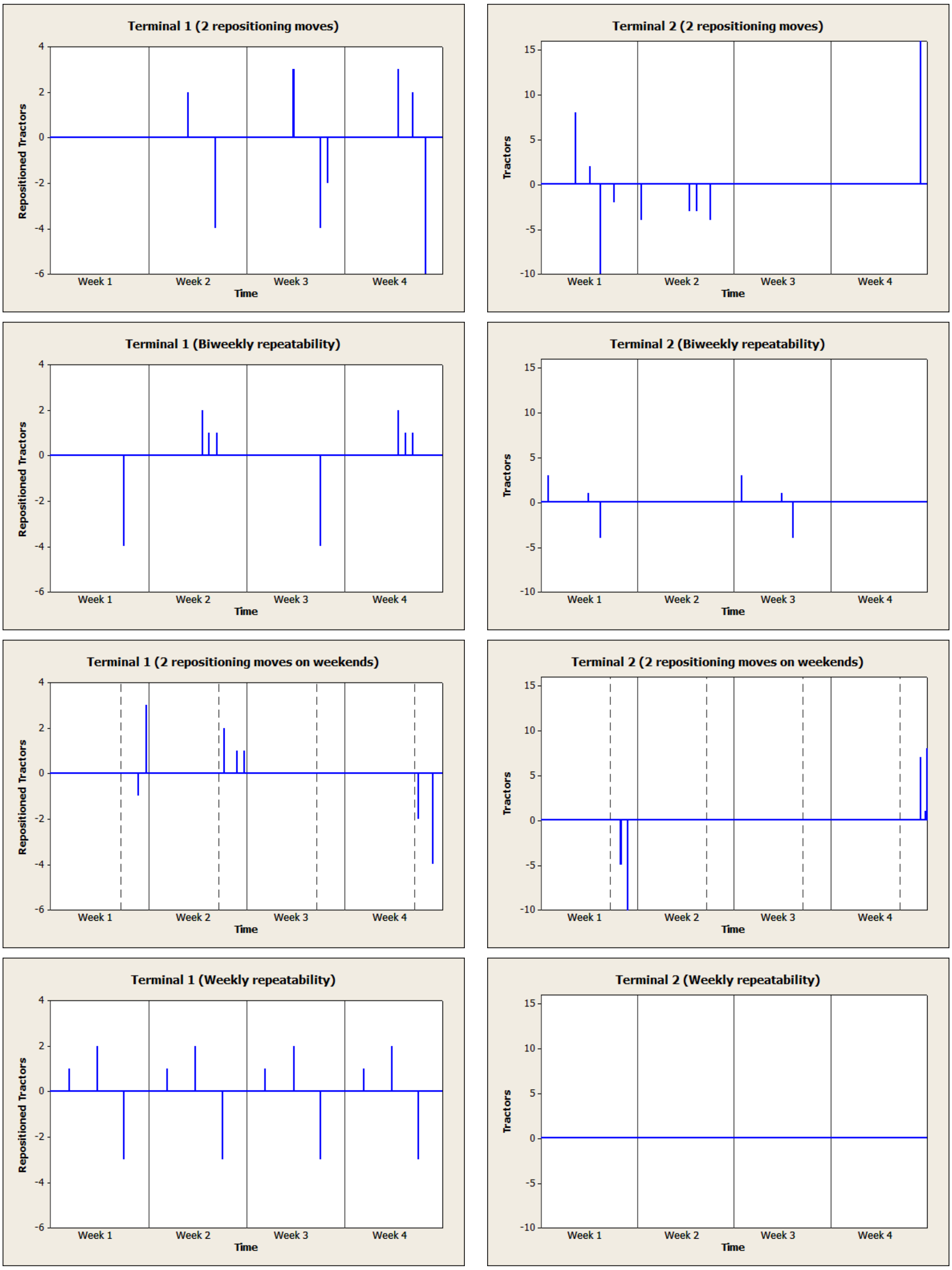


Figure 8: Details of the repositioning moves for two terminals

2.4 Chapter Appendix - Mathematical Formulations

This appendix contains the mathematical formulations corresponding to the models presented in Section 2.2.

Notation

Consider the time-expanded networks whose construction was described in Section 3.1. For a given time-expanded network, let $\mathcal{N}$ be the set of nodes, and $\mathcal{A}$ be the set of arcs such that $\mathcal{A} = \mathcal{A}^I \cup \mathcal{A}^D \cup \mathcal{A}^R$, where $\mathcal{A}^I$ is the set of inventory arcs, $\mathcal{A}^D$ is the set of demand arcs, and $\mathcal{A}^R$ is the set of repositioning arcs. We can refer to a node $n \in \mathcal{N}$ as $n = (L, t)$, where L is the specific terminal and t is the specific time associated with node n; and we can also refer to an arc $a \in \mathcal{A}$ as $a = (L_1, t_1, L_2, t_2)$, where (L_1, t_1) and (L_2, t_2) are respectively the tail and head nodes of arc a. Consider also the definition of the following parameters:

- T = Length of the planning horizon.
- W = Number of time periods in one week.
- C^m = Cost of operating and maintaining a tractor during the entire planning horizon.
- D_a = Distance (in miles) associated with arc a, $\forall a \in \mathcal{A}^R$.
- N_a = Number of tractors scheduled to traverse arc a, $\forall a \in \mathcal{A}^D$.
- C^ℓ = Cost per mile of repositioning a tractor under the linear costing scheme.
- C_1^n = Cost per mile of repositioning the first tractor in a batch under the nonlinear costing scheme.
- C_2^n = Cost per mile of repositioning each tractor in addition to the first one in a batch under the nonlinear costing scheme.

- S_B^{max} = Maximum batch size for repositioning tractors under the nonlinear costing scheme.

We also use the following additional notation:

- $\delta^+(n)$ and $\delta^-(n)$ denote respectively the set of arcs originating at node n and the set of arcs ending at node n, $\forall n \in \mathcal{N}$.
- $\mathcal{T}(t) = \{a = (L_1, t_1, L_2, t_2) \in \mathcal{A} : t_1 \leq t, t_2 > t\}$ is the set of arcs that belong to the temporal cut at t, where $0 \leq t \leq T$.
- $\mathcal{A}_w^R$ denotes the set of repositioning arcs which correspond to repositioning moves performed during weekdays.
- $\mathbb{R}^p$ denotes the p-dimensional real numbers.
- $\mathbb{Z}^+$ is the set of nonnegative integers.

Decision Variables

- x_a = Number of tractors traversing arc a, $\forall a \in \mathcal{A}$.
- y_a = Number of batches of tractors traversing arc a, $\forall a \in \mathcal{A}^R$.

Model without repositioning moves

The model associated with the simplest time-expanded network, which involves only demand and inventory arcs, is simply a minimum cost network flow (MCNF) model:

$$
\begin{array}{lll}
Min & \sum_{a \in \mathcal{T}(t)} C^m x_a & \\
\text{s.t.} & \sum_{a \in \delta^+(n)} x_a - \sum_{a \in \delta^-(n)} x_a = 0 & \forall n \in \mathcal{N} \\
 & x_a = N_a & \forall a \in \mathcal{A}^D \\
 & x_a \geq 0 & \forall a \in \mathcal{A}
\end{array}
$$

Models with linear repositioning costs

Once repositioning opportunities are evaluated through the incorporation of repositioning arcs in the time-expanded networks, the following model results:

$$\begin{array}{lll} Min & \sum_{a\in\mathcal{A}^R}\left(C^{\ell}D_a\right)x_a+\sum_{a\in\mathcal{T}(t)}C^m x_a & \\ \text{s.t.} & \sum_{a\in\delta^+(n)}x_a-\sum_{a\in\delta^-(n)}x_a=0 & \forall n\in\mathcal{N} \\ & x_a=N_a & \forall a\in\mathcal{A}^D \\ & x=\{x_a\}\in X & \\ & x_a\in\mathbb{Z}^+ & \forall a\in\mathcal{A} \end{array}$$

The set $X\subseteq\mathbb{R}^{|\mathcal{A}|}$ comprises the additional repeatability and regularity conditions imposed by some of the repositioning strategies described in Section 3.2. The following table shows the specific definition of X for each of the repositioning strategies:

Strategy	Definition of the set X
2Rep	$X=\mathbb{R}^{\vert\mathcal{A}\vert}$
BiWkly	$X=\{\quad x\in\mathbb{R}^{\vert\mathcal{A}\vert}:$ $x_{a^1}=x_{a^2};\quad \forall a^1=\left(L_1^1,t_1^1,L_2^1,t_2^1\right),a^2=\left(L_1^2,t_1^2,L_2^2,t_2^2\right)\in\mathcal{A}^R\quad s.t.\quad L_1^1=L_1^2,L_2^1=L_2^2$ $t_1^1\leq 2W,t_2^2>2W$ $\left(t_1^2-t_1^1\right)mod\ 2W=0$ $\}$
Wknd	$X=\{x\in\mathbb{R}^{\vert\mathcal{A}\vert}:\quad x_a=0;\quad\forall a\in\mathcal{A}_w^R\}$
Wkly	$X=\{\quad x\in\mathbb{R}^{\vert\mathcal{A}\vert}:$ $x_{a^1}=x_{a^2};\quad \forall a^1=\left(L_1^1,t_1^1,L_2^1,t_2^1\right),a^2=\left(L_1^2,t_1^2,L_2^2,t_2^2\right)\in\mathcal{A}^R\quad s.t.\quad L_1^1=L_1^2,L_2^1=L_2^2$ $t_1^1\leq W,t_2^2>W$ $\left(t_1^2-t_1^1\right)mod\ W=0$ $\}$

Models with nonlinear repositioning costs

The original nonlinear programming model that incorporates the nonlinear repositioning costs is the following:

$$\begin{array}{lll} Min & \sum_{a\in\mathcal{A}^R}D_a\left(C_1^n\left\lceil\frac{x_a}{S_B^{max}}\right\rceil+C_2^n\left(x_a-\left\lceil\frac{x_a}{S_B^{max}}\right\rceil\right)\right)+\sum_{a\in\mathcal{T}(t)}C^m x_a & \\ \text{s.t.} & \sum_{a\in\delta^+(n)}x_a-\sum_{a\in\delta^-(n)}x_a=0 & \forall n\in\mathcal{N} \\ & x_a=N_a & \forall a\in\mathcal{A}^D \\ & x\in X & \\ & x_a\in\mathbb{Z}^+ & \forall a\in\mathcal{A} \end{array}$$

This model is reformulated into the following integer programming problem using the additional variables y_a.

$$\begin{array}{lll}
Min & \sum_{a\in\mathcal{A}^R} D_a \left(C_2^n x_a + \left(C_1^n - C_2^n\right) y_a\right) + \sum_{a\in\mathcal{T}(t)} D x_a & \\
\text{s.t.} & \sum_{a\in\delta^+(n)} x_a - \sum_{a\in\delta^-(n)} x_a = 0 & \forall n \in \mathcal{N} \\
& x_a = N_a & \forall a \in \mathcal{A}^D \\
& y_a \geq \frac{x_a}{S_B^{max}} & \forall a \in \mathcal{A}^R \\
& x \in X & \\
& x_a \in \mathbb{Z}^+ & \forall a \in \mathcal{A} \\
& y_a \in \mathbb{Z}^+ & \forall a \in \mathcal{A}^R
\end{array}$$

The definitions of the set X for the specific repositioning strategies are the same as above.

CHAPTER III

FREIGHT TRANSPORTATION FLEET SIZING WITH REPOSITIONING CONSIDERATIONS

In the previous chapter, a fleet sizing and empty repositioning problem was addressed in which a single cost function was used to balance the savings from reducing the size of a fleet of tractors with the costs of additional tractor repositioning moves. A potential drawback of such an approach is that the value of a fleet of resources of a given size cannot entirely be quantified by just looking at the cost of operating and maintaining it; higher fleet sizes might be justified as a means to hedge against uncertainty in operational conditions such as future demand patterns and resource breakdowns. Carefully accounting for all those sources of variability will most likely lead to stochastic programming or robust optimization models; however, simpler deterministic models with two separate objective functions can still shed some light into the interactions between fleet size and repositioning. As a result, analyzing the efficiency frontier (also called optimal Pareto frontier) between fleet size and repositioning seems to be a practical approach to provide the operator with good information to decide where on the frontier to position itself. The analysis of the efficiency frontier between fleet size and repositioning is the problem addressed in this chapter.

We will simplify our analysis by only considering linear repositioning costs and ignoring any repeatability constraints, but we will generalize our application context to any transportation operator that manages a fleet of homogeneous resources such as containers, trucks, rail cars, aircraft, or buses, and uses them to fulfill a fixed schedule of loaded transportation requests among different terminals (or depots) throughout a planning horizon. The operator has to make two main decisions to satisfy the fixed schedule of loaded moves: 1) the fleet size (i.e., the number of resources to use), and 2) the resource schedule throughout the planning horizon (i.e., the activities that each resource performs). Available resources must be assigned to the different loaded tasks, but due to imbalances between the number of requests to/from the different terminals in the transportation network, resources may sometimes need to be repositioned from terminals with a surplus of resources to terminals with a deficit of resources. Once a resource has completed a request, it can be left idle at its current location to be used later to satisfy another loaded task originating at the current terminal, or it can be repositioned to a different terminal to be used to fulfill a scheduled loaded request originating there. As a result, a resource schedule must specify for each resource its loaded moves (covering scheduled tasks), its empty (deadheading) moves, and its idle times.

An important feature of the fixed loaded schedules is their periodicity. Aperiodic loaded schedules occur within a finite planning horizon and vary from one planning horizon to the next; in this case, the ending conditions of the resources are not relevant (i.e., it does not matter in which terminal each resource winds up at the end of the current planning horizon) mostly because resources schedules are determined using rolling planning horizons, so any resource schedule that covers all the loaded requests without exceeding the number of available resources is feasible. On the other hand, periodic loaded schedules repeat themselves continuously; the finite planning horizon is just representative of a larger (potentially infinite) time horizon. This characteristic imposes an additional constraint for a resource schedule to be feasible, namely, that each of the terminals must end up with the same number of resources it started with at the beginning of the planning horizon (since otherwise the same schedule could not be replicated in the next planning horizon). Regardless of its importance, the inclusion of the periodicity of a scheduled is sometimes a modeling choice and as such it might be ignored when dealing with periodic schedules (specially in cases that assume a large enough time between two successive planning cycles) or added when dealing with aperiodic schedules (specially in cases in which even though no ending conditions of the resources can be specified, a certain regularity on the ending locations is desired).

We model the associated situation in two different ways: on one hand we develop mathematical programming formulations of flows on event-based, time-expanded networks; on the other hand, we define perfect matching problems on bipartite networks. For aperiodic schedules, all of the points in the optimal Pareto frontier can be computed in polynomial time solving linear programming formulations of flows on the time-expanded networks or solving minimum weight perfect matching problems on the bipartite networks. In addition, we define an incremental problem to efficiently compute adjacent points in the frontier by solving a single shortest path problem in either type of network. Aperiodic schedules are more difficult. The end points in the frontier can be computed in polynomial time by solving a sequence of two linear problems and the rest of the points on the frontier can be computed using either integer programming flow formulations or assignment formulations with additional side constraints.

The rest of the chapter is organized as follows: Section 3.1 reviews the related literature on fleet size and repositioning problems related to transportation schedules. Section 3.2 presents the modeling framework used and describes how to construct the time-expanded and bipartite networks. Section 3.3 discusses the computation of every point on the optimal Pareto frontier, and presents efficient procedures to compute adjacent Pareto points. Finally, Section 3.4 discusses the practical applications of the models developed, compares the two different types of networks, and presents the results of computational experiments using information from a major less-than-truckload (LTL) carrier.

3.1 Related Literature

The problem of minimizing the number of resources to meet a fixed transportation schedule has been studied by the research community for a long time. The work in this area can be classified into two main categories according to the characteristics of the fixed schedule of resources. The first category comprises aperiodic schedules with finite planning horizons, while the second category covers periodic schedules with potentially infinite planning horizons.

[14] was among the pioneering works in this area. It reported that the minimum number of vehicles to meet a fixed aperiodic schedule can be solved in polynomial time by modeling the problem of minimizing the number of tankers to meet a fixed schedule as a classic transportation problem and solving it using the well-known simplex method.

[27] solved the simpler problem of minimizing the number of resources to meet a fixed schedule of jobs (i.e., transportation schedules in which the starting and ending terminals are the same for all the scheduled requests), along with the complicating variant in which each job has a time window to start and end. The results of this research were later used by [9], which studied how repositioning could help reduce the required fleet size for a multi-terminal urban bus transportation system. A two-phase approach was developed based on a so-called "deficit function" to reduce the total bus fleet required to meet a fixed aperiodic scheduled of trips by inserting deadheading trips. A lower bound on the minimum fleet size that can be attained by exploiting all possible deadheading opportunities was also introduced. This lower bound was later improved by [50]. [8] extended this work further by permitting variable departure times along with the possible insertions of deadheading trips.

[23] also studied an urban bus system involving only two terminals between which buses circulate performing alternating deadheading schedules in which some buses return empty while others return in service. Three main problems were addressed:

finding the minimum fleet size needed to meet a given alternating deadheading schedule; constructing the alternating deadheading schedule that minimizes the required fleet size subject to level-of-service constraints; and finding the alternating deadheading schedule that minimizes wait time for a given fleet size.

On the periodic side, [36] considered and solved in polynomial time the problem of minimizing the number of vehicles to meet a fixed periodic schedule by posing it as special case of the "minimum chain-cover" problem for periodic partially ordered sets (posets), and then solving this more general problem as a finite network flow problem. The application context was that of an airline that wishes to schedule a minimum number of airplanes to meet a fixed daily-repeating set of flights, in which deadheading between airports is allowed. [36] also reported that the problem of minimizing the number of vehicles to meet a periodic schedule in which deadheading is not allowed was solved by [5] and [6] using railroad scheduling as the application context. [26] showed that deficit functions become periodic for periodic schedules in which the total number of scheduled tasks into and out of a terminal are equal.

Regarding solution techniques, multicriteria optimization has long been used to model problems with competing objective functions. In particular, multiobjective combinatorial optimization problems related to transportation network design and routing are vast and have received a lot of attention by the research community. [13, 12, 52, 19] review the works in this area, which include the development of exact and heuristic techniques to compute all (or a subset) of the points on the optimal Pareto frontier.

Even though their single criterion counterparts are polynomially solvable, multicriteria network flow problems and assignment problems are NP-hard even for two criteria [18]. Theory and algorithms for solving the multiobjective minimum cost flow problem are reviewed in [30]. An algorithm for the continuous versions of bicriteria network flow problems is presented in [32]. [33] expanded on these ideas and developed a procedure to compute the efficient solutions to the bicriteria integer network flow problem. This latter problem has also been studied by [43, 46, 41, 44]. On the other hand, [42] studied the biobjective assignment problem.

In addition, the efficient computation of exact (or approximate) Pareto points has also been addressed in other contexts. [38] developed a heuristic two-phase solution procedure for the biobjective dial-a-ride problem, and [53] developed tabu-search and genetic algorithms to solve a bicriteria general job shop scheduling problem arising in the printing and boarding industry.

Finally, other examples in the literature involve tactical models that explicitly take into account the impact of operational decisions. In the context of supply chain design, [47] developed a nonlinear model that determines distribution center locations to optimize cost and service objectives, explicitly considering the routing costs of assigning demands to distribution centers and the inventory policy costs at the distribution centers. Additionally, [55] addressed a fleet sizing problem in the context of the truck-rental industry, incorporating the operational decisions of demand allocation and empty truck repositioning with the tactical decisions of asset procurement and sales into a linear programming model to determine the optimal fleet size and mix.

3.2 Modeling Frameworks

Most large transportation service providers face two issues associated with fulfilling their scheduled tasks throughout a planning horizon: establishing the number of resources (fleet size) they should use and determining the schedule that each resource should follow within the planning horizon. A resource schedule indicates the activities that the resource performs at every point in time within the horizon, and these activities include staying idle at some terminal, being moved between two terminals to satisfy a scheduled request, or being repositioned between two terminals. There is a trade-off between fleet size and repositioning; the larger the fleet size, the less likely it is to require to relocate a resource to a different terminal to serve the same number of loaded tasks (up to a point where repositioning can no longer be reduced by increasing the fleet size), and conversely, a smaller fleet size increases the need of resource deadheading. The fixed schedules can be aperiodic or periodic; the former involve scheduled tasks occurring within a finite planning horizon which are likely to vary in the next planning horizon; the latter involve scheduled tasks that repeat over a long (possibly infinite) time horizon.

Several modeling alternatives can be used to represent resource schedules. In this chapter, we present two of them which we will later contrast in Section 3.4.2. In the first approach, we use mathematical programming formulations of flows on time-expanded networks. In this setting, a resource schedule corresponds to a path in the network for the aperiodic case or a cycle in the network for the periodic case. In the second approach, we define perfect matching problems on suitably defined bipartite networks in which a matching or assignment corresponds to the next task to be executed by a given resource.

Next, we outline the construction of both types of networks, for which we consider the following input parameters:

- $\mathcal{L}$ = the finite set of terminals (or depots) in the transportation network.
- t_{max} = the number of periods in the planning horizon.
- $\mathcal{S}$ = the set of all scheduled loaded requests, that is, the fixed transportation schedule to be met. Each scheduled loaded request $s \in \mathcal{S}$ is specified by a 5-tuple $s = (l_1, t_1, l_2, t_2, d)$, with $l_1, l_2 \in \mathcal{L}$, $0 \leq t_1, t_2 \leq t_{max}$, and $d \in \mathbb{Z}_+$. l_1 is the departure terminal, t_1 is the departure time, l_2 is the arrival terminal, t_2 is the arrival time, and d is the number of resources required to satisfy the

transportation request. $\mathbb{Z}_+$ denotes the set of nonnegative integers.

