A Robust Rolling Horizon Framework for Empty Repositioning

Ilke Bakir

Department of Operations, Faculty of Economics and Business, University of Groningen, The Netherlands

Alan Erera, Martin Savelsbergh

H. Milton Stewart School of Industrial & Systems Engineering, Georgia Institute of Technology, Atlanta, U.S.A.

Abstract

Naturally imbalanced freight flows force consolidation carriers to reposition resources empty. When constructing empty repositioning plans, the cost of repositioning resources empty needs to be weighed against the cost of corrective actions in case of unavailable resources. This is especially challenging given the uncertainty of future demand. We design and implement a robust rolling horizon framework for constructing effective empty repositioning plans. An extensive computational study demonstrates the benefits of explicitly accounting for uncertainty in future demand by using robust optimization, and pragmatically controlling the level of conservatism in hedging against this uncertainty. We also investigate practical strategies for reducing the complexity of managing the repositioning of empty resources in transportation service networks covering huge geographic areas.

Keywords: Empty repositioning; Robust optimization; Time-expanded networks; Sharing group policies

1 Introduction

One of the major challenges faced by a transportation service provider is the largely imbalanced nature of freight flows and the resulting need to reposition resources empty. The imbalanced nature of load requests causes containers to accumulate at terminals that have mostly inbound freight, while other terminals, which have mostly outbound freight, are in constant need of containers. When resources are needed at a terminal for loaded outbound moves, it may not be possible to direct a sufficient number of loaded inbound moves to that terminal in time, which may result in potentially costly corrective actions (e.g., rerouting freight on more circuitous paths, which, in turn, may result in missed service commitments, or outsourcing to third parties). To prevent such situations as much as possible, carriers have to move resources empty. Balancing the transportation

cost incurred by moving resources empty and the benefit of having resources at the right location at the right time to accommodate future load requests is critical when making empty repositioning decisions.

The main challenge when developing effective empty repositioning strategies is future load uncertainty. As a result of the uncertainty in future load requests, a carrier may, at times, be unable to satisfy load requests or may, at times, be moving resources empty unnecessarily. Satisfying demand without employing a needlessly large fleet and without excessively repositioning empty resources requires careful planning in environments where demand is uncertain. Incorporating demand uncertainty into the empty repositioning strategies can be done in different ways. In this paper, we focus on empty repositioning strategies based on robust optimization methods.

A secondary challenge arises when a transportation service provider serves a large region and operates a large number of terminals, because, ideally, empty repositioning moves should be carried out between terminals that are in close proximity to each other, which not only controls cost (since long empty repositioning moves are obviously more expensive), but also helps control the complexity of the day-to-day operations. A natural strategy that accomplishes this, and is often seen in practice, is to divide the region served into subregions for empty repositioning purposes and to manage each subregion somewhat independently. These so-called *sharing group policies* partition the terminals in the transportation network into sharing groups, each with a designated empty hub. In a sharing group policy, empty repositioning within a sharing group is between non-hub terminals and the hub, and empty repositioning between sharing groups is between the empty hubs. Such policies naturally introduce pooling benefits.

Early optimization approaches for solving empty repositioning problems are deterministic and model the problem on a time-expanded (or time-space) network with nodes representing terminals at specific time periods and arcs representing loaded and possible empty repositioning moves. Resource requirements at terminals are typically point estimates, because future load requests are uncertain in most settings. An important and recognized shortcoming of these approaches is that they do not capture future demand uncertainty, which may result in suboptimal empty repositioning plans or even infeasibility after the load requests have materialized.

To model the empty repositioning problem with demand uncertainty more accurately, we propose a variant of the robust optimization approach of Erera et al. (2009) and demonstrate its efficacy with a comprehensive computational study. Empty repositioning plans generated by the robust approach guarantee the existence of short empty repositioning moves to recover feasibility after

demand has materialized for every possible demand realization, or for a subset of possible demand realizations, assuming demand realizations come from a known uncertainty set. Since repositioning decisions are made prior to the materialization of demand, the proposed robust optimization model is similar to a two-stage stochastic (integer) programming model, in which first and second stage decisions represent the repositioning plan and the recourse actions given a demand realization, respectively.