- $\tau_{l_1 l_2}$ = the travel time from terminal l_1 to terminal l_2, $\forall\, l_1, l_2 \in \mathcal{L}$

3.2.1 Construction of the time-expanded networks

Time-expanded networks are commonly used to model logistical problems. In this application, we use event-based time-expanded networks in which each node represents a specific terminal at a particular point in time, and each arc represents either the movement of resources between different terminals at different times to satisfy a scheduled request (demand arc), the deadheading movement of resources between different terminals at different times (repositioning arc) or the idleness of resources at a given terminal (inventory arc). Networks corresponding to aperiodic schedules include in addition a source and a sink node, as well as source arcs and sink arcs which represent initial and ending locations of the resources.

For aperiodic schedules, the steps to construct the time-expanded network are the following:

1. For each scheduled request $s = (l_1, t_1, l_2, t_2, d) \in \mathcal{S}$:
   - Create a departure node (l_1, t_1) and an arrival node (l_2, t_2). In addition, create a demand arc from the departure node to the arrival node.
   - For each terminal $l_r \in \mathcal{L} \backslash \{l_2\}$, if $t_2 + \tau_{l_2 l_r} \leq t_{max}$ create a repositioning node $(l_r, t_2 + \tau_{l_2 l_r})$ plus a repositioning arc joining the arrival node to the repositioning node. Note that the repositioning node might not need to be created if it coincides with the departure/arrival node of some scheduled request previously handled.
2. Create an inventory arc connecting each node created in the previous step to the closest node corresponding to the same terminal at a later time if such node exists.

3. Create a source node (labeled $(Source, -1)$ to be consistent with the node notation), and for each terminal create a source arc connecting the source node to the terminal node with the earliest time. Create also a sink node $(Sink, t_{max}+1)$, and for each terminal create a sink arc connecting the terminal node with the latest time to the sink node.

For periodic schedules, the steps to construct the time-expanded network are the following:

1. For each scheduled request:
   - Create a departure node (l_1, t_1) and an arrival node (l_2, t_2). In addition, create a demand arc from the departure node to the arrival node.
   - For each terminal $l_r \in \mathcal{L} \backslash \{l_2\}$, if $t_2 + \tau_{l_2 l_r} \leq t_{max}$ create a repositioning node $(l_r, t_2 + \tau_{l_2 l_r})$, otherwise, create a repositioning node $(l_r, t_2 + \tau_{l_2 l_r} - t_{max})$; create also a repositioning arc joining the arrival node to the repositioning node.
2. Create an inventory arc connecting each node created in the previous step to the closest node corresponding to the same terminal at a later time if such node exists. In addition, for each terminal, connect its latest node to its earliest node.

Figure 9 exemplifies the construction of the time-expanded networks for the following set of five scheduled requests involving four terminals, and a planning horizon $[0, 10]$: $\mathcal{S} = \{\alpha = (2, 0, 1, 2, 1), \beta = (4, 3, 1, 4, 1), \gamma = (2, 4, 3, 5, 1), \delta = (3, 6, 2, 7, 1), \epsilon = (2, 6, 4, 9, 1)\}$. The repositioning times are $\tau_{12} = \tau_{21} = 2, \tau_{13} = \tau_{31} = 3, \tau_{14} = \tau_{41} = 1, \tau_{23} = \tau_{32} = 1, \tau_{24} = \tau_{42} = 3$, and $\tau_{34} = \tau_{43} = 4$.

Note that in the periodic case, the arcs corresponding to moves that begin during the planning horizon but are completed after the end of the horizon are wrapped around and connected to earlier periods of the horizon.

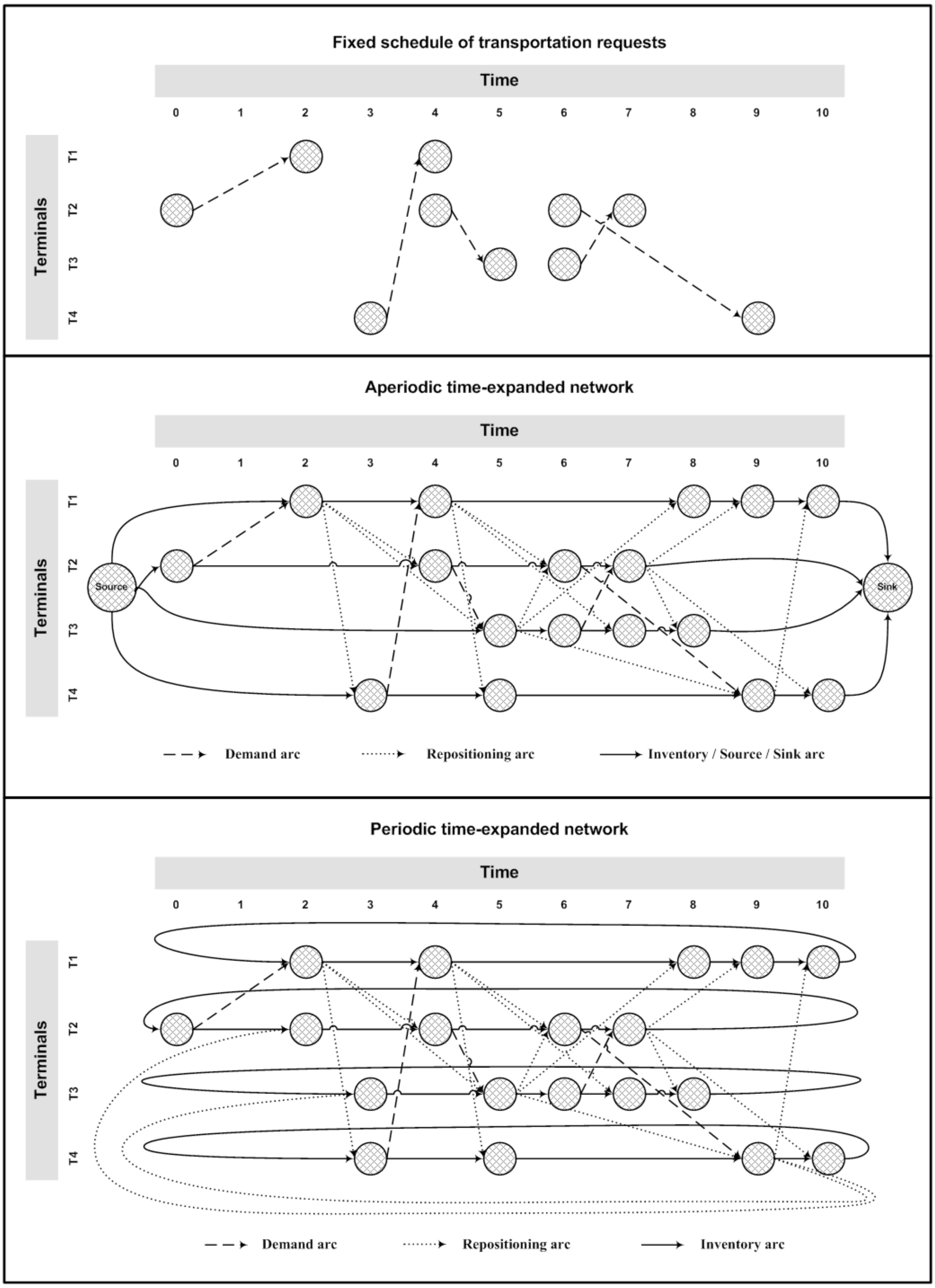

Fixed schedule of transportation requests
Time
0
1
2
3
4
5
6
7
8
9
10
Terminals
T1
T2
T3
T4
Aperiodic time-expanded network
Source
Sink
Demand arc
Repositioning arc
Inventory / Source / Sink arc
Periodic time-expanded network
Inventory arc

Figure 9: Construction of the time-expanded networks

We will denote a time-expanded network as $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, where $\mathcal{N}$ is the set of nodes, and $\mathcal{A}$ is the set of arcs. For periodic schedules $\mathcal{A} = \mathcal{A}^I \cup \mathcal{A}^D \cup \mathcal{A}^R$, whereas for aperiodic schedules $\mathcal{A} = \mathcal{A}^I \cup \mathcal{A}^D \cup \mathcal{A}^R \cup \mathcal{A}^{So} \cup \mathcal{A}^{Si}$. $\mathcal{A}^I, \mathcal{A}^D, \mathcal{A}^R, \mathcal{A}^{So}$, and $\mathcal{A}^{Si}$ are respectively the sets of inventory, demand, repositioning, source, and sink arcs. As shown before, we can refer to a node $i \in \mathcal{N}$ as $i = (l, t)$, where l is the specific terminal and t is the specific time associated with node i; we can also refer to an arc $a \in \mathcal{A}$ as $a = (i, j)$, where i and j are respectively the tail and head nodes of arc a. For demand arcs, $d_{ij}, \forall (i, j) \in \mathcal{A}^D$, will denote the number of resources required to satisfy the associated request; and for repositioning arcs, $\tau_{ij}, \forall (i, j) \in \mathcal{A}^R$, will denote the travel time between the departure and arrival terminals.

The computation of the points in the optimal Pareto frontier will use mathematical programming formulations of flows on the described time-expanded networks. These formulations will use as decision variables x_{ij} = the amount of flow on arc (i, j) (measured in units of resources), $\forall (i, j) \in \mathcal{A}$, and we will denote with $x \in \mathbb{R}^{|\mathcal{A}|}$ a vector containing all flow values. The flow on the inventory arcs, demand arcs, and repositioning arcs correspond respectively to resources remaining idle at the corresponding terminal, loaded movement of resources serving the scheduled requests, and empty movement of resources, and, in the case of aperiodic networks, the flows on the source and sink arcs represent starting and ending locations of the resources, respectively.

3.2.2 Construction of the bipartite networks

As an alternative, given a fleet of size k, resource schedules can be modeled as perfect matchings in a bipartite network which includes resource nodes and task nodes. This approach is based on a similar idea to the one presented by [14] to model the tanker scheduling problem as a classical transportation problem. In this case, only a single type of bipartite network will be defined and periodic schedules will require additional side constraints to be properly modeled. The construction of the bipartite networks is outlined next:

1. Create a resource node for each of the k available resources (to represent units of equipment which have not been used on the given planning horizon). We will denote the set of all these nodes as $V^k_{R_{start}}$.

2. Create d resource nodes associated with the end of each scheduled task $s = (l_1, t_1, l_2, t_2, d) \in \mathcal{S}$ (to represent the availability of each of the d units of equipment which just satisfied that task). We will denote the set of all these nodes as $V_{R_{request}}$.

3. Create d task nodes associated with the beginning of each scheduled request $s = (l_1, t_1, l_2, t_2, d) \in \mathcal{S}$ (to represent the demand of d units of equipment to perform that request). The set of all such nodes will be denoted as $V_{T_{request}}$.

4. Create a task node associated with each of the k available units of equipment (to represent the end of the usage of each resource over the planning horizon). The set of all such nodes will be denoted as $V^k_{T_{end}}$.

5. Create edges joining every node in $V_{R_{start}}$ to every node in $V^k_T = V_{T_{request}} \cup V^k_{T_{end}}$.

6. Create edges joining every node in $V^k_R = V^k_{R_{start}} \cup V_{R_{request}}$ to every node in $V_{T_{end}}$.

7. For every pair of nodes $\upsilon_R \in V_{R_{request}}, \upsilon_T \in V_{T_{request}}$, create edge (υ_R, υ_T) if the request $\alpha = (l^\alpha_1, t^\alpha_1, l^\alpha_2, t^\alpha_2, d^\alpha)$ that generated υ_T can be feasible performed after the request $\beta = (l^\beta_1, t^\beta_1, l^\beta_2, t^\beta_2, d^\beta)$ that generated υ_R (i.e., if $t^\alpha_1 \geq t^\beta_2 + \tau_{l^\beta_2 l^\alpha_1}$).

Let E^k represent the set of all edges created. Then, $G^k = (V^k_R \cup V^k_T, E^k)$ is an undirected bipartite graph in which a perfect matching or assignment corresponds to a feasible resource schedule as each of the matchings indicates either the next scheduled request that an available resource has to fulfill, or the end of usage of that resource in the given planning horizon. We will also define weights w_e, $\forall e \in E^k$, whose specific values will depend on the problem being solved. Figure 11, part a) shows the construction of G^k for the same schedule of requests presented earlier.

3.3 Efficiency frontier of fleet size versus repositioning

In multi-criteria optimization models, a Pareto solution or Pareto point is a feasible solution x such that any other feasible solution with a strictly better value for one of the objectives attains a worse value than x for at least one other objective. The set of all optimal Pareto points is known as the optimal Pareto frontier, or optimal efficiency frontier. This chapter studies the optimal Pareto frontier of fleet size versus repositioning associated with entirely fulfilling aperiodic or periodic schedules of fixed transportation requests. Figure 10 exemplifies an optimal Pareto frontier. We will refer to points A and C as the end points of the frontier, and points $B_1, B_2, \ldots$ as the interior points of the frontier. Throughout this chapter we assume that the cost of operating and maintaining a fleet of resources of a certain size is a linear function on the number of resources. We also assume that the cost of repositioning one resource between two terminals is a linear function on the travel time between the two terminals. Using these metrics, point $A = (k_{min}, r_{max})$ corresponds to the minimum repositioning time r_{max} that can be attained given the minimum fleet size k_{min} which can fulfill the fixed schedule (it is denoted as r_{max} because it is the maximum repositioning time that should ever occur in any efficient resource schedule); point $C = (k_{max}, r_{min})$ corresponds to the minimum fleet size k_{max} that is required to execute a resource schedule with the minimum possible repositioning time r_{min} (k_{max} is the largest fleet size required in any efficient resource schedule); and points $B_1 = (k_{min} + 1, r_1), B_2 = (k_{min} + 2, r_2), \ldots$ correspond to the minimum repositioning time r_n required to fulfill the fixed schedule using $k_{min} + n$ resources, $\forall n \in \{1, \ldots, k_{max} - k_{min} - 1\}$.

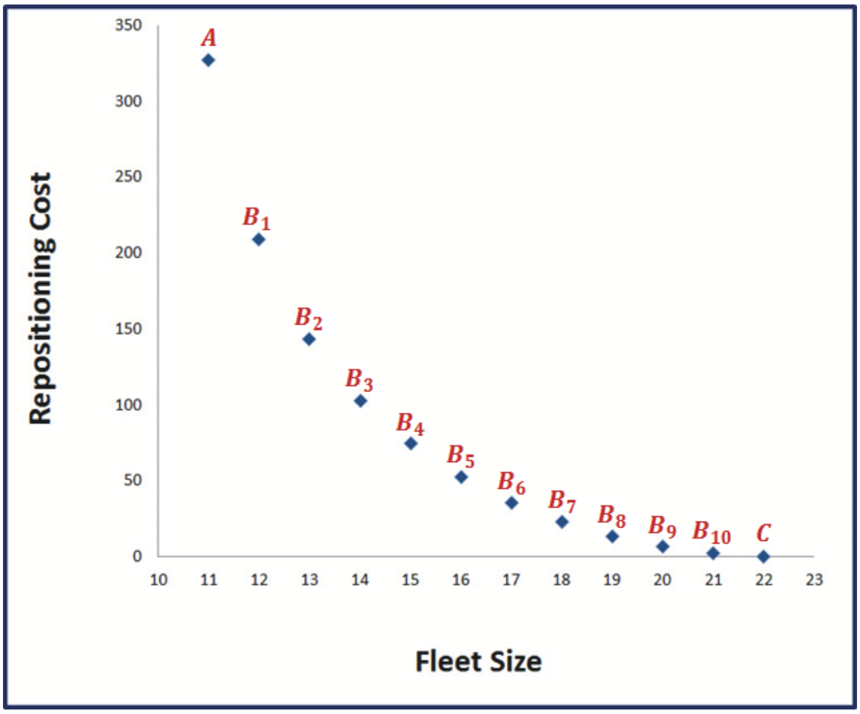


Figure 10: Example of an optimal Pareto frontier

One basic problem associated with computing the points in the optimal Pareto frontier is the computation of k_{max} and k_{min}, which are respectively the largest and smallest fleet sizes that could ever be required to satisfy a given fixed schedule of transportation requests; k_{max} is the fleet size associated with the resource schedule having the smallest possible total repositioning time, whereas k_{min} is the fleet size which if decreased by one unit would result in an inability to cover all the scheduled tasks regardless of the repositioning moves introduced in the resource schedule. In Figure 10, $k_{min} = 11$ (point A) and $k_{max} = 22$ (point C).

We are interested in developing models to individually compute each Pareto point (together with its associated resource schedule), and we are also interested in the following **incremental problem**: given an optimal resource schedule with $k \geq k_{min}$ resources, how to obtain an optimal resource schedule with $k + 1$ resources? Note that developing efficient procedures to solve the incremental problem will yield an alternative and expectedly more efficient procedure to compute all of the Pareto points, since instead of solving an independent model to determine each point, one can start at point $A = (k_{min}, r_{max})$, and solve the incremental problem until point $C = (k_{max}, r_{min})$ is reached.

The next subsections present models for the computation of all of the points on the optimal Pareto frontier for aperiodic and periodic schedules, as well as algorithms to solve the associated incremental problems.

3.3.1 Efficiency frontier for aperiodic schedules

3.3.1.1 Computing all the Pareto points using time-expanded networks

Consider the aperiodic time-expanded networks whose construction was outlined in Section 3.2.1. Using the notation introduced there, k_{min} can be computed using the following linear program:

$$\begin{aligned} k_{min} = \quad & min \quad f \\ & s.t. \quad \textstyle\sum_{\{j:(i,j)\in A\}} x_{ij} - \sum_{\{j:(j,i)\in A\}} x_{ji} = \begin{cases} f & i = (Source, -1) \\ -f & i = (Sink, t_{max}+1) \\ 0 & otherwise \end{cases} \\ & \qquad x_{ij} = d_{ij} \qquad\qquad\qquad\qquad\qquad \forall (i,j) \in \mathcal{A}^D \\ & \qquad x_{ij} \geq 0 \qquad\qquad\qquad\qquad\qquad\; \forall (i,j) \in \mathcal{A} \end{aligned}$$

Note that the previous problem corresponds to a minimum cost network flow (MCNF) problem and as such is guaranteed to produce integral flows. The variable f represents the external supply of resources into the source node (out of the sink node).

On the other hand, given that k_{max} is associated with the resource schedule that attains the minimum possible repositioning time r_{min}, and observing that $r_{min} = 0$ (consider for instance a resource schedule in which each task is executed by one exclusive resource), it follows that k_{max} can be computed by solving an MCNF problem on a time-expanded network built without any repositioning arcs. As an alternative, the "deficit-function" techniques outlined by [9] can also be used to compute k_{max}. These techniques are essentially counting techniques that consider the number of resources moved into and out of a terminal at different points in time. It follows that the end point $C = (k_{max}, 0)$ can be computed in polynomial time.

The rest of the points in the efficiency frontier (points $A, B_1, B_2, \ldots$) can also be computed in polynomial time via linear programming:

$$\begin{array}{rll} r_n = & min & \sum_{(i,j)\in\mathcal{A}^R} \tau_{ij}x_{ij} \\ & s.t. & \sum_{\{j:(i,j)\in A\}} x_{ij} - \sum_{\{j:(j,i)\in A\}} x_{ji} = \begin{cases} k_{min}+n & i = (Source, -1) \\ -(k_{min}+n) & i = (Sink, t_{max}+1) \\ 0 & otherwise \end{cases} \\ & & x_{ij} = d_{ij} \qquad\qquad\qquad\qquad\qquad\qquad \forall(i,j) \in \mathcal{A}^D \\ & & x_{ij} \geq 0 \qquad\qquad\qquad\qquad\qquad\qquad \forall(i,j) \in \mathcal{A} \end{array}$$

Point $A = (k_{min}, r_0)$, whereas points $B_n = (k_{min} + n, r_n), \forall n \in \{1, \ldots, k_{max} - k_{min} - 1\}$. Note also that point $C = (k_{max}, r_{k_{max}-k_{min}})$. For notational convenience in the following section, we will let $k = k_{min} + n$ and refer to the previous LPs as **LP(k)**, for some $k_{min} \leq k \leq k_{max}$; we will also let $c_{ij} = \tau_{ij}, \forall(i,j) \in \mathcal{A}^R$ and $c_{ij} = 0, \forall(i,j) \in \mathcal{A}\backslash\mathcal{A}^R$ to express the objective function in $LP(k)$ as $\sum_{(i,j)\in A} c_{ij}x_{ij}$. Given the correspondence of $LP(k)$ to an MCNF problem, feasible solutions to $LP(k)$ are referred to as flows. Once an optimal flow x has been determined for a given k, the specific resource schedules (which correspond to paths of arcs carrying a flow of 1 from the source node to the sink node) can be identified by running a flow decomposition algorithm.

3.3.1.2 Solving the incremental problem using time-expanded networks

In the context of aperiodic time-expanded networks, the incremental problem of computing adjacent Pareto points can be posed as: given an optimal solution to $LP(k)$, $k \geq k_{min}$, how to compute an optimal solution to $LP(k+1)$? In what follows, we will show how to efficiently solve this problem.

First, we need to recall some additional concepts. A pseudoflow in $LP(k)$ is an infeasible solution to $LP(k)$ that satisfies all the bounds and integrality requirements but violates at least one flow balance constraint. Given a pseudoflow, the difference between the right-hand side and the left-hand side of a balance constraint is the imbalance at node i. A positive imbalance is the excess at node i, whereas the absolute value of a negative imbalance is the deficit at node i.

Given a network $G = (V, E)$, we can associate with each node $i \in N$ a real number $\pi(i)$, which is referred to as the potential of node i. In addition, the reduced cost c_{ij}^{π} of an arc (i, j), with respect to node potentials $\pi = (\pi(1), \pi(2), \dots, \pi(|N|))$, are defined as $c_{ij}^{\pi} = c_{ij} - \pi(i) + \pi(j)$.

Given a network G, and a flow x, $G(x)$ denotes the residual network associated with flow x. This network contains all the nodes in G and the arcs or arc reversals with positive residual capacity. In the case of an aperiodic time-expanded network $\mathcal{G}$, all arcs $(i, j) \in \mathcal{A} \backslash \mathcal{A}^D$ are uncapacitated, and arcs in $\mathcal{A}^D$ have a fixed flow. Thus, $\mathcal{G}(x)$ can be constructed from $\mathcal{G}$ by including all arcs in $\mathcal{A} \backslash \mathcal{A}^D$ with a cost c_{ij} and, for every arc $(i, j) \in \mathcal{A} \backslash \mathcal{A}^D$ with $x_{ij} > 0$, adding arc reversal (j, i) with a residual capacity equal to x_{ij} and a cost $-c_{ij}$.

Next, we state without proof the following property, theorem, and lemmas, which will be used to prove some results in this section. Their proofs can be found in [2], where they correspond respectively to Property 2.5, Theorem 9.3, and Lemmas 9.11 and 9.12.

Property 1. *a) For any directed cycle W and for any node potentials π, $\sum_{(i,j) \in W} c_{ij}^{\pi} = \sum_{(i,j) \in W} c_{ij}$.*

b) For any directed path P from node u to node v and for any node potentials π, $\sum_{(i,j) \in P} c_{ij}^{\pi} = \sum_{(i,j) \in W} c_{ij} - \pi(u) + \pi(v)$.

Theorem 1. *A feasible solution x^* is an optimal solution of the minimum cost flow problem if and only if some set of node potentials π satisfy the following reduced cost optimality conditions: $c_{ij}^{\pi} \geq 0, \quad \forall (i,j) \in G(x^*)$*

Lemma 1. *Suppose that a pseudoflow (or a flow) x satisfies the reduced cost optimality conditions with respect to some node potentials π. Let the vector ℓ represent the shortest path distances from some node u to all other nodes in the residual network $G(x)$ with c_{ij}^{π} as the length of an arc (i,j). Then, the following properties are valid:*

a) The pseudoflow x also satisfies the reduced cost optimality conditions with respect to the node potentials $\pi' = \pi - \ell$.

b) The reduced costs $c_{ij}^{\pi'}$ are zero for all arcs (i,j) in a shortest path from node u to every other node.

Lemma 2. *Suppose that a pseudoflow (or a flow) x satisfies the reduced cost optimality conditions and we obtain x' from x by sending flow along a shortest path from node u to some other node v; then x' also satisfies the reduced cost optimality conditions.*

We will now introduce and prove a theorem that shows how to efficiently solve the incremental problem on the aperiodic time-expanded networks.

Theorem 2. *Let x_k be an optimal solution to LP(k), for some $k_{min} \leq k \leq k_{max} - 1$ (i.e., the flow corresponding to the resource schedule that attains minimum repositioning time using k resources); let $\mathcal{P}_S$ be a shortest path from $(Source, -1)$ to $(Sink, t_{max}+1)$ in $\mathcal{G}(x_k)$; and let x' be the flow that results from x_k by augmenting one unit of flow along $\mathcal{P}_S$. Then, $x_{k+1} = x'$.*

Proof. First, observe that $\mathcal{G}(x_k)$ contains a directed path from $(Source, -1)$ to every other node, and it does not contain negative cycles (otherwise, the optimality of x_k would be contradicted). Thus, $\mathcal{P}_S$ is well defined. Let $\pi \in \mathbb{R}^{|\mathcal{N}|}$ be an optimal set of node potentials satisfying Theorem 1's reduced cost optimality conditions in

$G(x_k)$. Note that x_k is a pseudoflow in $LP(k+1)$, in which there is a unique unit of flow excess at $(Source, -1)$ and a unique unit of flow deficit at $(Sink, t_{max}+1)$. On one hand, augmenting one unit of flow along $\mathcal{P}_S$ eliminates the flow excess at $(Source, -1)$ and the flow deficit at $(Sink, t_{max}+1)$, and it does not affect the flow balance constraints. Thus, x' is a feasible flow in $LP(k+1)$. On the other hand, by Lemma 2, x' also satisfies the reduced cost optimality conditions (together with node potentials $\pi' \in \mathbb{R}^{|\mathcal{N}|}$ updated according to Lemma 1). Therefore, x' is an optimal solution to $LP(k+1)$. □

Theorem 2 implies that we can compute all Pareto points by starting with $x_{k_{min}}$ and iteratively finding a shortest path from $(Source, -1)$ to $(Sink, t_{max}+1)$ and augmenting one unit of flow along such path in the residual network built with respect to the original costs c_{ij}. Note that since the arc reversals of repositioning arcs have negative costs, Bellman-Ford's algorithm needs to be applied to compute the desired shortest paths. However, we can use node potentials to obtain and maintain nonnegative arc costs in the residual networks so that we can use the more efficient Dijkstra's algorithm to find the shortest paths (Property 1 guarantees the correctness of Theorem 2 under this cost transformation). Algorithm 1 summarizes the steps to find all Pareto points by solving the incremental problem iteratively.

To show the correctness of Algorithm 1 it suffices to argue that the arc costs in the residual network start nonnegative and remain nonnegative throughout all the steps: Since the optimal dual variables for the balance constraints are optimal node potentials, it follows that the costs of the residual network start nonnegative; in addition, since, the node potentials are updated at each step according to Lemma 1, it follows from Lemma 2 that the costs in the residual network remain nonnegative throughout the algorithm.

Algorithm 1 Incremental computation of Pareto points on time-expanded networks

```
Compute $k_{min}$ and $k_{max}$.
Solve $LP(k_{min})$. Let $x$ and $\pi$ be respectively the vector of optimal primal variables
and the vector of optimal dual variables for the balance constraints, and let $r_{max}$
be the optimal objective function value.
Output point $(k_{min}, r_{max})$.
Build $\mathcal{G}(x)$ with respect to costs $c^{\pi}_{ij} = c_{ij} - \pi(i) + \pi(j)$.
Set $Repositioning \leftarrow r_{max}$.
for $k = k_{min}$ to $k_{max} - 1$ do
    Use Dijkstra's algorithm to find the shortest paths in $\mathcal{G}(x)$ from $(Source, -1)$
    to every other node. Let the vector $\ell$ represent the shortest path distances
    from $(Source, -1)$ to all other nodes, and let $\mathcal{P}_S$ be the shortest path from
    $(Source, -1)$ to $(Sink, t_{max} + 1)$.
    Update $x$ by augmenting one unit of flow along $\mathcal{P}_S$.
    Update $\pi \leftarrow \pi - \ell$.
    Update $\mathcal{G}(x)$ and its costs $c^{\pi}_{ij}$.
    Update $Repositioning \leftarrow Repositioning - \ell((Sink, t_{max} + 1))$.
    Output point $(k + 1, Repositioning)$.
end for
```

3.3.1.3 Computing all Pareto points using bipartite networks

Consider the bipartite networks G^k for a given integer k, whose construction was outlined in Section 3.2.2. We will use the notation defined there and we will also let $D = \sum_{s \in \mathcal{S}} d^s$, where d^s is the number of resources required to satisfy request s. In this case, k_{min} can be computed by solving a minimum weight perfect matching on G^D with weights:

$$w_e = \begin{cases} 1 & \forall e = (\upsilon_R, \upsilon_T) \quad \text{s.t.} \quad \upsilon_R \in V^D_{R_{start}} \quad \text{and} \quad \upsilon_T \in V_{T_{request}} \\ 0 & otherwise \end{cases}$$

Observing that $r_{min} = 0$, k_{max} can also be computed by solving a minimum weight perfect matching on the network H^D, which is the same as the network G^D, but without the edges (υ_R, υ_T), with $\upsilon_R \in V_{R_{request}}, \upsilon_T \in V_{T_{request}}$ for which the terminal associated with υ_R is different from the terminal associated with υ_T (i.e, H^D is a network in which no repositioning moves are allowed). Using the same weights w_e as above, this computation yields point $C = (k_{max}, 0)$.

Now, given a fleet of size $k \geq k_{min}$, the problem of finding a resource schedule with minimum repositioning time reduces to finding a minimum weight perfect matching in G^k, using as weights w_e the repositioning times incurred by the move related to edge e. That is, edges incident to $V_{R_{start}} \cup V_{T_{end}}$ have a weight of zero, whereas edges incident to both $V_{R_{request}}$ and $V_{T_{request}}$ have a weight equal to the repositioning time between the terminal that generated the resource node and the terminal that generated the task node. The weight W^k of the optimal matching in G^k corresponds to the minimum repositioning time using k resources. The Pareto points are then (k, W^k), $\forall k_{min} \leq k \leq k_{max}$.

3.3.1.4 Solving the incremental problem using bipartite networks

In the context of bipartite networks, the incremental problem of computing adjacent Pareto points can be posed as: given a minimum weight perfect matching in G^k, $k \geq k_{min}$ (with w_e representing the repositioning time from the resource node to the task node), how to obtain a minimum weight perfect matching in G^{k+1}? In this section, we will show how to efficiently solve this problem.

First, we need to define another auxiliary bipartite graph, based on G^k. Let M be a minimum weight perfect matching on G^k and note that it is also a matching on G^{k+1}. Let $\vec{G}_M^{k+1} = (V_R^{k+1} \cup V_T^{k+1}, A^{k+1})$ be the directed bipartite graph that is constructed from G^{k+1} and M by orienting the edges as follows: edges in the matching are oriented from task node to resource node and get the signs of their weights reversed, whereas edges not in the matching are oriented from resource node to task node and maintain their original weights. We will denote these new weights by $\vec{w}_a, \forall a \in A^{k+1}$

Figure 11 shows the construction of G^k and $\vec{G}_M^{k+1}$ for the set of scheduled requests $\mathcal{S} = \{\alpha = (2,0,1,2,1), \beta = (4,3,1,4,1), \gamma = (2,4,3,5,1), \delta = (3,6,2,7,1), \epsilon = (2,6,4,9,1)\}$, with repositioning times given by: $\tau_{12} = \tau_{21} = 2, \tau_{13} = \tau_{31} = 3, \tau_{14} = \tau_{41} = 1, \tau_{23} = \tau_{32} = 1, \tau_{24} = \tau_{42} = 3$, and $\tau_{34} = \tau_{43} = 4$. Part a) shows the undirected bipartite graph G^k, part b) shows a minimum weight perfect matching in G^k, and part c) shows the directed bipartite graph $\vec{G}_M^{k+1}$.

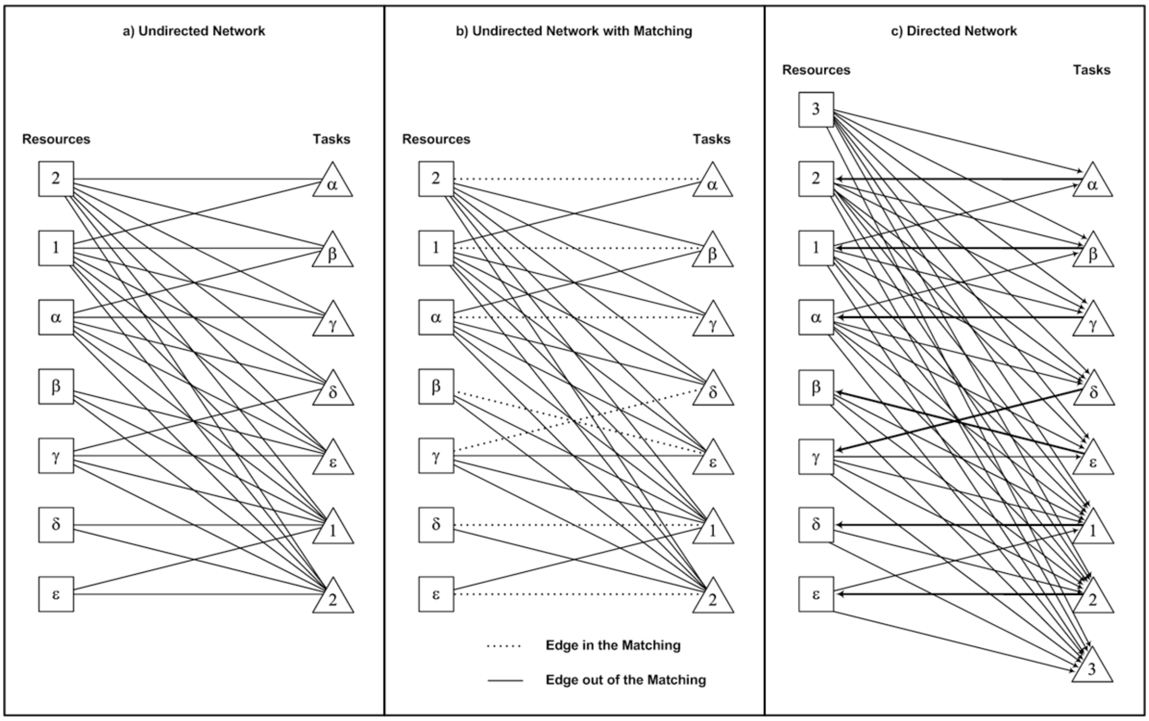


Figure 11: Construction of the bipartite networks

We will now introduce and prove a theorem that shows how to efficiently solve the incremental problem on the bipartite networks. Consider the following additional notation:

- $U \Delta V = U \cup V \backslash U \cap V$, that is, the symmetric difference of sets U and V.
- $E[P], E[P_S], E[C], E[C_i] :=$ undirected edge sets of P, P_S, C and C_i
- $A[P], A[P_S], A[C] :=$ arc set of P, P_S, and C
- $W(U) = \sum_{e \in U} w_e$ for some $U \subseteq E^k$ or some $U \subseteq E^{k+1}$
- $\vec{W}(U) = \sum_{a \in U} \vec{w}_a$ for some $U \subseteq A^{k+1}$

Theorem 3. *Let M be a minimum weight perfect matching in G^k, for some $k_{min} \leq k \leq k_{max} - 1$ (i.e., the assignment corresponding to resource schedule that attains* *minimum repositioning time using k resources); let* $\upsilon_R^{k+1} = V_R^{k+1} \backslash V_R^k$ *and* $\upsilon_T^{k+1} = V_T^{k+1} \backslash V_T^k$*; and let* P_S *be a shortest path from* υ_R^{k+1} *to* υ_T^{k+1} *in* $\vec{G}_M^k$*. Then,* $M' = M \Delta E[P_S]$ *is a minimum weight perfect matching in* G^{k+1}*.*

Proof. First of all, note that $\vec{G}_M^{k+1}$ does not contain negative cycles (otherwise, the optimality of M would be contradicted). Therefore, P_S is well defined.

Now, let N be a minimum weight perfect matching in G^{k+1}. Note that N exists because $M \cup (\upsilon_R^{k+1}, \upsilon_T^{k+1})$ is a perfect matching in $\vec{G}_M^{k+1}$. It then suffices to show that $W(M') = W(N)$.

($\geq$): Note that since υ_R^{k+1} and υ_T^{k+1} are exposed nodes with respect to M, P_S is an M-augmenting path in G^{k+1}, therefore $\mid M' \mid = \mid M \mid +1$ and by the optimality of N it follows that $W(M') \geq W(N)$.

($\leq$): Consider $\widetilde{G}_N = (V_1^{k+1} \cup V_2^{k+1}, M \Delta N)$, with edge orientations and weights as in $\vec{G}_M^k$, and consider its connected components. Note that only υ_R^{k+1} and υ_T^{k+1} have degree 1, all other nodes have either degree 0 (isolated nodes) or degree 2. Therefore, the connected components are a path from υ_R^{k+1} to υ_T^{k+1} and a (possibly empty) set of disjoint cycles of even cardinality.

Claim: Let C be a cycle in $\widetilde{G}_N$, then $\vec{W}(A[C]) = 0$.

Proof:

- Suppose $\vec{W}(A[C]) < 0$. Then $M \Delta E[C]$ is a matching in G^k with weight $W(M \Delta E[C]) = W(M) + \vec{W}(A[C]) < W(M)$, which contradicts the optimality of M.

- Suppose $\vec{W}(A[C]) > 0$. Then $N \Delta E[C]$ is a matching in G^{k+1} with weight $W(N \Delta E[C]) = W(N) - \vec{W}(A[C]) < W(N)$, which contradicts the optimality of N.

Since the previous claim is true for any cycle C_i in $\widetilde{G}_N$, let $N' = N\Delta(\bigcup_{C_i} E[C_i])$. It follows that N' is an optimal matching in G^{k+1} such that $\widetilde{G}_{N'} = (V_1^{k+1} \cup V_2^{k+1}, M \Delta N')$

contains only a path from υ_R^{k+1} to υ_T^{k+1}, say P.

It then follows that:

$W(N) = W(N') = W(M\Delta E[P]) = W(M) + \vec{W}(A[P]) \geq W(M) + \vec{W}(A[P_S]) = W(M\Delta E[P_S]) = W(M')$

Hence, $W(M') \leq W(N)$ □

Theorem 3, whose proof is along the lines of the proof of the Hungarian Method presented in [45], implies that we can compute all Pareto points by starting with a minimum weight perfect matching in G^k and iteratively finding a minimum weight perfect matching in G^{k+1} by solving a shortest path from υ_R^{k+1} to υ_T^{k+1} in $\vec{G}_M^k$. Given that some of the arc costs are negative, the Bellman-Ford label-correcting algorithm needs to be applied. However, similar to the way the Hungarian algorithm's running time is improved in [45], node potentials could be used to obtain and maintain nonnegative weights at each iteration so that Dijkstra's algorithm can be applied to compute the required shortest paths.

3.3.2 Efficiency frontier for periodic schedules

3.3.2.1 Computing all Pareto points using time-expanded networks

A basic concept used in the formulations on time-expanded networks for periodic schedules is that of a temporal cut at time t, $\mathcal{T}(t) = \{(i,j) = ((\ell_1, t_1), (\ell_2, t_2)) \in \mathcal{A} : t_1 \leq t, t_2 > t\}$ for some $t \in [0, t_{max}]$, which is defined as the subset of arcs in the time-expanded network corresponding to events which start on or before time t and are completed after time t.

Consider the following set:

$$X = \left\{ x \in \mathbb{R}^{|\mathcal{A}|} : \begin{array}{ll} \sum_{\{j:(i,j)\in A\}} x_{ij} - \sum_{\{j:(j,i)\in A\}} x_{ji} = 0 & \forall i \in \mathcal{N} \\ x_{ij} = d_{ij} & \forall (i,j) \in \mathcal{A}^D \\ x_{ij} \geq 0 & \forall (i,j) \in \mathcal{A} \end{array} \right\}$$

Note that X is an integer polyhedron as it is the feasible region of an MCNF problem.

For periodic schedules, k_{min} can be computed using the following LP in which t is any real number between $[0, t_{max}]$: $k_{min} = \text{Min}_{x \in \mathbb{R}^{|\mathcal{A}|}} \left\{ \sum_{(i,j) \in \mathcal{T}(t)} x_{ij} : x \in X \right\}$. However, the computation of k_{max} is not as straightforward as it was for aperiodic schedules because $r_{min} = 0$ only if the fixed schedule of requests is balanced, which happens when the number of resources moving loads into a particular terminal equals the number of resources moving loads out of the same terminal. If the fixed schedule is unbalanced some minimum repositioning will always have to be performed, and $r_{min} > 0$. Nevertheless, r_{min} can be computed with the LP $r_{min} = \text{Min}_{x \in \mathbb{R}^{|\mathcal{A}|}} \left\{ \sum_{(i,j) \in \mathcal{A}^R} \tau_{ij} x_{ij} : x \in X \right\}$.

Now, consider the sets:

$$X^F = \left\{ x \in X : \sum_{(i,j) \in \mathcal{T}(t)} x_{ij} = k_{min} \right\} \text{ and } X^R = \left\{ x \in X : \sum_{(i,j) \in \mathcal{A}^R} \tau_{ij} x_{ij} = r_{min} \right\}$$

Note that both X^F and X^R are faces of the polyhedron X, and as such, they are also integer polyhedra. It then follows that r_{max} and k_{max} can be computed using the following LP's:

$$r_{max} = \text{Min}_{x \in \mathbb{R}^{|\mathcal{A}|}} \left\{ \sum_{(i,j) \in \mathcal{A}^R} \tau_{ij} x_{ij} : x \in X^F \right\}$$

$$k_{max} = \text{Min}_{x \in \mathbb{R}^{|\mathcal{A}|}} \left\{ \sum_{(i,j) \in \mathcal{T}(t)} x_{ij} : x \in X^R \right\}$$

Therefore, the end points of the frontier, $A = (k_{min}, r_{max})$ and $C = (k_{max}, r_{min})$, can be computed in polynomial time.

The rest of the points in the optimal Pareto frontier (points $B_n = (k_{min} + n, r_n), \forall n \in \{1, \ldots, k_{max} - k_{min} - 1\}$) can be computed with the following integer program:

$$\begin{array}{rll} r_n = & min & \sum_{(i,j)\in\mathcal{A}^R} \tau_{ij}x_{ij} \\ & s.t. & \sum_{\{j:(i,j)\in\mathcal{A}\}} x_{ij} - \sum_{\{j:(j,i)\in\mathcal{A}\}} x_{ji} = 0 \quad \forall i \in \mathcal{N} \\ & & \sum_{(i,j)\in\mathcal{T}(t)} x_{ij} = k_{min} + n \\ & & x_{ij} = d_{ij} \qquad\qquad\qquad\qquad\qquad \forall (i,j) \in \mathcal{A}^D \\ & & x_{ij} \in \mathbb{Z}_+ \qquad\qquad\qquad\qquad\qquad \forall (i,j) \in \mathcal{A} \end{array}$$

Note that the previous IP's, denoted $\mathbf{IP(k)}$, for some $k_{min} \leq k = k_{min} + n \leq k_{max}$ correspond to MCNF problems with an additional bundle constraint that specifies the number of available resources. The addition of such constraint destroys the total unimodularity property of the constraint matrix and therefore integrality requirements must be explicitly enforced; nevertheless in our computational experiments, majority of the $IP(k)$'s solved at the root node of the branch and bound tree without adding any cutting planes to the formulation, that is, the linear programming relaxations tend to yield integral flows. However, about 7% of the IP's solved required either some branching or the addition of some cutting planes to attain integral flows. We suspect that the problem of computing the interior points of the frontier may be NP-hard, though we do not have a formal proof for such a result.