In our robust approach, instead of explicitly solving a second stage problem, the existence of recovery actions is ensured by an additional set of constraints in the first stage problem. Furthermore, the level of uncertainty the recovery actions are to hedge against is controlled by limiting deviations from nominal values in a way that is similar to uncertainty budgets, an idea introduced in Bertsimas and Sim (2003). The core of the optimization problem that has to be solved is a minimum cost flow problem with flow balance and flow bound constraints to ensure that nominal demand is met and empty resources are repositioned at minimum cost. The problem is augmented with constraints that ensure recoverability, *i.e.*, that additional empty resources can be repositioned, if necessary, to accommodate deviations from nominal demand. Since only low-cost repositioning moves are considered for recovery actions, the costs associated with potential recovery actions are ignored.

In reality, the empty repositioning problem has a multi-stage nature, because demand information becomes available over time. To accurately represent this multi-stage nature, the proposed robust empty repositioning model is embedded in a rolling horizon framework. Consequently, each time the model is solved, it is assumed that only the current day's load requests are known with certainty and that the only information available about future load requests are nominal demand quantities and uncertainty intervals. Furthermore, only the current day's decisions are implemented before the horizon is rolled forward.

By taking advantage of the relatively minor data requirements of robust optimization models compared to other optimization methods that address uncertainty, the proposed robust approach provides high-quality solutions while being simple in modeling and implementation, and computationally tractable even for large-scale systems. Computational experiments demonstrate that this approach generates empty repositioning plans that are much less sensitive to deviations from expected demand with only a slight increase in transportation cost compared to plans generated with a deterministic approach using only nominal demand quantities. For example, in a 100-terminal network, averaged over 30 instances, the robust approach increases the service level (*i.e.*, the fraction of load requests that can be satisfied without any corrective actions) by 2.6% when compared to a non-robust approach, while incurring only a 1.6% increase in transportation cost. By adjusting

the conservatism of the robust approach, for the same set of instances, it is possible to achieve a 2.0% increase in service rate, while only incurring a 1.0% increase in transportation cost.

As briefly mentioned before, sharing group policies for empty repositioning are often used in practice when operating large-scale transportation service networks. In addition to being sensible from an operational perspective, such policies are also useful from a computational viewpoint. Allowing empty repositioning between too many pairs of terminals will result in empty repositioning plans that tend to be more sensitive to demand uncertainty (due to a lack of risk pooling), and will increase the computational burden of generating the plans (due to the large, sometimes excessive, number of empty repositioning options). On the other hand, employing a sharing group policy with too few sharing groups or sharing groups with high (internal) flow imbalance can also result in empty repositioning plans with high costs (either because of long distances between empty hubs or because of frequent repositioning between empty hubs). Therefore, it is important for freight transportation service providers to carefully design the sharing group configuration if they plan to employ a sharing group based empty repositioning strategy. As part of our computational study, we conduct experiments to assess the effects of the sharing group configuration on empty repositioning costs (using both the nominal model and the robust model). We find that it is beneficial to define the sharing groups in such a way that they have roughly equal inbound and outbound loaded flows (if possible) and to carefully trade-off the distance between non-hub terminals and the hub in a sharing group and the distance between hubs of different sharing groups.

To summarize, the contributions of this paper are as follows:

(i) The robust optimization model and recoverability conditions of Erera et al. (2009) are extended to handle demand uncertainty related to forecast load requests (rather than demand uncertainty related to forecast in- and outflows at terminals).

(ii) New mechanisms to control the level of conservatism and the computational requirements of the solution of the robust optimization model are introduced.