3.3.2.2 Solving the incremental problem using time-expanded networks

Given x_k, an optimal solution to $IP(k),\ k \geq k_{min}$, how to compute x_{k+1}? The natural extension from the approach used in the aperiodic case to solve the incremental problem would be identify the most negative cycle in $\mathcal{G}(x) = (\mathcal{N}^{\mathcal{R}}, \mathcal{A}^{\mathcal{R}})$, which is an NP-hard problem in general [2]. In any case, even if such a cycle could be found efficiently, it this approach would still fail to provide a solution procedure for the incremental problem in the periodic case. To exemplify why, consider the following instance involving three scheduled requests: $\alpha = (1,2,0,1,1), \beta = (3,4,2,3,1), \gamma = (5,6,4,5,1)$, with repositioning times $t_{ij} = |i-j|, \forall\ i \neq j \in 1,2,\ldots,6$. Figure 12 illustrates how augmenting a unit of flow along a single cycle in the residual network of a periodic schedule can increase the fleet size more than one unit. For the sake of clarity only the arcs with nonzero flows are depicted. The top network illustrates the flow that minimizes repositioning using a single resource; the middle network shows the most negative cycle in the residual network; and the bottom network shows the flow that results by augmenting one unit of flow along such cycle.

Thus, the solution to the incremental problem requires finding the most negative circulation in the residual network so that the flow across a temporal cut equals 1. Such a circulation can be found with an IP formulation similar to $IP(k)$.

3.3.2.3 Computing the Pareto points with bipartite networks

The Pareto points can alternatively be computed using the bipartite-network framework by including additional side constraints to enforce the periodicity of the resource schedules. Let

$$E_R^k(l) = \{(\upsilon_R, \upsilon_T) \in E^k : \upsilon_R \in V_{R_{start}}, \upsilon_T \in V_{T_{request}} \text{ and is associated with terminal } l\ \}$$

$$E_T^k(l) = \{(\upsilon_R, \upsilon_T) \in E^k : \upsilon_R \in V_{R_{request}} \text{ and is associated with terminal } l\ , \upsilon_T \in V_{T_{end}}\}$$

Then, the computation of the Pareto points involves the same minimum perfect matching problems used for the aperiodic case with the following additional constraints: $\left|E_R^k(l) \cap M\right| = \left|E_T^k(l) \cap M\right|, \forall\ l \in \mathcal{L}$. Note that these alternative models do not offer any advantage over the IP formulations on time-expanded networks since polynomial-time matching algorithms do not handle side constraints. The same applies to the incremental problem since no simple path algorithm can guarantee the satisfaction of the additional constraints.

3.4 Use of the efficiency frontiers in practice

This section discusses some of the issues that arise when using the efficiency frontiers of fleet size versus repositioning in practice. In particular, we will discuss a couple of

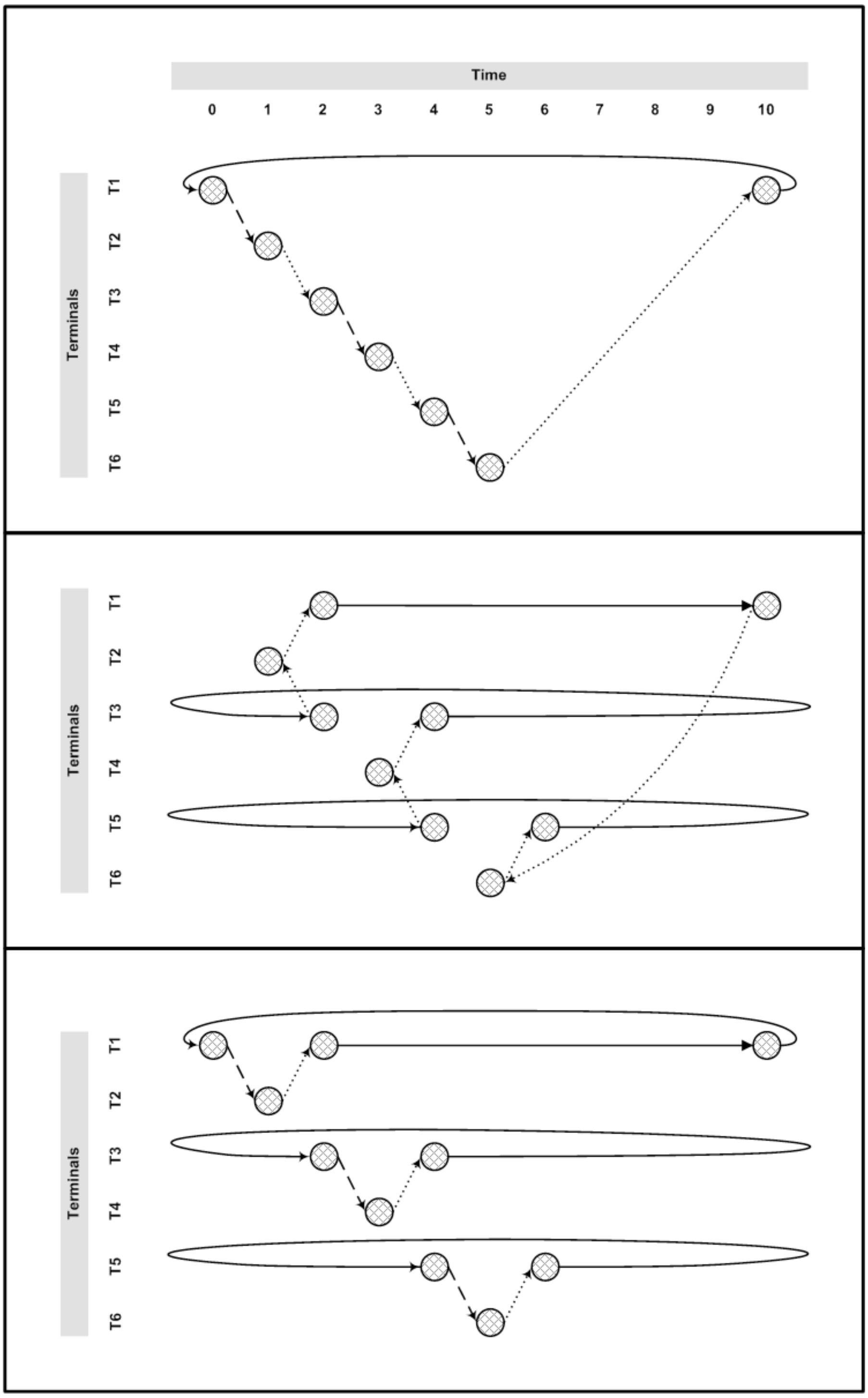


Figure 12: Flow augmentation along a single cycle

practical applications for which the information from the frontiers is useful, we will contrast the practical use of both types of modeling frameworks, and we will present the results of some computational experiments from a real instance.

3.4.1 Practical applications

The framework presented in this chapter can be used to assist transportation operators in tactical and operational decision making. The following couple of examples illustrate how.

On the tactical side, consider the case of LTL trucking carriers. Tractor fleet sizing is a tactical decision problem for these carriers because monthly or quarterly adjustments to the fleet size are appropriate in practice; therefore, under different economic scenarios, they can take advantage of the fleet size and repositioning trade-off to help them decrease their total costs; for instance, rises in fuel prices, such as the ones experienced in Fall of 2008 due to shortages in fuel production, might dictate enlarging the tractor fleet size as viable alternative to reduce the mileage of repositioning moves and thereby reduce the operating costs; conversely, a decline in the volumes of quantities shipped throughout a transportation network, as the ones experienced in 2008 and 2009 due to the economic recession, might imply that decreasing the number of tractors (and making up for them by increasing deadheading moves) can help the carrier lower its overhead costs and remain competitive. In this setting, the fixed schedule assumption is justified because carriers normally make fleet sizing decisions using actual historical dispatch data for a recent month or quarter (for instance, an average-demand or peak-demand month or quarter during the time period since the previous fleet size adjustment).

On the operational side, consider the case of a bus company such as the ones described in [25, 9, 23]. The sizes of bus fleets are generally determined by the number of units that are required during the busiest periods (this problem itself may benefit at the tactical level from the frontier information); however, for the non-peak periods, the company has a choice of how many units to utilize to fulfill its published (and thus fixed) schedules.

In the following subsections we contrast the two modeling frameworks that we have presented and we show the results of some computational experiments on the time required to compute all Pareto points using different strategies. The data used in both cases correspond to historical dispatch information from a major national LTL carrier which operates 350 terminals across the U.S. This carrier was interested in determining the correct tractor fleet size it should operate as well as the impact of repositioning on such a decision. The input data have the following characteristics:

- The loaded requests span four weeks of operations (considered to be representative of the level of activity the carrier usually experiences) and include 115,140 scheduled tractor dispatches.
- The scheduled trailer moves correspond to complete tours, which means that all of the terminals are balanced, that is, the the total number of scheduled requests out of a terminal equals the total number of scheduled requests into the same terminal.
- Repositioning moves can only occur among 135 out of the 350 terminals; they cannot take longer than 11 hours; and they are restricted to at most 2 per day, with fixed departure times.

3.4.2 Comparison of modeling frameworks

As it was presented in Section 3.3, both the time-expanded networks and the bipartite networks can in principle be used to define models to individually compute each of the Pareto points, and to compute adjacent Pareto points. In fact, most of the results shown for both modeling frameworks parallel each other even though they were derived independently.

Furthermore, both types of networks can be modified easily to handle fixed schedules with any or all of the following characteristics: 1) some of the scheduled requests start or end outside of the planning horizon, 2) repositioning of resources can occur only among a subset of the terminals, 3) maximum repositioning times are imposed , or 4) repositioning moves can be performed only at fixed times throughout the planning horizon. In fact, the computational experiments presented in this section are based on data which encompasses all of these characteristics.

In contrast, there are some important differences between the sizes of the two different types of networks. Given a fixed schedule of requests, the bipartite networks tend to require fewer nodes but significantly more edges than the time-expanded networks; moreover, the edge set of the bipartite networks grows at a much faster rate than the arc set of the time-expanded networks as the planning horizon grows. In both types of networks, the main decision to be made for each available resource is the next scheduled request it will fulfill out of all the requests for which the resource can be feasibly repositioned. In the case of bipartite-networks, this decision is explicitly modeled and each of the edges represents a feasible assignment of the next request to fulfill. There exists a correspondence between edges in the bipartite network and paths in the time-expanded network since assigning a resource to a specific task is equivalent to repositioning the resource to the desired terminal and keeping it in inventory until the departure time of the request, so to represent a potential assignment the bipartite-network requires one edge and the time-expanded network at least two arcs; however, due to the fact that many of the paths associated with edges in the bipartite-network overlap and arcs in the time-expanded network can carry any nonnegative flow, significantly fewer arcs are required to represent all the feasible assignments. The overlapping of paths makes it necessary to decompose the flows in the end to determine the specific schedules for each resource, but that is a minor disadvantage. Table 4 presents a summary of the sizes of the networks required to model different portions of the data set introduced before.

Table 4: Growth of the network sizes

		Length of the planning horizon			
		1 week	2 weeks	3 weeks	4 weeks
Scheduled Dispatches		29,922	57,985	86,697	115,140
Bipartite network $G^{k_{min}}$	# Nodes	64,568	120,926	178,602	235,714
	# Edges	230,964,344	524,122,592	1,028,068,471	1,706,614,773
Aperiodic time-expanded network	# Nodes	112,911	222,248	332,333	442,128
	# Arcs	206,683	410,989	616,692	821,836

Table 4 shows that even though from a theoretical perspective both modeling frameworks can be used to compute the Pareto points, from a computational standpoint, the bipartite network framework is impractical.

3.4.3 Running times to compute all Pareto points

In order to evaluate the savings in running time that can result from computing all the Pareto points using the incremental procedure outlined in Algorithm 1 as opposed to solving an independent linear program each time, we computed all the Pareto points for the national LTL carrier. We present the results considering the input data both as aperiodic and as periodic.

When we consider the input to be aperiodic, the computation of the end point $C = (k_{max}, r_{min})$ takes 20 seconds, and the computation of the end point $A = (k_{min}, r_{max})$ takes 77 seconds. For this input data, $k_{min} = 2,158$ and $k_{max} = 2,487$. Table 5 presents some statistics on the times required to compute the interior points $B_n = (k_{min} + n, r_n) \forall n \in \{1, 2, \ldots, 328\}$ using each of the following methods:

- **LP:** Solving each $LP(k)$ independently.
- **LP(WS):** Solving $LP(k)$ by warm starting it with the optimal basis for $LP(k-1)$ (i.e., a dual feasible basis) and using the dual simplex algorithm.
- **Inc:** Using Algorithm 1.

Table 5: Computing the interior points for aperiodic input

	LP	**LP(WS)**	**Inc**
Total Time	2.9 hr	14.18 min	6.87 min
Average Time	31.869 sec	2.587 sec	1.256 sec
Standard deviation	2.608 sec	1.119 sec	0.023 sec

Table 5 shows that solving the LP's with a warm start reduces the total time by 92%. The incremental procedure involving shortest paths yields an additional 52% reduction for total savings of 96%. In addition, the standard deviation of the time to compute a single point is also significantly reduced by 99% overall.

When we consider the input to be periodic, the computation of the end point $C = (k_{max}, r_{min})$ takes 18 seconds, and the computation of the end point $A = (k_{min}, r_{max})$ takes 89 seconds. For this input data, coincidentally $k_{min} = 2,158$ and $k_{max} = 2,487$ (this need not be the case in general). Table 6 presents some statistics on the times required to compute the interior points $B_n = (k_{min} + n, r_n) \forall n \in \{1, 2, \ldots, 328\}$ using each of the following methods:

- **IP:** Solving each $IP(k)$ independently.
- **IP(SV):** Solving $IP(k)$ by providing the optimal flow values to $IP(k-1)$ as starting values to the flow variables.

Table 6: Computing the interior points for aperiodic input

	IP	**IP(SV)**
Total time	6.45 hr	4.98 hr
Average time	70.776 sec	54.628 sec
Standard deviation	28.397 sec	11.182 sec

Table 5 shows that solving the IP's by providing starting values reduces the total time by 23% and the standard deviation of the time to compute a single point by 60%.

Finally, Figure 13 shows the smoothed optimal Pareto frontiers of fleet size (in number of tractors) versus repositioning (in tractor-minutes) when the input data is considered aperiodic and periodic. In both cases, k_{max} represents only a 15% fleet increase over k_{min} and both frontiers exhibit diminishing marginal returns as each additional tractor yields smaller repositioning savings. In relative terms, both frontiers exhibit similar results and asymptotic changes are observed near the end points of the frontiers. Increasing the fleet size 1%, 2% and 3% with respect to k_{min} respectively yields repositioning savings of 30%, 47% and 59% in the aperiodic case, and savings of 28%, 45%, and 57% in the periodic case. On the other hand, given that the company was more interested in reducing its fleet size, it is interesting to note that while reducing the tractor fleet size in 329 units might not be practically attainable, reducing the fleet size 25% and 50% of the total feasible reduction only requires respectively 3.75% and 13.2% of r_{max} in the periodic case and 4.71% and 14.77% of r_{max} in the periodic case.

A remark worth mentioning is that the input data includes scheduled requests involving both loaded and empty tractor moves. We expect the repositioning savings to be larger if those scheduled empty moves are removed from the input.

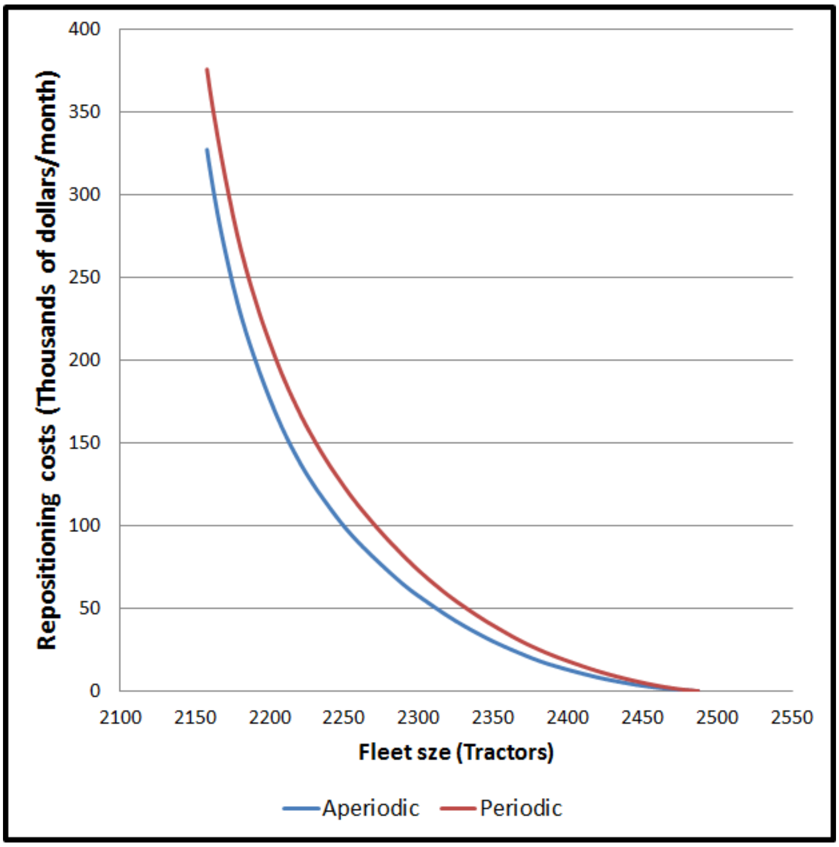


Figure 13: Optimal Pareto frontiers for a national LTL carrier

CHAPTER IV

ROBUST EMPTY REPOSITIONING IN VERY LARGE-SCALE FREIGHT CONSOLIDATION NETWORKS

The previous two chapters addressed transportation resource management problems involving costly resources such as tractors. In both cases, fleet sizing was an important component of the models because reducing the required owned or leased fleet size could have an important impact on profits. Furthermore, we were interested in evaluating the impact of fleet size reductions by using repositioning strategies that exploit regional changes in freight demand over the course of a monthly planning horizon and deploy tractors to different parts of the network at different times based on need. In that context, the use of known historical data corresponding to an average-demand or peak-demand month was appropriate. In this chapter, we turn our attention to dynamic empty-trailer repositioning problems arising in very large-scale freight consolidation networks. These problems require the explicit consideration of the dynamic and uncertain nature of the estimates of future trailer requirements to reduce the chances of trailer stockouts and increase service levels. Furthermore, since the focus is on operational decisions, we consider the fleet size as given.

In trucking operations, almost all carriers serve sets of loaded requests that are imbalanced in both time and space. Some customer regions are typically net resource attractors while others are net resource generators. Due to such imbalances, carriers need to move resources empty (*i.e.*, without serving a loaded request) between terminals. Furthermore, unlike other application contexts, trucking companies react to customer demands, but can do little to modify them, and information about customer locations and demand quantities and timing are all uncertain to some degree before the actual execution. Developing dynamic empty repositioning plans remains a major challenge for trucking transportation providers operating very large-scale consolidation networks because it involves the following key issues: (1) repositioning decisions are updated over time (daily, weekly, etc.); (2) uncertain demands for loaded resources are revealed over time; and (3) at each decision epoch, the number of empty resources available for repositioning and in transit depends on prior repositioning decisions and uncertain demands.

In practice, one approach used by sophisticated carriers to plan empty repositioning in advance is to solve deterministic network flow optimization models over time-expanded networks. Network nodes represent terminals at relevant points in time, and point forecasts of net supplies at the nodes, usually historical averages, are developed to estimate future empty trailers available or required at the different terminals. Network arcs represent planning decisions of trailer idling and trailer repositioning together with their corresponding costs. Feasible flows on such a network correspond to repositioning plans, and optimal cost flows can be found efficiently via linear programming or network optimization algorithms. These models are usually implemented in a rolling horizon framework in which the model for a weekly or monthly planning horizon is solved, but only a small subset of decisions (say, for the first day) is actually executed. Then the planning horizon is "rolled" and a new model is formulated and solved, and the process repeated. Rolling the horizon involves discarding the input for the first day, updating the state of the system based on the executed actions, and updating forecasts for net supplies, including new information for an additional day at the end of the horizon.

A major drawback of this approach is that there may be significant uncertainty around the point forecasts; this uncertainty may grow towards the end of the planning horizon, but is present even in the near future. Deterministic models provide no mechanism for building safety stocks of resources anywhere in the system. Thus, repositioning plans created with deterministic models may be at best suboptimal or at worst badly infeasible when the realized net supplies differ from their point forecasts. Infeasibility of the plans arises when insufficient resources are available to satisfy the loaded requests at some terminals and additional resources are too far away from the terminals where they are needed so that it is not possible to reposition them in time to serve customer demands. This, in turn, implies either not satisfying some of the customer loaded requests, negatively impacting service levels, or having to procure costly external resources to cover those requests. To address these drawbacks, effective planning must appropriately account for the dynamics and uncertainty of the resource imbalances over time and use approaches that explicitly hedge against this uncertainty. Explicitly incorporating the dynamics and uncertainty of future net supplies leads to a multistage optimization model in which (partial) information is revealed as time progresses.

A number of papers dealing with stochastic and dynamic variants of empty repositioning or related problems have been reported in the literature. Research along these lines has made use of models from stochastic programming, dynamic programming, and robust optimization. Stochastic models that focus on expected cost minimization for a dynamic resource allocation problem in truckload trucking are initially developed in [39] and [40]. Modeling approaches for stochastic empty container management problems are presented in [11]. Dynamic programming models together with effective approaches to approximate value functions for multistage problems are proposed in [22] and [10]. Most of the successful approaches for expected value minimization use a scenario-based approach, such as the adaptive dynamic programming approaches to approximating nonlinear value functions that have been applied to single and multicommodity problems [28, 29]. Another example of stochastic optimization for transportation resource management is the multi-scenario optimization model in [16] that addresses a container maritime-repositioning problem where several parameters are uncertain and deterministic models do not prove effective for decision-making. Each of the stochastic models described above assume that probabilistic information describing the future evolution of resource demands can be generated at a level of accuracy sufficient to warrant an expected value minimization objective. Furthermore, they assume that models can be built to appropriately capture all system costs and constraints, including costs of shortages. Formulating and solving large-scale multistage stochastic optimization problems to minimize expected costs can be a difficult task.

Alternative approaches that have received more attention recently rely on ideas from robust optimization. In particular, a two-stage robust optimization approach for solving network flow and design problems with uncertain demand is presented in [4], while a two-stage robust optimization framework for problems with right-hand-side uncertainty, and specifically for empty repositioning problems is reported in [21] and [35]. A primary motivation for the development of the approach in the latter references is that it is often not possible or advisable to trade off shortage costs with transportation costs, and therefore it may be more sensible to look for plans with low transportation costs that ensure that future shortages will be small or nonexistent. Additionally, the approach attempts to allow the decision-maker to control the conservatism of the plans generated; a more risk averse plan can be generated that ensures that shortages do not arise for a larger set of future scenarios. A key limitation of this existing work on robust optimization for dynamic and stochastic empty repositioning problems is that the approaches were not designed to handle large-scale networks, nor were they thoroughly developed for deployment within a rolling horizon framework. This is a major shortcoming, since robust optimization requires simpler input requirements to model forecast uncertainty and could lead to more computationally tractable models than stochastic programming or dynamic programming approaches. Similar in spirit to the ideas we develop in this chapter, [57] has recently proposed using a two-stage stochastic programming model iteratively to solve approximately a complex multistage stochastic optimization problem arising in the context of drayage operations.

The primary goal of the study in this chapter is to address this shortcoming, and study how to effectively use a two-stage robust optimization approach that attempts to control future resource shortages within a rolling horizon framework for very large-scale network applications. The main contributions are that (1) we develop approaches for embedding two-stage robust optimization models within a rolling horizon framework for dynamic empty repositioning, (2) we demonstrate that such approaches enable the solution of very large-scale instances that use real data from a national package/parcel express carrier and produce plans with significantly fewer unmet loaded requests and a modest increase in execution costs over those plans generated by deterministic optimization models, and (3) we show that less conservative implementations of robust optimization models (via a reduction of the planning horizon or via a simplification of the uncertainty sets against which protection is sought) are required within rolling horizon frameworks.

4.1 Empty repositioning problems

We consider an empty-trailer repositioning problem faced by a national parcel/express carrier, but the results will be valid for any transportation operator that uses a centralized planner to manage a homogeneous fleet of reusable resources such as containers, railroad cars or trucks. In parcel/express operations, empty trailers are required at outbound doors of the terminals to be filled with packages bound to other terminals in the linehaul network. Empty trailers are brought into service during the sorting times at the terminals, and remain unavailable while being loaded, transported, and finally unloaded at their destination terminal. Once they are empty they are brought to the terminal yard where they wait to be used again in the same terminal or to be transported to a different terminal with larger outbound activity. In this context, although execution costs of empty-trailer repositioning plans are important and must be kept low, service levels are a main concern. Terminals cannot afford to have shortages of empty trailers, since this will delay the outbound movement of sorted freight. Since terminals also have limited freight storage capability, this is a major concern. As such, methodologies used to develop empty repositioning plans must address the uncertainty in point forecasts of future net supplies at the terminals.

A traditional approach used by more sophisticated carriers for generating cost-effective repositioning plans is to use a deterministic minimum cost network flow model (MCNF) over a time-expanded network for some planning horizon. Assuming a planning horizon including $\tau + 1$ discrete periods, $\{0, 1, 2, \cdots, \tau\}$, a time-expanded network $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ can be constructed as follows. Let $\mathcal{D} =$ be the set of terminals (or depots) in the transportation network. Let n_t^d represent the time-space node corresponding to terminal d at time t, and let s be a sink node. The complete set of nodes is given by $\mathcal{N} = \left(\bigcup_{\forall t, \forall d} n_t^d\right) \bigcup s$. An integer $b(n_t^d)$ is associated with each node n_t^d to represent the forecast of the net supply of empty trailers that will be available at terminal d at time t. An integer $b(s) = -\sum_{n \in \mathcal{N} \setminus \{s\}} b(n)$ is assigned as the net supply at the sink to meet the feasibility condition of MCNF problems that the sum of all node supplies must equal zero. The network includes inventory arcs $\left(n_t^d, n_{t+1}^d\right), \forall d \in \mathcal{D}, \forall\ 0 \leq t \leq \tau - 1$, representing trailers held in inventory at a terminal from one time period to the next, and inventory arcs $\left(n_\tau^d, s\right), \forall d \in \mathcal{D}$, representing final inventories of trailers at each of the terminals. Costs associated with inventory arcs are usually ignored since they tend to be similar at the different terminals and much smaller than repositioning costs. The network also includes repositioning arcs $\left(n_t^i, n_{t+h}^j\right)$ with costs c_{ij} per trailer, which correspond to the potential move of empty trailers from terminal i at time t to terminal j at time $t + h_{ij}$; where h_{ij} is the repositioning travel time between terminals i and j. Let $\mathcal{A}^I$, and $\mathcal{A}^R$ denote the set of all inventory and repositioning arcs, respectively. Thus $\mathcal{A} = \mathcal{A}^I \cup \mathcal{A}^R$. The deterministic MCNF model is:

$$\begin{array}{lll} min & \sum_{a \in \mathcal{A}^R} c_a x_a & \\ s.t. & \sum_{a \in \delta^+(n)} x_a - \sum_{a \in \delta^-(n)} x_a = b(n) & \forall\ n \in \mathcal{N} \\ & x_a \geq 0 & \forall a \in \mathcal{A} \end{array}$$

The decision variables x_a represent the amount of flow on the inventory and repositioning arcs and correspond, respectively, to trailers remaining idle at the corresponding terminal, and trailers being repositioned to a different terminal. The optimal solution to this model can be found in polynomial time via linear programming or using well-known network flow algorithms. We will use repositioning plans generated with this deterministic minimum cost network flow model as a benchmark for the more sophisticated models that incorporate robust optimization ideas.

Optimization models used in empty-trailer repositioning should capture two important facts: first, partial information is revealed across time (i.e., different "stages" should represent points in time where information is updated or new information is revealed); and second, the generated plans must hedge against uncertain future trailer requirements at the different terminals to ensure appropriate service levels. These requirements point to the need for a multistage network flow optimization problem over a time-expanded network in which the net supplies at the nodes are uncertain and revealed dynamically at different stages in time. Over some fixed time horizon, there is a notion of an *a posteriori* optimal solution for each realization of the net supplies; if a feasible flow exists for each realization, the deterministic MCNF problem given earlier could be used to find the optimal *a posteriori* flow. Since a full realization of the net supplies is not available at any decision stage, one potential optimization approach is to find flows at each decision stage that minimize total expected costs over a planning horizon, including cost penalties for unmet demands (stochastic programming). Another approach would be to appropriately capture all relevant costs and instead find flows that minimize total costs, but also satisfy bounds and flow-balance constraints to some specified probability (chance-constrained optimization). A third approach is to find low-cost flows at each stage that can be modified via a limited set of future decisions to recover feasibility for all possible outcomes, or a meaningful subset, of the uncertain net supplies (robust optimization).

Given the complexity of the true multistage empty repositioning problem, simpler models have been developed to generate very good approximate solutions. In particular, two-stage robust optimization models that use simple net supply interval forecasts have been studied by [21]. The models proposed in that work assume the following sequence of events takes place: a repositioning plan is constructed, then all the uncertain information in the planning horizon is revealed at once, and then a limited set of recourse or recovery decisions are available to modify the plans and recover feasibility in a single pass. The key ideas will be presented next.

4.2 Two-stage robust empty repositioning

This section summarizes some of the main results of the two-stage robust optimization approach for empty repositioning problems developed by [21]. First of all, the uncertainty of the net supplies at the different time-space nodes is modeled with symmetric intervals around the point forecasts $b(n)$, i.e., $\left[b(n) - \widehat{b}(n), b(n) + \widehat{b}(n)\right]$, where $\widehat{b}(n) \geq 0$. The realized net supply at $n \in \mathcal{N}$, $\widetilde{b}(n)$ is assumed to fall within the interval, and the net supplies in the initial period are assumed to be certain, thus $\widehat{b}(n_0^d) = 0,\ \forall d \in \mathcal{D}$.

To allow control of the conservatism of the robust repositioning models, a parameter k specifies the maximum number of net supplies that may simultaneously take an extreme value in a realization. A value of $k = 0$ corresponds to absolute certainty in which every net supply realization is assumed to conform to its point forecast, whereas a value of $k = \infty$ corresponds to absolute uncertainty in which all net supplies could take their worst case values.

This two-stage approach seeks to find a minimum cost repositioning plan that (1) satisfies flow bounds and balance equalities for the nominal net supply values, and (2) is recoverable for every joint realization in which each time-space node net supply value lies within its interval and no more than k values simultaneously take their worst-case value. A plan is recoverable if there exists a set of recovery actions that can transform the plan such that it satisfies flow bounds and balance equalities given the realized net supply values. The recovery actions are usually a limited subset of future repositioning movement decisions which include only low-cost options movements used to recover feasibility.

Let φ^k be a limited perturbation set as a function of the uncertainty-budget parameter k defined as follows:

$$\varphi^k = \left\{ \begin{array}{ll} \Delta \in \mathbb{Z}_+ : & \Delta(n) = \hat{b}(n)z(n), \sum_{n\in\mathcal{N}\backslash s} |z(n)| \leq k, |z(n)| \leq 1 \ \forall n \in \mathcal{N}\backslash s, \\ & \Delta(s) = -\sum_{n\in\mathcal{N}\backslash\{s\}} \Delta(n) \end{array} \right\}$$

Assuming that each realization of interest is given by $b + \Delta$, for some $\Delta \in \varphi^k$, the following integer programming problem can be solved to generate a k-robust empty repositioning plan, *i.e.,* a set of flows that satisfy the nominal net supplies, and for which recovery actions exist to modify the flows and recover feasibility for all net supply realizations in φ^k. This formulation (1) minimizes the total repositioning costs during the first stage, subject to constraints for first and second-stage decisions. First-stage flows are required to satisfy (2) flow balance and (3) nonnegativity constraints. Second-stage flows require sets of constraints for each possible realization of the net supplies against which protection is sought. These constraints include (4) balancing the difference between the realized and nominal net supplies at each node, (5) honoring integrality requirements of the overall resulting flows (first- plus second-stage decisions), and (6) ensuring nonnegativity of the recovery actions on repositioning arcs.

$$\begin{array}{llll}
min & \sum_{a\in\mathcal{A}^R} c_a x_a & & (1)\\
s.t. & \sum_{a\in\delta^+(n)} x_a - \sum_{a\in\delta^-(n)} x_a = b(n) & \forall\ n \in \mathcal{N} & (2)\\
 & x_a \in \mathbb{Z}_+ & \forall a \in \mathcal{A} & (3)\\
 & \sum_{a\in\delta^+(n)} w_a^\Delta - \sum_{a\in\delta^-(n)} w_a^\Delta = \Delta(n) & \forall\ n \in \mathcal{N},\ \forall\ \Delta \in \varphi^k & (4)\\
 & x_a + w_a^\Delta \in \mathbb{Z}_+ & \forall a \in \mathcal{A},\ \forall \Delta \in \varphi^k & (5)\\
 & w_a^\Delta = 0 & \forall a \notin \mathcal{A}^W,\ \forall \Delta \in \varphi^k & (6)\\
 & w_a^\Delta \geq 0 & \forall a \in \mathcal{A}^R,\ \forall \Delta \in \varphi^k & (7)
\end{array}$$

where $\mathcal{A}^W$ is the set of arcs in which recovery decisions are allowed to be nonzero.

A major drawback of the previous model is the fact that it requires a set of constraints (4-6) for each possible net supply realization. Nevertheless, a number of concepts were developed to set up an equivalent integer programming program whose number of constraints is independent of the size of the uncertainty set.

The following are three important concepts that aided in such transformation:

Definition 1. ***RECOVERY NETWORK***. *A recovery network $G_R = (N_R, A_R)$ can be defined whose node set N_R is the same as the node set $\mathcal{N}$, and whose arc set A_R contains all inventory arcs in $\mathcal{A}^I$ and all repositioning arcs in $\mathcal{A}^R$ on which recovery flow is permitted to be nonzero.*

Definition 2. ***NODE SET VULNERABILITY***. *For a set of nodes $C \subset \mathcal{N}$, its vulnerability $\vartheta(C, k)$ is defined as*

$$\vartheta(C,k) = max_z \left\{ \sum_{n\in C} \hat{b}(n)z(n) : \sum_{n\in C} |z(n)| \leq k, |z(n)| \leq 1, \forall\ n \in C \right\}$$

Definition 3. ***INBOUND-CLOSED (IBC) NODE SET***. *A set of nodes $U \subset N_R$ is inbound closed if there exists no arc in A_R from any node $i \in N_R \backslash U$ to any node $j \in U$.*

The main result we will use is that solving the following integer programming problem is equivalent to solving the original formulation that enforced constraints for each realization of interest. Thus, the following IP will produce a k-robust empty repositioning plan, *i.e.,* a set of trailer flows that satisfy the nominal net supplies, and for which recovery actions exist to modify the flows and recover feasibility for all net supply realizations in which at most k nodes take their worst-case value at the same time.

$$\begin{array}{llll} min & \sum_{a\in\mathcal{A}^R} c_a x_a & & (1)\\ s.t. & \sum_{a\in\delta^+(n)} x_a - \sum_{a\in\delta^-(n)} x_a = b(n) & \forall\ n \in \mathcal{N} & (2)\\ & \sum_{a\in\delta^+(U)\cap\mathcal{A}^I} x_a \geq \vartheta(U,k) & \forall\ U \subset \mathcal{N} : U \text{ is inbound closed in } G_R & (3)\\ & x_a \in \mathbb{Z}_+ & \forall a \in \mathcal{A} & (4) \end{array}$$

This formulation (1) minimizes the total repositioning costs, subject to (2) flow-balance constraints, (3) robust constraints that establish that the flow on inventory arcs leaving an inbound closed set has to be at least equal to the vulnerability of the set (*i.e.*, enough inventory within the set must exist to hedge against demand surges as no further reactive repositioning can be used to bring extra resources into the set), and (4) flow integrality requirements.

We end this section with some additional remarks. First, one approach with practical appeal designates *a priori* some terminals that serve only as providers of reactive resources, and some that serve only as recipients. In the remainder of this chapter we will focus on recovery networks that include this structure. We will refer to the providers of reactive resources as empty hubs, and to the recipients as non-hubs. We will allow recipients to be assigned to only one empty hub. Second, in this work we will control solution conservatism without using parameter k and will instead employ different approaches, as will be explained in the following sections. Thus, for the rest of the chapter parameter k will be set to ∞; this assumption simplifies the definition of the vulnerability $\vartheta(C)$ of a set of nodes $C \subset \mathcal{N}$ to $\vartheta(C) = \sum_{n\in C} \hat{b}(n)$.

4.3 Dynamic empty repositioning in very large-scale networks

As mentioned earlier, the empty trailer repositioning problem involves information that is revealed and/or updated dynamically, and as such, leads to a multistage optimization model, in which the state of the system (numbers of trailers at the different terminals and the number of trailers in transit) is updated at different stages. The size of the network is a very important complicating factor because some of the models that have been proposed for such problems do not scale well when applied to very-large scale networks involving hundreds of terminals and hundreds of thousands of repositioning opportunities. We will explore how to apply the results from two-stage robust optimization models implemented within a rolling planning horizon to approximate this multistage problem and compare the resulting plans against those generated by simpler deterministic models. We will explore two alternatives: (1) A two-stage robust optimization approximation and (2) A two-stage robust optimization approximation with consideration of the rolling horizon implementation.

The two-stage robust optimization approach described in Section 4.2 was envisioned and tested on networks resulting from the operations of a major tank container fleet operator. In that application, the reactive repositioning sharing groups (an empty hub and its assigned non-hubs) arose naturally by geography and were small (at most three terminals in the sharing group). As a result, the number of IBC node sets in the recovery network did not get too large and the corresponding flow-bundling robustness constraints could be explicitly added into the integer programming formulation to generate the desired empty repositioning plans. However this approach becomes intractable as the networks and the number of terminals in a sharing group gets larger.

Lemma 3. *The number of IBC sets in a connected component of the recovery network G_R involving hub j and a set $\mathcal{L} \subset \mathcal{D}$ of its assigned non-hub terminals (with $|\mathcal{L}| = L$)*

over a planning horizon involving $\tau+1$ *discrete periods* $\{0, 1, 2, ..., \tau\}$ *is bounded below by* $(\tau+2)^L + \tau$ *and bounded above by* $(\tau+2)^L(\tau+1)$.

Proof. Let $N^d_\rho = \left\{n^d_0, n^d_1, \ldots, n^d_\rho\right\}$ for each $d \in \mathcal{D}$ be the ordered set of nodes n^d_t associated with terminal d from time $t = 0$ to time $t = \rho$. There are $\tau + 1$ IBC sets involving involving only the hub j, namely $\left\{N^j_0, N^j_1, \ldots, N^j_\tau\right\}$. Additionally, given a subset $\mathcal{I} \subseteq \mathcal{L}$ of non-hubs, an IBC set has the form $U = \bigcup_{i\in\mathcal{I}} N^i_{\rho(i)} \bigcup N^j_{\rho(j)}$ where $0 \leq \rho(i) \leq \tau$ and $max_i \{0, \rho(i) - h_{ji}\} \leq \rho(j) \leq \tau$. Thus, given $\mathcal{I}$, there are at least $(\tau+1)^{|\mathcal{I}|}$ different IBC sets, and there are at most $(\tau+1)^{|\mathcal{I}|}(\tau+1)$. Furthermore, there are $\binom{L}{\ell}$ different subsets with ℓ non-hubs. Therefore, it follows that:

$$LB = \tau + 1 + \sum_{\ell=1}^{L} \binom{L}{\ell} (\tau+1)^\ell = (\tau+2)^L + \tau$$

$$UB = \tau + 1 + (\tau+1) \sum_{\ell=1}^{L} \binom{L}{\ell} (\tau+1)^\ell = (\tau+2)^L(\tau+1)$$

□

Lemma 3 shows that the number of IBC sets grows exponentially with the number of non-hubs in a sharing group; thus, for very-large scale networks it will be infeasible to add all the flow bundle constraints required in the robust IP. More importantly, given that these models are usually implemented in a rolling-horizon framework, not all the robust constraints in the planning horizon are needed to guarantee the feasibility of the model that will be solved the next period (say, the next day). This situation is illustrated with the following example.

Consider the time-expanded network described before, but with repositioning opportunities among all pairs of terminals, all of which take one period. Also assume that the net supplies $\tilde{b}^t_i$ at terminal i at time t are computed from loaded moves between pairs of terminals, all of which take one period as well, and whose trailer requirements are such that $\tilde{b}^t_i \in \left[b^t_i - \hat{b}^t_i, b^t_i + \hat{b}^t_i\right]$. Using the decision variables $x^t_{ij} =$

the number of trailers repositioned from terminal i to terminal j at the start of time t, the multi-stage optimization model for this situation is given by:

$$
\begin{array}{llll}
min & \sum_{t=0}^{\tau-1} \sum_{(i,j)\in\mathcal{A}^R} c_{ij}x_{ij}^t & & (1)\\
s.t. & \sum_{\{j:(i,j)\in\mathcal{A}^R\}} x_{ij}^t \leq \tilde{b}_i^t & \forall\ i \in \mathcal{D}, \forall\ 0 \leq t \leq \tau - 1 & (2)\\
 & \sum_{\{j:(j,i)\in\mathcal{A}^R\}} x_{ji}^t \geq \tilde{b}_i^{t+1} & \forall\ i \in \mathcal{D}, \forall\ 0 \leq t \leq \tau - 1 & (3)\\
 & x_{ij}^t \geq 0 & \forall\ (i,j) \in \mathcal{A}^R, \forall\ 0 \leq t \leq \tau - 1 & (4)
\end{array}
$$

where $\tilde{b}_i^t \in \left[b_i^t - \hat{b}_i^t, b_i^t + \hat{b}_i^t\right]$ is revealed with certainty only at time $t-1$.

This formulation (1) minimizes the total repositioning over the planning horizon subject to: (2) flow out of a terminal must not exceed its supply at time t, (3) flow into a terminal must be at least its demand at time $t+1$, and (4) flows must be integer.