(iii) A rolling horizon framework utilizing the proposed robust model is presented and extensively tested in realistic settings. The robust model is benchmarked against a non-robust approach and shown to produce empty repositioning plans that can increase the service level appreciably for little extra cost.

(iv) The effects and benefits of sharing group configurations on (robust and non-robust) repositioning are analyzed. Sharing group design principles, based on geography and demand intensities, are derived.

The remainder of the paper is organized as follows. In Section 2, we give a brief overview of the relevant literature. In Section 3, we introduce the proposed robust optimization approach. Finally, we present and discuss the results of a comprehensive computational study in Section 4, and summarize our findings in Section 5.

2 Literature Review

Empty repositioning problems have been a research focus for a long time. Earliest dynamic models in the area utilize deterministic optimization, where uncertain demand is modeled via point estimates. Some examples of such studies are Leddon and Wrathall (1967) and Misra (1972), which study fleet management in rail operations, and White (1972), which focuses on container management. Deterministic models are still being used by freight transportation companies to manage their operations, since these models are easy to implement and solve.

Even though deterministic models that utilize point estimates for demand have been proven useful in real-life applications, their feasibility is susceptible to deviations of demand realizations from point estimates. To address this issue, expected cost minimization approaches, which require solving dynamic or stochastic programming models, are widely used. Early examples of this line of research, focusing on truckload trucking operations are Powell (1986), Powell (1987), Frantzeskakis and Powell (1990), and Cheung and Powell (1996). Later, adaptive estimation techniques are introduced for tractability of the dynamic programming approaches. Powell and Carvalho (1998) introduces linear functional approximations to capture the future effects of decisions made today; and non-linear functional approximations are introduced and effectively used in Godfrey and Powell (2002a) and Godfrey and Powell (2002b). From a similar perspective, Du and Hall (1997) and Hall and Zhong (2002) suggest using principles from inventory theory to obtain repositioning policies based on reorder points. They both propose these policies as a simple alternative to dynamic programming approaches, which are computationally burdensome.

Stochastic programming has been widely applied to model and solve dynamic empty repositioning problems subject to uncertainty. Crainic et al. (1993) proposes a two-stage restricted recourse model, where all empty container allocation decisions are assumed to be made in the first stage, never to be re-optimized in the future, and second-stage decisions only act as corrective actions to the pre-specified repositioning plan for the entire horizon. This type of approach is tractable, but restrictive in its assumptions. Cheung and Chen (1998), Long et al. (2012), and Long et al.

(2015) present two-stage stochastic models for empty repositioning and demonstrate that Sample Average Approximation (SAA) is an effective solution methodology for these stochastic models. Di Francesco et al. (2009) considers an empty container repositioning problem in maritime operations with multiple uncertain parameters with very low-quality data and proposes a multi-scenario optimization model, which can be solved via decomposition and parallel computing. In a related line of research, Di Francesco et al. (2013) studies an empty container repositioning problem with uncertain port disruptions and proposes a multi-scenario time-space formulation. Zhang et al. (2011) proposes to use a two-stage stochastic programming model iteratively to approximately solve a complex multi-stage stochastic optimization problem arising in drayage operations. A variant of the container positioning problem, which simultaneously considers empty and loaded container movements, is studied in von Westarp and Schinas (2016) from a fuzzy optimization perspective.

Robust optimization is another approach for modeling uncertainty, and is often preferred because of its simplicity, computational tractability, and ability to generate good quality solutions (in terms of recoverability) with insufficient or unreliable data. The first work in robust optimization with coefficient uncertainty is Soyster (1973). It shows that such uncertainties can be incorporated into a linear programming form and solved so that feasibility is guaranteed for all possible realizations of the uncertain coefficients. The foundations of robust convex optimization are established in Ben-Tal and Nemirovski (1998). Later, detailed analyses of robust linear programs are presented in Ben-Tal and Nemirovski (2000) and Bertsimas and Sim (2004). The latter introduces the idea of uncertainty sets with budgets, where the budget limits the number of uncertain coefficients that can simultaneously assume their worst-case value.