In this situation it is not difficult to see that, from a feasibility perspective, all one needs to do is guarantee that the necessary trailers will be available in the next immediate period (*i.e.*, the only IBC sets that need to be considered are those spanning periods t and $t+1$). Enforcing further protection into the future at stage t is not required because regardless of the demand realizations for future periods, in stage $t+1$ there will be a new opportunity to reposition resources in response to newly revealed needs. Similar, but more complex arguments can hold for more complicated settings in which not all repositioning moves are allowed and repositioning times take more that one period. They key is that in a rolling horizon implementation, when the horizon is rolled forward, new decision opportunities become available, and as such, in the model for stage t, the robust constraints become less and less relevant for feasibility when they involve periods further into the future. Although, it may still be useful to add some set of robust constraints in the future since they should help control the costs of responding to realized net supplies that deviate from their nominal prediction

4.3.1 Plan Generation

The original two-stage robust optimization idea requires that robust constraints be constructed that protect against joint uncertain outcomes involving the empty hub and all subsets of non-hubs in its influence sharing group. However, based on the previous observations, this is computationally prohibitive for very large-scale networks and might not be required. We will now outline two different approaches to generate empty repositioning plans that use results from the two-stage robust model, but are computationally tractable.

4.3.1.1 Two-stage robust-optimization approximation

A feasible alternative to deploy the two-stage robust optimization approach for the empty-trailer repositioning problem is to limit the number of IBC sets in the recovery network for which robust constraints are included. We will introduce both a robustness horizon $T_R \leq \tau$ and a limit L on the number of non-hubs that can participate in the IBC set. The robustness horizon limits the number of periods into the future for which robustness constraints are added based on the observation that periods further into the future become less relevant for protection against uncertainty because new decisions will become available once the horizon is rolled. On the other hand, limiting the number of non-hubs participating in an IBC set is also intended to reduce the combinatorial explosion in the robustness constraints. The assumption behind this idea is that serious empty resource deficits will occur at no more than a few terminals in each sharing group on any given period. These two parameters also provide a mechanism to control the conservatism because larger values of T_R or L increase the number of uncertainty sets for which the resulting plan is recoverable. The resulting model to be solved each period is given by:

The following formulation (1) minimizes the total repositioning costs, subject to

(2) flow-balance constraints, (3) robust constraints that establish that the flow on inventory arcs leaving an inbound closed set has to be at least equal to the vulnerability of the set, and (4) flow-integrality requirements. The constraints on the IBC sets in the recovery network for which robust constraints (3) are created enforce the robustness horizon $T_R \leq \tau$ and the limit L on the number of non-hubs that can participate in an IBC set.

$$\begin{aligned}
&min \quad \textstyle\sum_{a \in \mathcal{A}^R} c_a x_a && (1)\\
&s.t. \quad \textstyle\sum_{a \in \delta^+(n)} x_a - \sum_{a \in \delta^-(n)} x_a = b(n) \quad \forall\, n \in \mathcal{N} && (2)\\
&\qquad \textstyle\sum_{a \in \delta^+(U) \cap \mathcal{A}^I} x_a \geq \sum_{n \in U} \hat{b}(n) && (3)\\
&\qquad\quad \forall\, U \subset \mathcal{N}: \quad U = \textstyle\bigcup_{i \in \mathcal{I}} N^i_{\rho(i)} \bigcup N^j_{\rho(j)} \text{ where } j \text{ is an empty hub,}\\
&\qquad\qquad \mathcal{I} \text{ is a subset of non-hubs assigned to } j \text{ s.t. } |\mathcal{I}| \leq L\\
&\qquad\qquad 0 \leq \rho(i) \leq T_R, \max_i \{0, \rho(i) - h_{ji}\} \leq \rho(j) \leq T_R\\
&\qquad x_a \in \mathbb{Z}_+ \quad \forall a \in \mathcal{A} && (4)
\end{aligned}$$

4.3.1.2 Two-stage robust-optimization approximation with rolling horizon considerations

The second approach to deploy a two-stage robust optimization is also based on the explicit consideration that the empty repositioning models will be implemented in a rolling-horizon framework. Consider a generic dynamic, stochastic planning problem deployed using a rolling horizon of length τ. Let ζ^t_s be the parameters used in the model spanning time t to time $t-1+\tau$ (cost and constraint coefficients, right-hand-sides, etc.) estimated at time s, and let x^t be the decisions that will be fixed at stage t. At stage t, the problem solved is:

$$\begin{aligned}
&min \quad f(\zeta^t_t, x^t, x^{t+1}, \ldots, x^{t-1+\tau})\\
&s.t. \quad g_j(\zeta^t_t, x^t, x^{t+1}, \ldots, x^{t-1+\tau}) \geq 0 \quad \forall\, j = 1, 2, \ldots, m
\end{aligned}$$

At stage $t+1$, the horizon is rolled, the input of the model is updated based on the actions x^t, estimates for parameters occurring in $t+1$ through $t-1+\tau$ are refined, and new information for $t+\tau$ is estimated. The model for stage $t+1$ becomes:

$$\begin{aligned} min \quad & f\left(\zeta_{t+1}^{t+1}, x^{t+1}, x^{t+2}, \ldots, x^{t+\tau}\right) \\ s.t. \quad & g_j\left(\zeta_{t+1}^{t+1}, x^{t+1}, x^{t+2}, \ldots, x^{t+\tau}\right) \geq 0 \quad \forall\, j=1,2,\ldots,m \end{aligned}$$

A model with explicit consideration of the rolling horizon implementation includes the constraints that appear in models corresponding to future stages within the current planning horizon:

$$\begin{array}{llcl} min & f\left(\zeta_t^t, x^t, x^{t+1}, \ldots, x^{t-1+\tau}\right) & & \\ s.t. & g_j\left(\zeta_t^t, x^t, x^{t+1}, \ldots, x^{t-2+\tau}, x^{t-1+\tau}\right) & \geq & 0 \quad \forall\, j=1,2,\ldots,m \\ & g_j\left(\zeta_t^{t+1}, x^{t+1}, x^{t+2}, \ldots, x^{t-1+\tau}, 0\right) & \geq & 0 \quad \forall\, j=1,2,\ldots,m \\ & & \vdots & \\ & g_j\left(\zeta_t^{t+\tau-1}, x^{t+\tau-1}, x^{t-1+\tau}, \ldots, 0, 0\right) & \geq & 0 \quad \forall\, j=1,2,\ldots,m \\ & g_j\left(\zeta_t^{t+\tau}, x^{t-1+\tau}, 0, \ldots, 0, 0\right) & \geq & 0 \quad \forall\, j=1,2,\ldots,m \end{array}$$

In this formulation, the objective function and the first set of constraints correspond to the original model solved at stage t, the rest of the sets of constraints appear in models that correspond to future stages within the current planning horizon. The values of the parameters in those sets of constraints are estimates available at time t, and the variables appearing in those constraints representing decisions beyond the current horizon are all set to zero.

In the context of empty repositioning, let $\mathcal{G}(t) = (\mathcal{N}(t), \mathcal{A}(t))$ be the time-expanded network associated with the model solved at stage t. Explicitly incorporating the rolling-horizon at each stage yields the following model for stage t:

$$
\begin{aligned}
min \quad & \textstyle\sum_{a \in \mathcal{A}^R(t)} c_a x_a & (1)\\
s.t. \quad & \textstyle\sum_{a \in \delta^+(n)} x_a - \sum_{a \in \delta^-(n)} x_a = b(n) \quad \forall\, n \in \mathcal{N}(t) & (2)\\
& \textstyle\sum_{a \in \delta^+(U) \cap \mathcal{A}^I(t)} x_a \geq \sum_{n \in U} \hat{b}(n) & (3)\\
& \quad \forall\, t \leq s \leq \tau - 1, & \\
& \quad \forall\, U \subset \mathcal{N}(s): \quad U = \textstyle\bigcup_{i \in \mathcal{I}} N^i_{s,\rho(i)} \bigcup N^j_{s,\rho(j)} \text{ where } j \text{ is an empty hub,} & \\
& \quad\quad \mathcal{I} \text{ is a subset of non-hubs assigned to } j \text{ s.t. } |\mathcal{I}| \leq L & \\
& \quad\quad s \leq \rho(i) \leq \min\left\{s - 1 + T_R, t - 1 + \tau\right\}, & \\
& \quad\quad \max_i \left\{s, \rho(i) - h_{ji}\right\} \leq \rho(j) \leq min\left\{s - 1 + T_R, t - 1 + \tau\right\} & \\
& x_a \in \mathbb{Z}_+ \quad \forall a \in \mathcal{A}(t) & (4)
\end{aligned}
$$

where we have generalized the definition of N^d_ρ to $N^d_{s,\rho} = \left\{n^d_s, n^d_{s+1}, \ldots, n^d_\rho\right\}$ for each $d \in \mathcal{D}$. Thus, $N^d_\rho = N^d_{0,\rho}$.

This formulation (1) minimizes the total repositioning costs, subject to (2) flow-balance constraints, (3) robust constraints that establish that the flow on inventory arcs leaving an inbound closed set has to be at least equal to the vulnerability of the set, and (4) flow-integrality requirements. The constraints on the IBC sets in the recovery network for which robust constraints (3) are created enforce the robustness horizon $T_R \leq \tau$ and the limit L on the number of non-hubs that can participate in an IBC set, and they also enforce the inclusion of robust constraints that appear in models corresponding to future stages within the current planning horizon. If the interval forecasts for the net supplies remain static (*i.e.,* they are not updated at new stages), then the additional constraints that will be added at stage t are exactly the same constraints that will appear in models corresponding to future stages within the current planning horizon. Otherwise, the right-hand-sides will be different in future models.

4.3.2 Plan Evaluation

Given that we cannot find optimal solutions to the multistage empty repositioning problem, in order to evaluate the performance of the resulting repositioning plans, we develop and use a simulation that mimics the daily generation and execution of repositioning plans over a long horizon. The models that will be used for the simulation are slightly different from the ones outlined above. Two main differences are: (1) demand arcs will be explicitly modeled (corresponding to demand requests that generate net supplies in the previous models), and (2) additional variables will be used to capture the demands that cannot be met. A large penalty M will be used as a cost for unmet demands in such a way that the model will only allow unmet demands when it is infeasible to meet them. This approach aligns with the application context in which service levels are a major driver of empty-trailer repositioning. Unmet demands will only be allowed during the portion of the horizon that is assumed to be known with certainty, $T_K \leq \tau$. Note that this model could accommodate actual costs of unmet demands if such estimates exist. Let $x_a =$ the flow on arc a $\forall a \in \mathcal{A}$, and $y_a =$ the unmet demand (in number of trailers) on arc a, $\forall a = \left(n_t^i, n_{t'}^j\right) \in \mathcal{A}^D$ $s.t.$ $t \leq T_K$. The formulation is:

$$
\begin{array}{lll}
Min & \sum_{a \in \mathcal{A}^R} c_a x_a + \sum_{\left(n_t^i, n_{t'}^j\right) \in \mathcal{A}^D : t \leq T_K} M y_a & (1) \\
\text{s.t.} & \sum_{a \in \delta^+(n)} x_a - \sum_{a \in \delta^-(n)} x_a = \dot{b}(n) \qquad \forall n \in \mathcal{N} & (2) \\
& x_a + y_a = D_a \qquad \forall a = \left(n_t^i, n_{t'}^j\right) \in \mathcal{A}^D : t \leq T_K & (3) \\
& x_a = D_a \qquad \forall a = \left(n_t^i, n_{t'}^j\right) \in \mathcal{A}^D : t > T_K & (4) \\
& \sum_{a \in \delta^+(U) \bigcap \mathcal{A}^I} x_a \geq \sum_{n \in U} \hat{b}(n) \qquad \forall U \subseteq \mathcal{N} : U \in X & (5) \\
& x_a \in \mathbb{Z}_+ \qquad \forall a \in \mathcal{A} & (6) \\
& y_a \in \mathbb{Z}_+ \qquad \forall a = \left(n_t^i, n_{t'}^j\right) \in \mathcal{A}^D : t \leq T_K & (7)
\end{array}
$$

In this formulation, $\dot{b}(n) \geq 0,\ n \neq s$ corresponds to the initial state of the system, *i.e.*, the number of resources available at the given node based on decisions made prior to the horizon start of the model. Additionally, parameter $D_a,\ \forall a \in \mathcal{A}^D$ represents the number of trailers required to satisfy the loaded request corresponding to arc a, and set X captures the specific constraints on the IBC sets depending on the approach used to generate the plans.

This formulation (1) minimizes the total repositioning costs plus the penalties associated with unmet demands, subject to: (2) flows must be balanced at each node, (3) - (4) trailer demands must be satisfied, but during $t \leq T_K$, unmet demands are allowed, (5) flows must build enough inventory into each IBC set, (6) - (7) flows and unmet demands must be integer.

The simulation proceeds as follows: each period, a repositioning plan is developed using a planning horizon of $\tau+1$ periods. The first period worth of data corresponds to known demands, while the rest of the data correspond to demand forecasts. The execution cost and unmet demands of the first period are recorded and the decisions for the first day are fixed. The next period, known demand values of that period are revealed, and a new repositioning plan is generated, but the decisions made in previous periods are not changed.

4.3.3 Empty Hub Selection

An important step to deploy either of the two-stage approximations described above requires the identification of the terminals which will serve as the empty-trailer hubs. Since the application context does not provide natural geographical divisions to define the sharing groups (such as in [21]), an optimization model will be defined that will attempt to capture all the important tradeoffs relevant in the selection of e-hubs.

Parameters

- $\mathcal{D}$ = Set of terminals.
- b_i = magnitude of terminal i's imbalance ($b_i \geq 0$). The "imbalance" is a metric of the level of activity that is expected at terminal i regarding empty-trailer repositioning. The metric used in our models is the sum of the absolute value of

each of the net supplies corresponding to a given terminal in the entire planning horizon [$SumAbs$]. Another possible metric is the absolute value of the sum of the net supplies of all the time-expanded nodes corresponding to a terminal [$AbsSum$].

- d_{ij} = distance (or driving time) between terminals i and j.

- $w_{ij} = b_i \cdot d_{ij}$ = weight of assigning terminal i to an e-hub at terminal j.

- k = number of e-hubs to be selected.

- d_{max} = maximum distance (or driving time) between a terminal and its assigned e-hub.

- $d^H_{min} = \alpha(p) \cdot d_{max}$ = minimum distance (or driving time) between two terminals chosen as e-hubs. We use this parameter to enforce that the selected e-hubs are spread throughout the service region and do not cluster around areas with a high volume of activity [1]. $\alpha(p) \in [0, 2]$, and p (which becomes the real parameter to define) is the maximum allowed percentage of overlap between the service regions of any pair of e-hubs; for instance $\alpha(0) = 2$ corresponds to no overlap ($p = 0\%$) between the service regions of any two selected e-hubs, whereas $\alpha(100) = 0$ corresponds to a potential maximum overlap of 100% between the service regions of any pair of e-hubs.

Decision Variables

- $Y_j = 1$ if terminal j is selected as an empty hub. 0 otherwise.

- $X_{ij} = 1$ if terminal i is assigned to an empty hub at terminal j. 0 otherwise.

- $I_{ij} = 1$ if terminals i and j are both selected as empty hubs. 0 otherwise.

[1] The service region of e-hub j is the circle with center at j and radius d_{max}. For any two selected e-hubs, the percentage of overlapped service region to individual service region is equal to the area of the intersection of the two circles over the area of either circle times 100

Formulation

$$
\begin{array}{llll}
Min & \sum_{i\in\mathcal{D}} w_{ij}X_{ij} & & (1)\\
\text{s.t.} & \sum_{j\in\mathcal{D}} X_{ij} = 1 & \forall i \in \mathcal{D} & (2)\\
 & \sum_{j\in\mathcal{D}} Y_j = k & & (3)\\
 & X_{ij} \leq Y_j & \forall i \in \mathcal{D}, \forall j \in \mathcal{D} & (4)]\\
 & d_{ij}X_{ij} \leq d_{max} & \forall i \in \mathcal{D}, \forall j \in \mathcal{D} & (5)\\
 & d_{ij}I_{ij} \geq d^H_{min} & \forall i \in \mathcal{D}, \forall j \in \mathcal{D} : i \neq j & (6)\\
 & I_{ij} \leq Y_i & \forall i \in \mathcal{D}, \forall j \in \mathcal{D} : i \neq j & (7)\\
 & I_{ij} \leq Y_j & \forall i \in \mathcal{D}, \forall j \in \mathcal{D} : i \neq j & (8)\\
 & I_{ij} \geq Y_i + Y_j - 1 & \forall i \in \mathcal{D}, \forall j \in \mathcal{D} : i \neq j & (9)\\
 & Y_j \in \{0,1\} & \forall j \in \mathcal{D} & (10)\\
 & X_{ij} \in \{0,1\} & \forall i \in \mathcal{D}, \forall j \in \mathcal{D} & (11)\\
 & I_{ij} \in \{0,1\} & \forall i \in \mathcal{D}, \forall j \in \mathcal{D} : i \neq j & (12)
\end{array}
$$

This formulation (1) minimizes the total weight of the assignment of terminals to empty hubs, subject to: (2) Each terminal should be assigned to a unique empty hub. (3) k terminals must be selected as empty hubs. (4) A terminal can be assigned to an empty hub only if it is selected. (5) Terminals can only be assigned to empty hubs within a maximum distance. (6) Two selected empty hubs must be separated by a minimum distance. (7)-(9) Definition of $I_{ij}(I_{ij} = Y_iY_j)$. (10)-(12) Binary restrictions. Note that constraints (7)-(10) guarantee that $I_{ij} \in \{0,1\}$, therefore, I_{ij}'s can be defined as a continuous variables.

4.4 Computational Results

In order to implement the proposed two-stage robust-optimization approximations for empty repositioning and evaluate the plans generated, we performed computational experiments using data from a national package/parcel express carrier. For repositioning data, we received information describing all possible connections between the different terminals and their corresponding travel times, as well as rail schedules for moves involving railheads. For loaded data, we received information spanning six weeks and involving loaded trailer dispatches among 264 terminals. We then used an extrapolation procedure to generate loaded trailer demands for different scenarios and generated 14 weeks of loaded data which will be used in our simulation. This procedure takes a stream of nominal loaded trailer dispatches (call it a set *Requests*) spanning 14 weeks of operations and perturbs each of their trailer requirements (*i.e.,* the number of trailers $D(r)$ that are moved from the origin terminal to the destination terminal of request r) to achieve an overall perturbation $\alpha = \frac{\sum_{r \in Requests} \tilde{D}(r) - D(r)}{\sum_{r \in Requests} D(r)}$. Each perturbed demand $\tilde{D}(r)$ can take values in the interval $[0, 2D(r)]$. In our computational experiments, we will use a value of α that, according to the carrier, corresponds to the level of variability observed on its national operations. This will constitute Scenario 1. We will also experiment with data generated using an overall perturbation equal to $\alpha/2$. This will constitute scenario 2. In each scenario, the nominal values will be used as forecasts and the perturbed values will be used as real demand realizations.

4.4.1 Creation of the time-expanded network

The time-expanded network used in our computational experiments has an irregular time discretization and is composed of two different pieces, the road network, which includes inventory and repositioning arcs between carrier-operated terminals, and the rail network, which includes repositioning options using rail moves between the terminals and railheads.

The road network

- **Nodes**: In the given data, each terminal contains between 1 and 4 different times during the day when loaded moves start (*i.e.*, when trailers become unavailable to be loaded and transport goods to a different terminal). Under the assumption that repositioning decisions can only be made at these points in time, we created a time-space node for each terminal at each of these times for each of the days included in the planning horizon. A sink node was created as well whose main purpose is to drain the resources at the end of the planning horizon with a large enough negative net supply. In the remainder of our discussion, we will ignore the specifics about the sink node.

- **Arcs**: The time-expanded network has three types of arcs: inventory, repositioning, and demand arcs. Inventory arcs join consecutive nodes associated with the same terminal and represent the possibility of leaving trailers idle at a given terminal. Repositioning arcs represent the option of moving empty trailers among terminals. We create a repositioning arc joining the time-space node of the terminal of origin at the time of origin with the closest time-space node of the terminal of destination whose time is greater than or equal to the time of origin plus the travel time between the two terminals. Finally, demand arcs correspond to loaded trailer dispatches. We create a demand arc joining the time-space node of the terminal of origin at the time of origin with the closest time-space node of the terminal of destination whose time is greater than or equal to the time of destination. Additionally, demand arcs have associated trailer demands, the number of trailers that need to be moved between the pair of terminals. In each model, the trailer demands for the first day correspond to realized trailer requirements for the given scenario, while trailer demands for the rest of the days correspond to forecasted trailer requirements.

- **Node net supplies**: Since the model used in the simulation explicitly represents loaded demands using arcs, supplies at the nodes will correspond only to the number of trailers available at the given node based on decisions made prior to the horizon start of the model (*i.e.,* the initial state of the system).

- **Interval forecasts for net empty-trailer requirements**: The net trailer requirements at each of the terminal time-space nodes corresponds to the empty trailers that are available or required there before any repositioning takes place. Even though in the model used in the simulation, specific loaded trailers arrive at/depart from each node and there is no need to compute net empty-trailer requirements, the flow-bundle constraints that enforce robustness use information from interval forecasts of these net empty-trailer supplies. Point forecasts $b(n)$ for net empty-trailers are given by the difference between the total loaded trailers into the node minus the total loaded trailers out of the node, and interval forecasts are computed as $[b(n) - \alpha b(n); b(n) + \alpha b(n)]$, where α corresponds to the overall perturbation of the corresponding scenario. Notice that since the interval forecasts are a function of nominal supplies, they remain static throughout the entire simulation, that is, they do not get refined as execution time gets closer.

The rail network

Trailers can also be repositioned among terminals using rail moves; however, unlike road moves, rail legs are only available at specific times during a given week. We model the rail network as follows: for each rail leg, we create nodes at the start time at the origin railhead and at the end time at the destination railhead, and we create the following arcs: one arc between the origin and destination railheads at the specific timing, one arc for each feasible terminal-origin railhead, and destination railhead-terminal connections. We do not allow inventory arcs at the railheads (to avoid repositioning plans that would send trailers from a terminal to a single railhead and then to another terminal), but we have added repositioning arcs corresponding to feasible indirect connections between railheads. In particular, we added arcs representing all possible arc moves that would take up to three days of travel time.

4.4.2 Simulation Results

We simulated the generation and execution of our repositioning plans over 14 weeks of operations for two different scenarios. Scenario 1 is the base scenario that exhibits an amount of variability similar to that exhibited by the carrier on its national operations. Scenario 2 exhibits half the amount of the variability of Scenario 1.

Each day, a repositioning plan over the following two weeks is developed. Input data include the relevant portion of the time-expanded network for the current model, the state of the system (number of trailers at each of the terminals and in transit) based on repositioning decisions made in prior days, and the relevant demand data. Data for day one correspond to realized trailer requirements for the given scenario and data for days 2 through 14 correspond to forecasted trailer requirements. The execution costs and unmet demands during that day are recorded and the decisions for the first day are fixed. The next day, the horizon is rolled, known trailer requirements within the new day are revealed, and a new model is solved. The process is repeated over 14 weeks. To avoid cool down effects, the execution costs and unmet demands are recorded for only the first 12 weeks worth of data for which models involving entire two-week horizons can be solved.

In what follows, we will use the following notation to refer to the model results: $LX - TRY - RH$ represents the model that imposes a limit of X non-hubs that can participate in an IBC set, with a robustness horizon $T_R = Y$. Parameter RH is optional, and distinguishes the variant of the models that explicitly consider the rolling horizon implementation. Using this notation, model $L0 - TR0$ corresponds to the deterministic MCNF model that adds no robust constraints.

Figures 14 and 15 show the results of the models for Scenario 1 in terms of unmet demands. The deterministic model provides a benchmark on the number of unmet

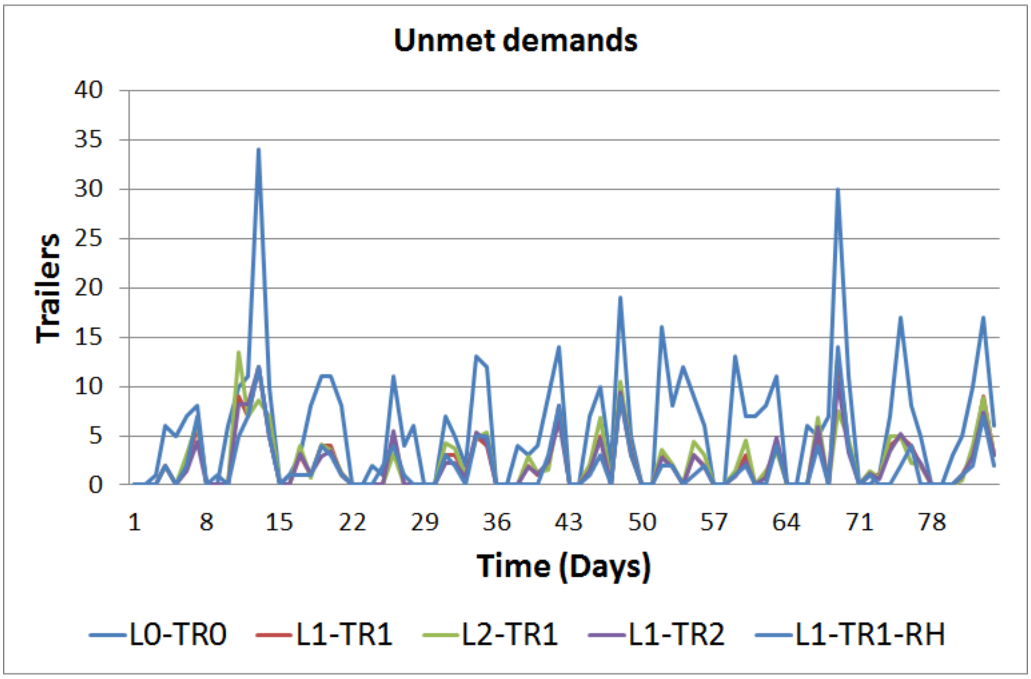


Figure 14: Unmet demands on a given day - Scenario 1

demands that result from ignoring the uncertainties in the forecasts for future trailer requirements. For this scenario, model $L1 - TR1 - RH$ provides the best results with a reduction of 70% of the cumulative unmet demands over 12 weeks with respect to those of the deterministic plans. Additionally, the models that ignore the rolling horizon implementation do not produce significantly different results when the robustness horizon is increased from one day to two days or when the number of allowed non-hubs in an IBC set is increased from one to two. On average, these models yield reductions of 64% of the cumulative demands.

Figures 16 and 17 show the results of the models for Scenario 1 in terms of plan execution costs. The deterministic model also provides a benchmark on the costs of the repositioning plans that result from ignoring the uncertainties in the forecasts for future trailer requirements. For this scenario, model $L1 - TR1 - RH$ also provides the best results with an increase of only 6% of the cumulative execution costs over 12 weeks over those of the deterministic plans. Once again, the models that ignore the rolling horizon implementation do not produce significantly different results when

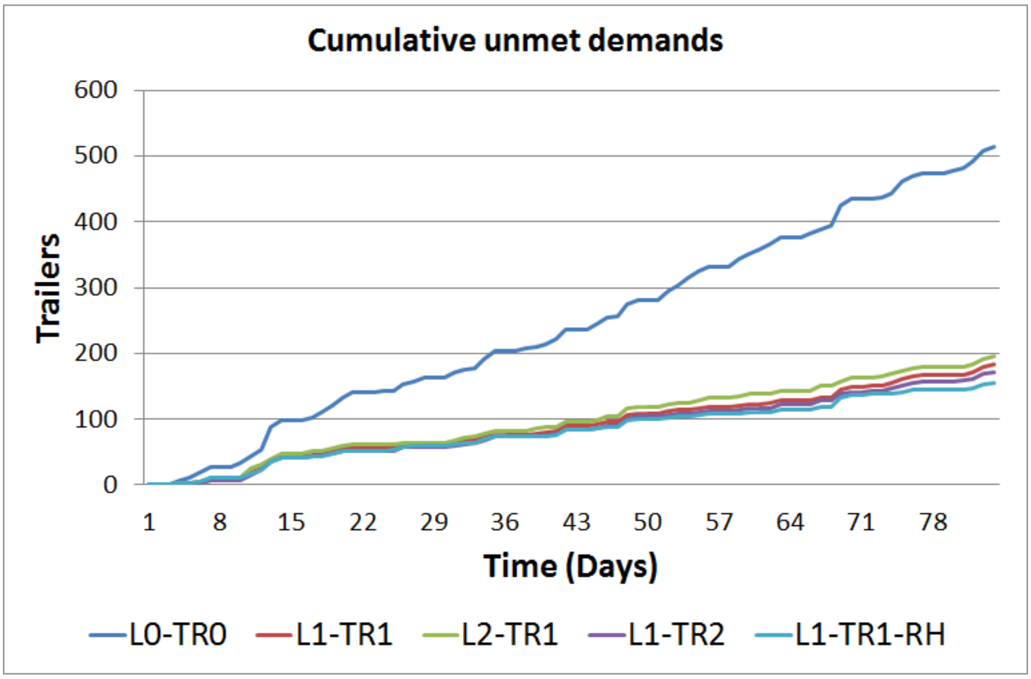


Figure 15: Cumulative unmet demands - Scenario 1

the robustness horizon is increased from one day to two days or when the number of allowed non-hubs in an IBC set is increased from one to two. On average, these models involved increments of 13% of the cumulative execution costs.

Figures 18 and 19 show the results of the models for Scenario 2 in terms of unmet demands. As expected, unmet demands are fewer in this scenario than in Scenario 1. For this scenario, model $L1 - TR1 - RH$ provides the best results with a reduction of 80% of the cumulative unmet demands over 12 weeks with respect to those of the deterministic plans. Once again, the models that ignore the rolling horizon implementation do not produce significantly different results when the robustness horizon is increased from one day to two days or when the number of allowed non-hubs in an IBC set is increased from one to two. On average, these models yield reductions of 73% of the cumulative demands.

Figures 16 and 17 show the results of the models for Scenario 2 in terms of plan execution costs. For this scenario, model $L1 - TR1 - RH$ also provides the best results with an increase of only 8% of the cumulative execution costs over 12 weeks

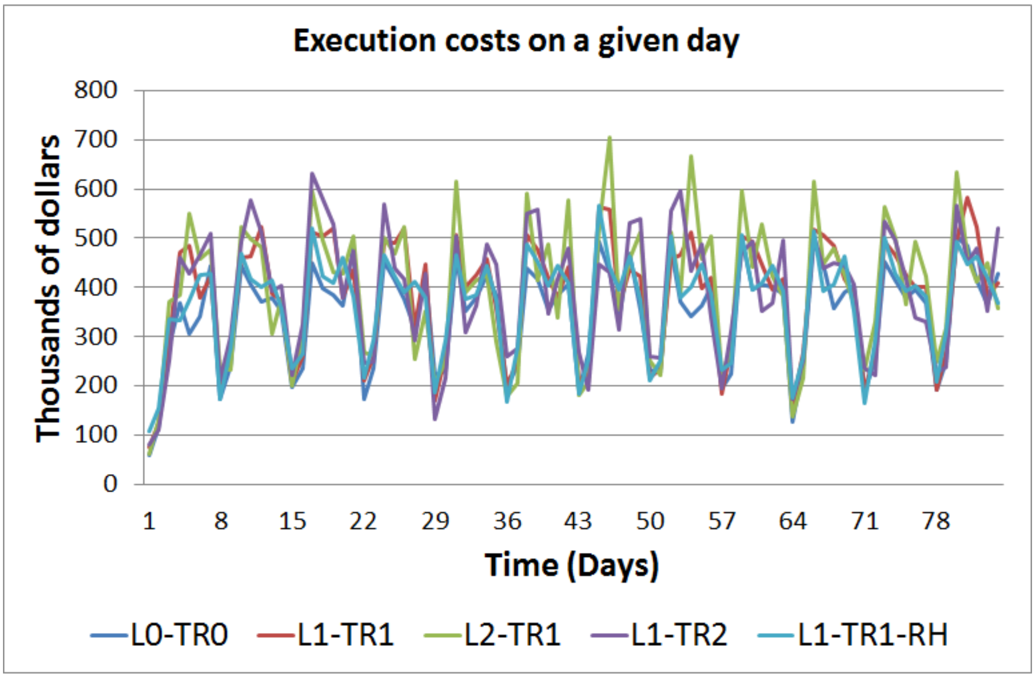


Figure 16: Execution costs on a given day - Scenario 1

over those of the deterministic plans. On average, the models that ignore the rolling horizon implementation involved increments of 13% of the cumulative execution costs.

In addition to studying the performance of the plans generated by our proposed approaches against that of plans generated with deterministic models, we were also interested in evaluating the effects that the choice of the planning horizon length has over the performance of the plans. To this effect, we conducted computational experiments in which we varied the length of the planning horizon used. Figures 22 and 23 illustrate two important points:

a) The robust optimization models can use shorter planning horizons to obtain better quality decisions than those obtained with the pure deterministic model. In particular, all the robust optimization models with a planning horizon of at least three days provide better results in terms of unmet demands than the deterministic model can attain with the same or longer planning horizons. In terms of execution costs though, robust models require planning horizons of

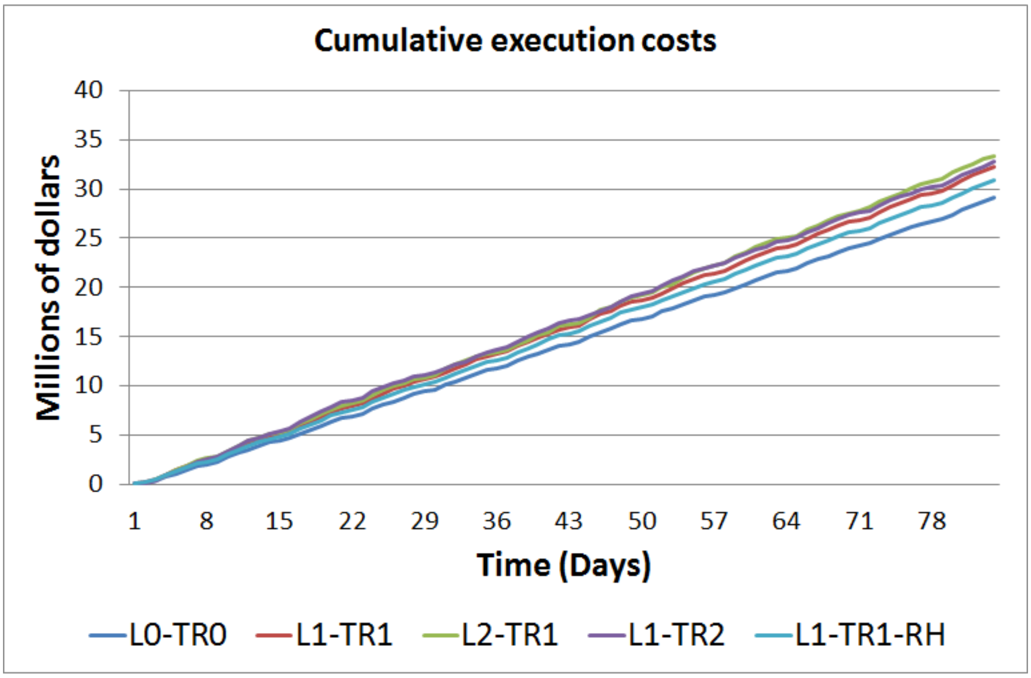


Figure 17: Cumulative execution costs - Scenario 1

around six days to achieve their best results.

b) The robust optimization models that explicitly consider the rolling horizon implementation provide the best results and can use less conservative uncertainty estimates than robust optimization models which ignore this key implementation idea. In particular, the robust models with rolling horizon implementations that use $T_R = 1$ and $L = 1$ perform better than robust optimization models with $T_R = 2$ or $L = 2$.