Robust discrete optimization is studied in Kouvelis and Yu (1997) and Bertsimas and Sim (2004). The former develops robust versions of many traditional discrete optimization problems, and the latter concentrates on network flow problems - particularly the theoretical and practical tractability of robust counterparts of polynomially solvable discrete optimization problems.

Ben-Tal et al. (2004) extends the robust linear optimization approach to cases where a subset of decisions are made before the realization of uncertain parameters and the remaining decisions can be made after the realization; calling the model Adjustable Robust Counterpart (ARC). Similar to the ARC approach, Atamtürk and Zhang (2007) develops a two-stage robust optimization approach for network flow and design problems with uncertain right-hand side, and underlines the importance of the ability of two-stage robust models to control the level of conservatism, unlike single-stage robust models where feasibility is guaranteed for all possible uncertainty realizations.

There is a wide range of recent studies that effectively implement robust optimization methods in dynamic resource management problems. Laporte et al. (2011) and Chung et al. (2010) concentrate on network design, where the latter proposes a tractable linear programming formulation for the robust optimization model. Ben-Tal et al. (2005) proposes adjustable robust formulations for the retailer-supplier flexible commitment problem, and Bertsimas and Thiele (2006) develops a robust formulation for inventory problems where demand uncertainty is modeled deterministically when certain network characteristics are present. See and Sim (2010) suggests robust formulations to the multi-period inventory problem, Chen and Iyengar (2012) proposes a method for the lot sizing model with uncertain and non-stationary demand structure, and Solyalı et al. (2012) and Solyalı et al. (2016) develop robust mixed-integer programs for the inventory routing problem with polyhedral demand uncertainty. Gabrel et al. (2014) provides a convex robust formulation for the location transportation problem. Shu and Song (2014) provides an in-depth analysis of a two-stage robust model for dynamic container deployment, which considers not only empty repositioning but also loaded container movements. Lee and Moon (2020) presents a multi-stage stochastic program and subsequently an adjustable robust optimization model for the empty container repositioning problem with foldable containers. Erera et al. (2009), which constitutes the basis for this study, derives a robust optimization framework for dynamic empty repositioning and develops feasibility conditions for various sets of allowed recovery actions.

3 Methodology

Consider a transportation service provider centrally managing a homogeneous fleet of reusable resources such as containers, railroad cars or trailers to serve a vast geographic area with a large number of terminals over a planning horizon of multiple time periods. Note that a time period can correspond to any measure of time, but it typically corresponds to a fraction of a day in this context. Load requests materialize over time and specify the number of resources to be transported from one terminal to another in a specific time period. Based on the known load requests as well as anticipated future load requests, central management determines a cost-effective empty repositioning plan to ensure resource availability to satisfy demand.

Before introducing the robust optimization model that addresses demand uncertainty, we first introduce the empty repositioning problem with deterministic demand. Assuming demand quantities are known with certainty at the beginning of the planning horizon, the traditional approach for finding a minimum-cost repositioning plan is to solve a deterministic minimum-cost flow prob-

lem with flow balance and flow bound constraints over the entire planning horizon, where flow lower bounds guarantee demand satisfaction. This deterministic problem will be referred to as the nominal repositioning problem (**NP**) from this point on.

To formally define the nominal problem, let D denote the set of terminals, and the planning horizon consist of ρ time periods $\{0, 1, 2, \ldots, \rho\}$. Let the set of nodes for each terminal $d \in D$ be $V^d = \{v_0^d, v_1^d, \ldots, v_\rho^d\}$, and $\mathcal{V} = \cup_{d \in D} V^d$. There are *inventory arcs* (v_t^d, v_{t+1}^d) for each $d \in D$ and $t = 0, 1, \ldots, \rho - 1$, and *repositioning arcs* $(v_t^i, v_{t+h_{ij}}^j)$ from terminal i to terminal j whenever repositioning is allowed between those terminals and the travel time is h_{ij} periods. Let $\mathcal{A}_I = \{(v_t^d, v_{t+1}^d) : d \in D, 0 \leq t < \rho\}$ denote the set of inventory arcs, and $\mathcal{A}_R$ denote the set of repositioning arcs, which represent the allowed repositioning moves in the network.