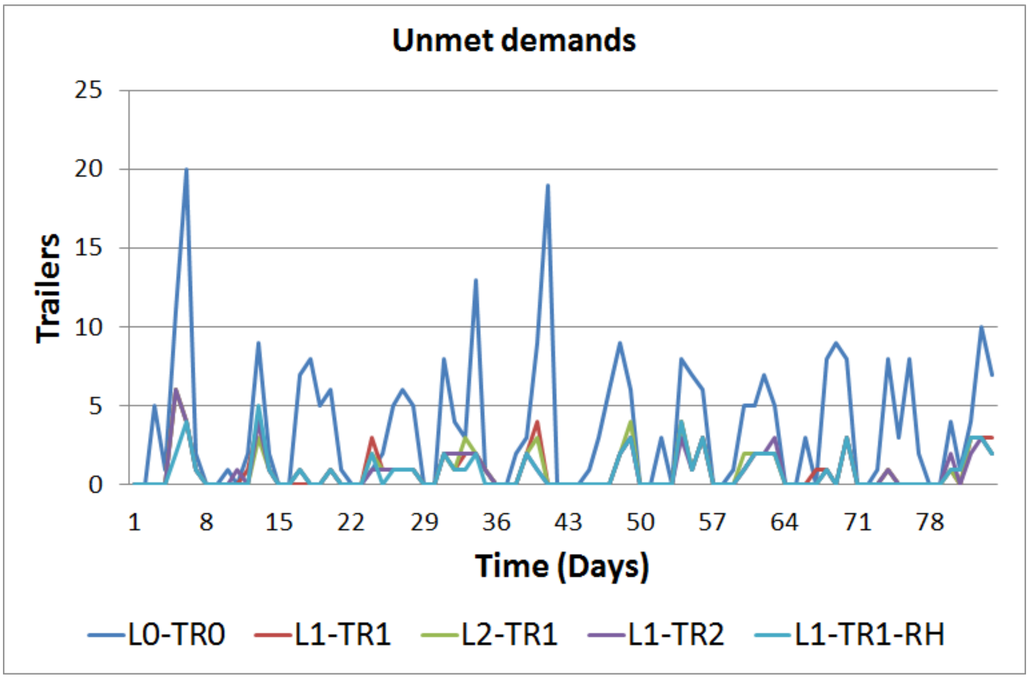


Figure 18: Unmet demands on a given day - Scenario 2

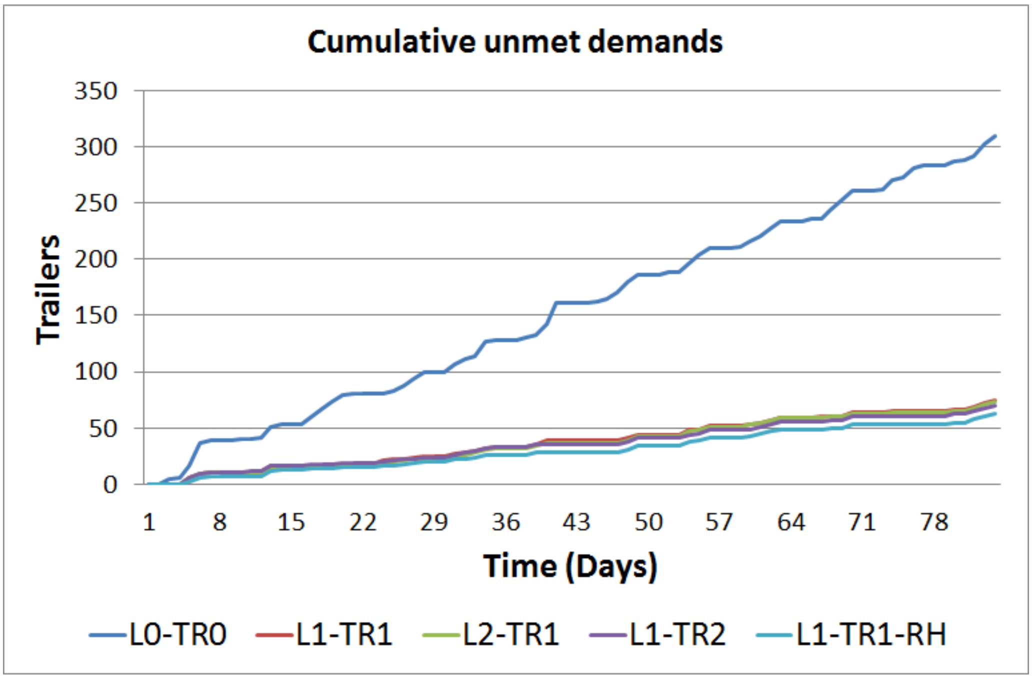


Figure 19: Cumulative unmet demands - Scenario 2

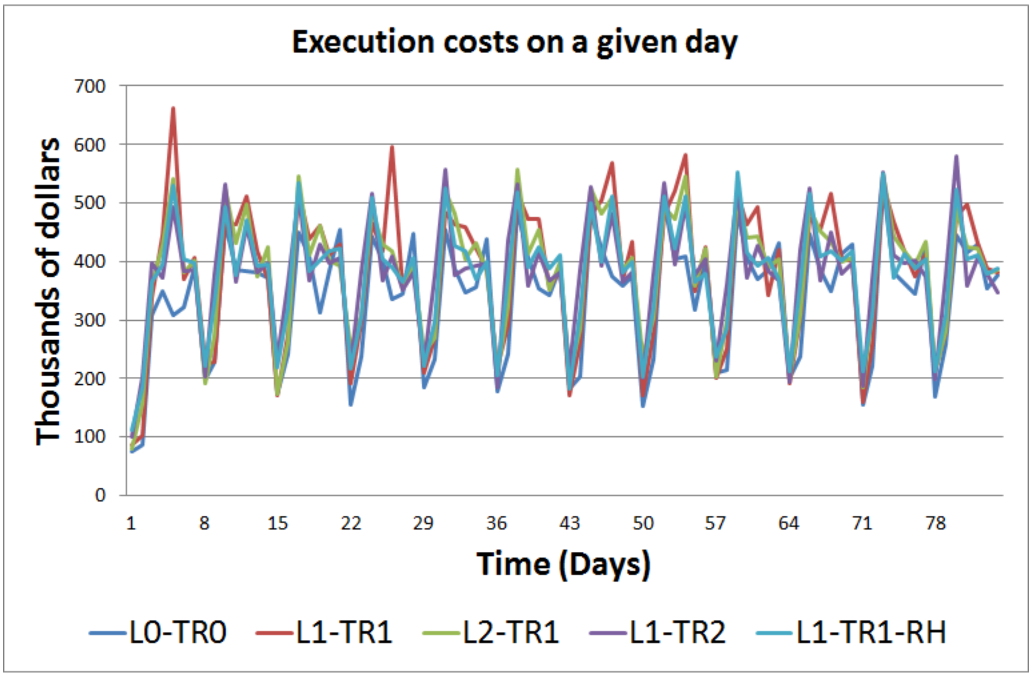


Figure 20: Execution costs on a given day - Scenario 2

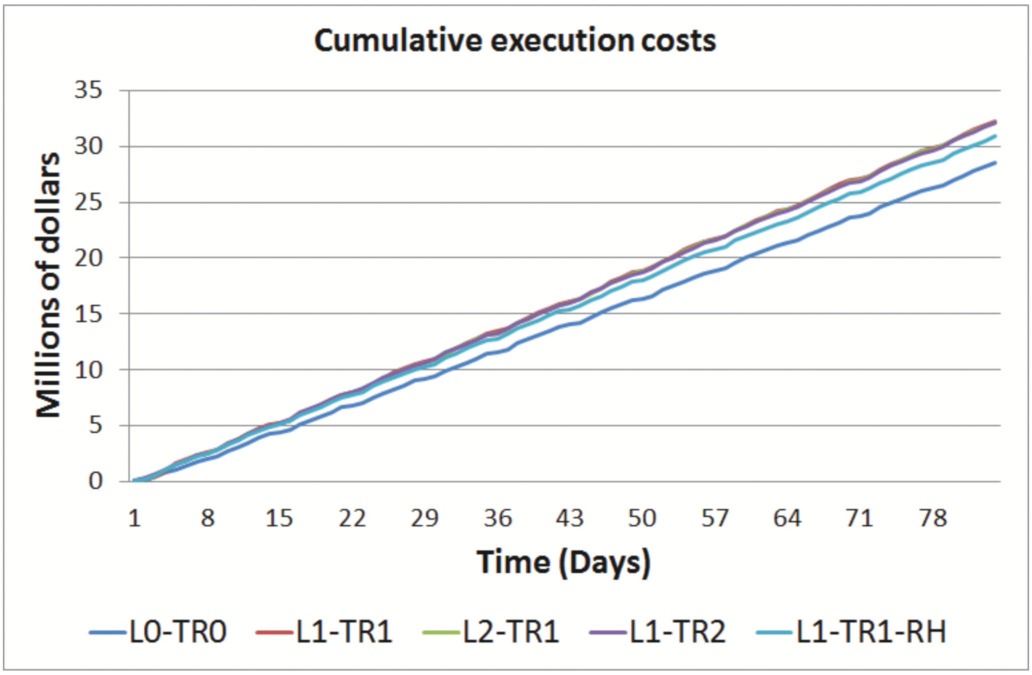


Figure 21: Cumulative execution costs - Scenario 2

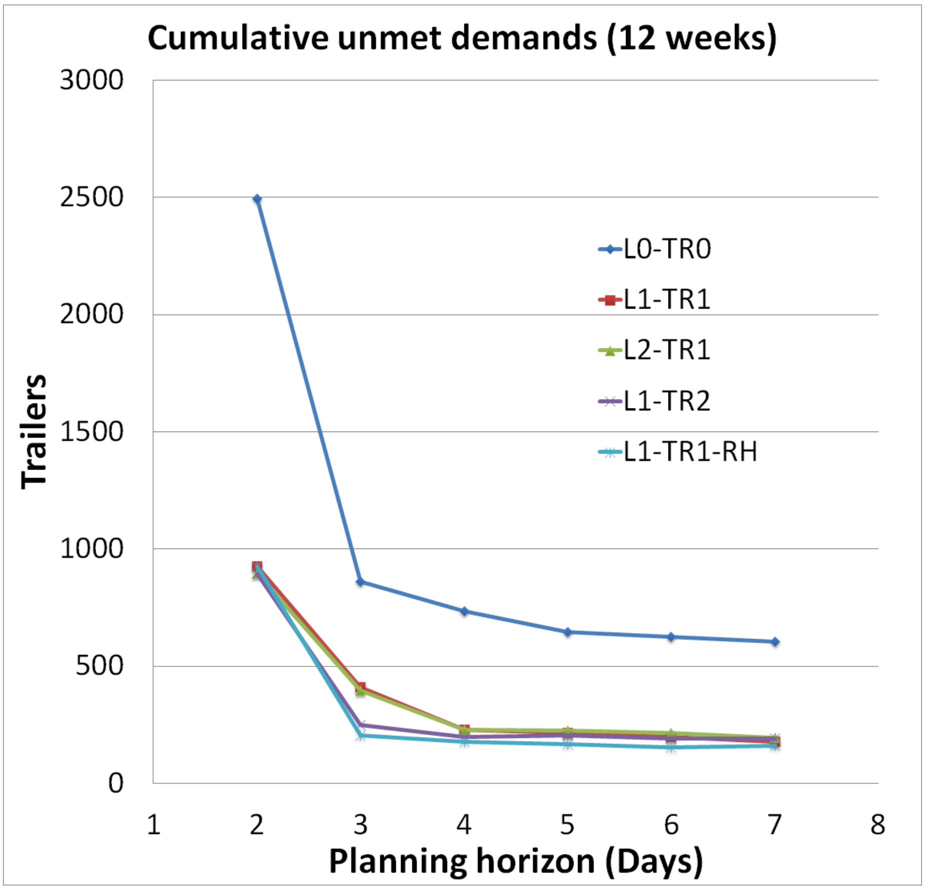


Figure 22: Cumulative unmet demands with different planning horizons - Scenario 1

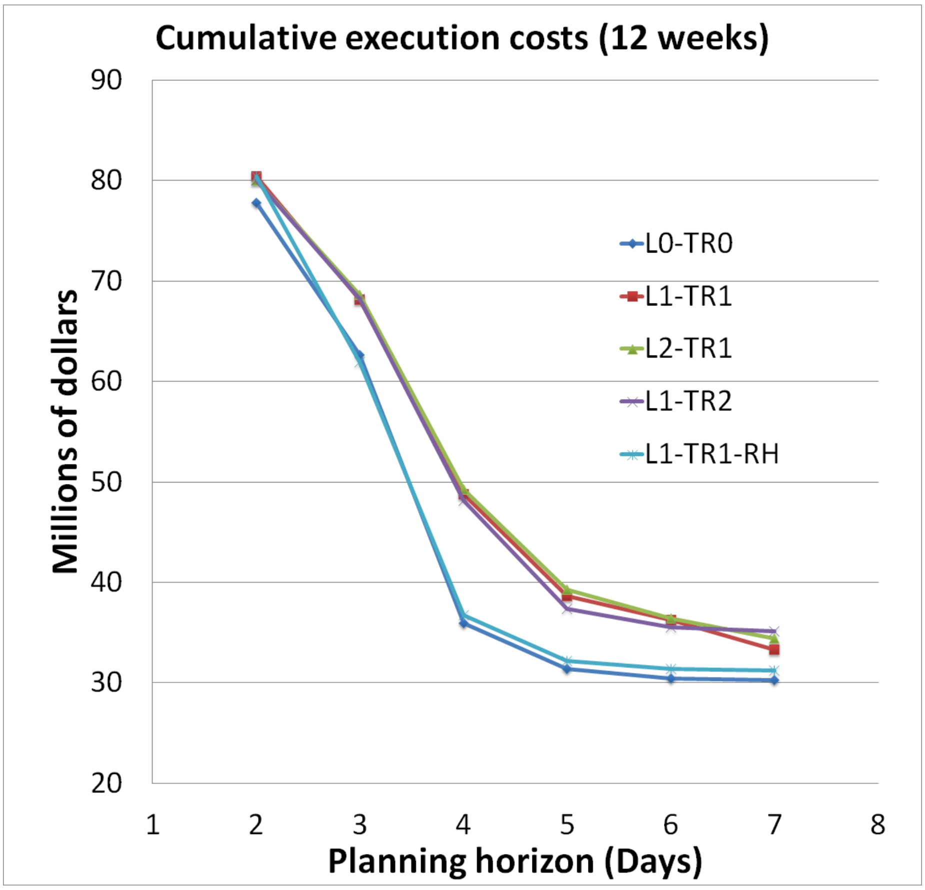


Figure 23: Cumulative execution costs with different planning horizons - Scenario 1

CHAPTER V

CONCLUSIONS AND FUTURE WORK

This dissertation proposed approaches that enable effective planning and control of mobile transportation resources in large-scale freight consolidation networks. We develop models, algorithms, and methodologies that are applied to fleet sizing and fleet repositioning.

Chapter 2 introduced a modeling framework for exploring the value of tractor repositioning strategies to reduce the costs of operating and maintaining a tractor fleet during a given planning horizon. Four tractor repositioning strategies were evaluated and their benefits were quantified using two different costing schemes. Results from a computational study, using real data from a national LTL carrier, show that cost savings of up to 5% can be achieved by performing extra tractor repositioning moves to reduce the fleet size. A number of issues remain and constitute venues for future research. We investigated repositioning as a mechanism to reduce the fleet size. This was done separate from the repositioning that takes place to balance resources throughout the network. Planning repositioning moves to balance resources and reduce fleet size simultaneously may yield further cost savings. In addition, a similar modeling framework can be used to evaluate the benefits of the "dual-use" of linehaul tractors. Dual-use of linehaul tractors refers to their use in pickup and delivery operations at terminals. Using linehaul tractors in pickup and delivery operations may lead to reductions in the size of the tractor fleets maintained at terminals for the pickup and delivery operations (the tractors used for pickup and delivery operations are smaller and cannot be used for linehaul operations). In fact, integrated models that consider repositioning to reduce linehaul tractor fleet size and pickup and delivery tractor fleet size are also a possible option. Furthermore, the models presented in Chapter 2 used historical trailer dispatch data and thus implicitly assume that historical dispatch data is representative of future dispatches. Modeling uncertainty in the scheduled trailer dispatch moves (for example, in terms of dispatch times or the number of tractors required to serve the loaded requests) is a natural and useful extension.

Chapter 3 elaborated on the work to understand the trade-offs between fleet size repositioning costs. It described procedures that compute the optimal Pareto frontier between fleet size and repositioning costs required to perform a fixed aperiodic or periodic schedule of transportation requests. Two different modeling frameworks (one using time-expanded networks and the other bipartite networks) were introduced and contrasted in terms of practical applicability. For aperiodic schedules, it was shown that all of the Pareto points can be computed in polynomial time via linear programming formulations of flows on time-expanded networks or minimum weight perfect matchings on bipartite networks. Furthermore, it was proved that adjacent Pareto points can be computed efficiently by solving a single shortest path problem in either type of network. Aperiodic schedules were found to be more difficult and it was shown that the natural extensions from aperiodic networks fail to provide efficient algorithms to compute adjacent Pareto points. Only the end points on the frontier were shown to be computed in polynomial time by solving a sequence of two linear problems and the rest of the points in the frontier can be computed using either integer programming flow formulations or perfect matchings with additional side constraints. The NP-completeness of the bicriteria optimization model for periodic networks remains an open problem. Future research should incorporate repeatability and regularity conditions into the analysis; these conditions are desirable in practice as they produce repositioning plans that are easier to implement and monitor.

Chapter 4 considered robust models for dynamic empty-trailer repositioning problems in very large-scale consolidation networks. We investigated approaches that deploy two-stage robust optimization models in a rolling horizon framework to address a multistage dynamic empty repositioning problem in which information is revealed over time. Using real data from a national package/parcel express carrier, we conducted a simulation to evaluate the performance of repositioning plans in terms of unmet loaded requests and execution costs, and showed that the plans generated with our proposed approaches can reduce the unmet loaded requests up to 80% with a modest increase of 8% in execution costs over those plans generated by deterministic optimization models. Additionally, we provided computational evidence supporting that (1) robust optimization models can use shorter planning horizons to obtain the same or better quality decisions than those obtained with pure deterministic models, and (2) robust optimization models designed explicitly for embedding within rolling horizon implementations can use less conservative uncertainty estimates than robust optimization models which ignore this key implementation idea. A number of questions remain open and constitute venues for future research, including: (1) How to appropriately select the terminals that will serve as empty-hubs? In this study we proposed an optimzation model that aimed to capture the important trade-offs in the empty-hub selection, but further research is required to shed more light into this important decision. (2) What role, if any, do trailer mix decisions have into the construction and implementation of robust empty-trailer repositioning plans? (3) What is the impact of correlations between net trailer imbalances? Uncertainty in net trailer imbalances is driven by uncertainty in loaded demands, which are in turn driven by changes in freight. Understanding the impact of correlations in those changes can lead to better definitions of the uncertainty sets against which protection is sought.

REFERENCES

[1] Abrache, J., C. T. and Gendreau, M., "A new decomposition algorithm for the deterministic dynamic allocation of empty containers," *Technical Report CRT-99-49, Centre de Recherche sur les Transport*, 1999, Montreal, Quebec, Canada.

[2] Ahuja, R., Magnati, T., and Orlin, J., *Network Flows: Theory, Algorithms and Applications.* Upper Saddle River, New Jersey: Prentice-Hall, Inc., 1993.

[3] A.T.A., *American Trucking Associations. Trucking and the Economy. 2007-2008 edition.* 2008.

[4] Atamturk, A. and Zhang, M., "Two stage robust network flow and design under demand uncertainty," *Operations Research*, vol. 55, pp. 662–673, 2007.

[5] Bartlett, T., "An algorithm for the minimum number of transport units to maintain a fixed schedule," *Naval Res. Logist. Quart.*, vol. 4, pp. 139–149, 1957.

[6] Bartlett, T. and Charnes, A., "Cyclic scheduling and combinatorial topology: Assignment and routing of motive power to meet scheduling and maintenance requirements; ii. generalization and analysis," *Naval Res. Logist. Quart.*, vol. 4, pp. 207–220, 1957.

[7] Beaujon, G. and Turnquist, M., "A model for fleet sizing and vehicle allocation," *Transportation Science*, vol. 25, no. 1, pp. 19–45, 1991.

[8] Ceder, A., "Estimation of fleet size for variable bus schedules," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1903, pp. 2–10, 2005.

[9] Ceder, A. and Stern, H., "Deficit function bus scheduling with deadheading trip insertions for fleet size reduction," *Transportation Science*, vol. 15, no. 4, pp. 338–363, 1981.

[10] Cheung, R. and Powell, W., "An algorithm for multistage dynamic networks with random arc capacities, with an application to dynamic fleet management," *Operations Research*, vol. 44, pp. 951–963, 1996.

[11] Crainic, T., Gendreau, M., and Dejax, P., "Dynamic stochastic models for the allocation of empty containers," *Operations Research*, vol. 41, pp. 102–126, 1993.

[12] Current, J. and Marsh, M., "Multiobjective transportation network design and routing problems: Taxonomy and annotation," *European Journal of Operational Research*, vol. 65, pp. 4–19, 1993.

[13] Current, J. and Min, H., "Multiobjective design of transportation networks: Taxonomy and annotation," *European Journal of Operational Research*, vol. 26, pp. 187–201, 1986.

[14] Dantzig, G. and Fulkerson, D., "Minimizing the number of tankers to meet a fixed schedule," *Naval Res. Logist. Quart.*, vol. 25, no. 1, pp. 217–222, 1954.

[15] Dejax, P. and Crainic, T., "A review of empty flows and fleet management models in freight transportation," *Transportation Science*, vol. 21, no. 4, pp. 227–248, 1987.

[16] Di Francesco, M., Crainic, T., and Zuddasa, P., "The effect of multiscenario policies on empty container repositioning," *Transportation Research Part E: Logistics and Transportation Review*, vol. 45, pp. 758–770, 2009.

[17] Du, Y. and Hall, R., "Fleet sizing and empty equipment redistribution for center-terminal transportation networks," *Management Science*, vol. 43, no. 2, pp. 145–157, 1997.

[18] Ehrgott, M., *Multicriteria optimization.* Lecture Notes in Economics and Mathematical Systems, Berlin: Springer-Verlag, 2000.

[19] Ehrgott, M. and Gandibleux, X., "A survey and annotated bibliography of multiobjective combinatorial optimization," *OR Spektrum*, vol. 22, pp. 425–460, 2000.

[20] Erera, A., Morales, J., and Savelsbergh, M., "Global intermodal container management for the chemical industry," *Transportation Research, Part E*, vol. 41, no. 6, pp. 551–566, 2005.

[21] Erera, A., Morales, J., and Savelsbergh, M., "Robust optimization for empty repositioning problems," *Operations Research*, vol. 57, no. 2, pp. 468–483, 2009.

[22] Frantzeskakis, L. and Powell, W., "A successive linear approximation procedure for stochastic, dynamic vehivle allocation problems," *Transportation Science*, vol. 24, pp. 40–57, 1990.

[23] Furth, P. G., "Alternating deadheading in bus route operations," *Transportation Science*, vol. 19, no. 1, pp. 13–28, 1985.

[24] Gavish, B. and Schweitzer, P., "An algorithm for combining truck trips," *Transportation Science*, vol. 8, pp. 13–23, 1974.

[25] Gavish, B., Schweitzer, P., and Shlifer, E., "Assigning buses to schedules in a metropolitan area," *Comput. & Ops Res.*, vol. 5, pp. 129–138, 1978.

[26] Gertsbach, I. and Gurevich, Y., "Constructing an optimal fleet for a transportation schedule," *Transportation Science*, vol. 11, no. 1, pp. 20–35, 1974.

[27] Gertsbakh, I. and Stern, H., "Minimal resources for fixed and variable job schedules," *Operations Research*, vol. 26, no. 1, pp. 68–85, 1978.

[28] Godfrey, G. and Powell, W., "An adaptive dynamic programming algorithm for single period fleet management problems i: Single period travel times," *Transportation Science*, vol. 36, pp. 21–39, 2002.

[29] Godfrey, G. and Powell, W., "An adaptive dynamic programming algorithm for single period fleet management problems ii: Multiperiod travel times," *Transportation Science*, vol. 36, pp. 40–54, 2002.

[30] Hamacher, H., Pedersen, C., and Ruzika, S., "Multiple objective minimum cost flow problems: A review," *European Journal of Operational Research*, vol. 176, pp. 1404–1422, 2007.

[31] Leddon, C. and Wrathall, E., "Scheduling empty freight car fleets on the louisville and nashville railroad," *Second International Symposium. Use of cybernetics on the Railways*, pp. 102–126, October 1-6. Montreal, Quebec, Canada.

[32] Lee, H. and Pulat, S., "Bicriteria network flow problems: Continuous case," *European Journal of Operational Research*, vol. 51, pp. 119–126, 1991.

[33] Lee, H. and Pulat, S., "Bicriteria network flow problems: Integer case," *European Journal of Operational Research*, vol. 66, pp. 148–157, 1993.

[34] Misra, S., "Linear programming of empty wagon disposition," *Rail Internat.*, vol. 3, pp. 151–158, 1972.

[35] Morales, J., *Planning Robust Freight Transportation Operations.* PhD thesis, Georgia Institute of Technology, 2006.

[36] Orlin, J., "Minimizing the number of vehicles to meet a fixed periodic schedule: An application of periodic posets," *Operations Research*, vol. 30, no. 4, pp. 760–776, 1982.

[37] Papier, F. and Thonemann, U., "Queueing models for sizing and structuring rental fleets," *Transportation Science*, vol. 42, no. 3, pp. 302–317, 2008.

[38] Parragh, S., Doerner, K., Hartl, F., and Gandibleux, X., "A heuristic two-phase solution approach for the multi-objective dial-a-ride problem," *Networks*, vol. 54, no. 4, p. 227242, 2009.

[39] Powell, W., "A stochastic formulation of the dynamic vehicle allocation problem," *Transportation Science*, vol. 20, pp. 117–129, 1986.

[40] Powell, W., "An operational planning model for the dynamic vehicle allocation problem with uncertain demands," *Transportation Research Part B*, vol. 21B, pp. 217–232, 1987.

[41] Przybylski, A., Gandibleux, X., and Ehrgott, M., "The biobjective integer minimum cost flow problemincorrectness of sedeo-noda and gonzlez-martins algorithm," *Computers Ops Res.*, vol. 33, pp. 1459–1463, 2006.

[42] Przybylski, A., Gandibleux, X., and Ehrgott, M., "Two phase algorithms for the bi-objective assignment problem," *European Journal of Operational Research*, vol. 185, pp. 509–533, 2008.

[43] Pulat, P., Huarng, F., and Lee, H., "Efficient solutions for the bicriteria network flow problem," *Computers Ops Res.*, vol. 19, no. 7, pp. 649–655, 1992.

[44] Raith, A. and Ehrgott, M., "A two-phase algorithm for the biobjective integer minimum cost flow problem," *Computers Ops Res.*, vol. 36, pp. 1945–1954, 2009.

[45] Schrijver, A., *Combinatorial Optimization. Polyhedra and Efficiency.* Germany: Springer, 2003.

[46] Sedeno-Noda, A. and Gonzalez-Martin, C., "An algorithm for the biobjective integer minimum cost flow problem," *Computers Ops Res.*, vol. 28, pp. 139–156, 2001.

[47] Shen, Z. and Daskin, M., "Tradeoffs between customer service and cost in an integrated supply chain design framework," *M&SOM*, vol. 7, pp. 188–207, 2005.

[48] Sherali, H., Bish, E., and Zhu, X., "Airline fleet assignment concepts, models, and algorithms," *European Journal of Operational Research*, vol. 172, no. 1, pp. 1–30, 2006.

[49] Sherali, H. and Tuncbilek, C., "Static and dynamic time-space strategic models and algorithms for multilevel rail-car fleet management," *Management Science*, vol. 43, no. 2, pp. 235–250, 1997.

[50] Stern, H. and Ceder, A., "An improved lower bound to the minimum fleet size problem," *Transportation Science*, vol. 17, no. 4, pp. 471–477, 1983.

[51] Turnquist, M. and Jordan, W., "Fleet sizing under production cycles and uncertain travel times," *Transportation Science*, vol. 20, no. 4, pp. 227–236, 1986.

[52] Ulungu, E. and Teghem, J., "Multi-objective combinatorial optimization problems: A survey," *Journal of Multi-Criteria Decision Analysis*, vol. 3, no. 2, pp. 83–104, 1994.

[53] Vilcot, G. and Billaut, J., "A tabu search and a genetic algorithm for solving a bicriteria general job shop scheduling problem," *European Journal of Operational Research*, vol. 190, pp. 398–411, 2008.

[54] White, W., "Dynamic transhipment networks: an algorithm and its application to the distribution of empty containers," *Networks*, vol. 2, pp. 211–236, 1972.

[55] Wu, P., Hartman, J. C., and Wilson, G. R., "An integrated model and solution approach for fleet sizing with heterogeneous assets," *Transportation Science*, vol. 39, no. 1, pp. 87–103, 2005.

[56] Zak, J., Redmer, A., and Sawicki, P., "Multiple objective optimization of the fleet sizing problem for road freight transportation," *Journal of Advanced Transportation*, vol. 42, no. 4, pp. 379–427, 2008.

[57] Zhang, G., Smilowitz, K., and Erera, A., "Dynamic planning for urban drayage operations," *Transportation Research Part E*, vol. 47, pp. 764–777, 2011.