We distinguish between two types of repositioning arcs: *loaded arcs*, which represent load requests from specific origin terminals to specific destination terminals in specific time periods, and *empty arcs*, which represent the option to move resources empty between two terminals in specified time periods and are determined based on the carrier's empty repositioning strategy. Let $\mathcal{A}_R^L$ denote the set of loaded arcs, $\mathcal{A}_R^E$ denote the set of empty arcs. Then, $\mathcal{A}_R = \mathcal{A}_R^L \cup \mathcal{A}_R^E$, where

$$\mathcal{A}_R^L = \{(v_t^i, v_{t+h_{ij}}^j) : \text{there is a load request from } i \text{ to } j \text{ in period } t\}, \text{ and}$$
$$\mathcal{A}_R^E = \{(v_t^i, v_{t+h_{ij}}^j) : \text{there is the option of moving resources empty from } i \text{ to } j \text{ in period } t, \; 0 \leq t \leq \rho - h_{ij}\}.$$

Finally, let each node $v \in \mathcal{V}$ have a net supply $b(v)$, which represents the number of empty resources available at the corresponding terminal in the corresponding time period. Let s be an auxiliary sink node with net supply $b(s) = -\sum_{v \in \mathcal{V}} b(v)$ and $\mathcal{A}_s$ denote the auxiliary arcs entering sink node s. Then, the time-expanded network representing this transportation system is $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, where $\mathcal{N} = \mathcal{V} \cup \{s\}$ and $\mathcal{A} = \mathcal{A}_I \cup \mathcal{A}_R \cup \mathcal{A}_s$.

Using the above notation and terminology, the nominal problem can be formulated as

$$\textbf{NP} \qquad \min_{\mathbf{x} \in \mathbb{Z}_+^{|\mathcal{A}|}} \{\mathbf{c}^T \mathbf{x} : \mathbf{A}\mathbf{x} = \mathbf{b}, \; x(a) \geq \ell(a) \; \forall a \in \mathcal{A}_R^L\}, \tag{1}$$

where the decision vector $\mathbf{x}$ quantifies the resource flow on each arc, $\mathbf{c} \in \mathbb{R}_+^{|\mathcal{A}|}$ denotes the transportation cost vector, $\mathbf{A} \in \{0, 1, -1\}^{|\mathcal{V}| \times |\mathcal{A}|}$ denotes the node-arc incidence matrix associated with $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, and $\mathbf{b} \in \mathbb{Z}^{|\mathcal{N}|}$ and $\boldsymbol{\ell} \in \mathbb{Z}_+^{|\mathcal{A}_R^L|}$ denote node net supplies and flow lower bounds (dictated by load requests), respectively. Note that **NP** is polynomially solvable with standard minimum-cost

flow algorithms or linear programming.

Now suppose that the load requests on loaded arcs are uncertain. To address this uncertainty, we formulate a mathematical model based on the Adjustable Robust Counterpart (ARC) approach of Ben-Tal et al. (2004), to be referenced as the robust repositioning problem (**RP**), which supports an uncertain demand structure where only nominal quantities (historical averages, forecasts, etc.) are known for future load requests.

Let the nominal demand on loaded arc $a \in \mathcal{A}_R^L$ be $\ell(a)$. We assume that every potential demand realization $\tilde{\ell}(a) \in \mathbb{Z}_+$ falls in the interval $[\ell(a) - \hat{\ell}^-(a), \ell(a) + \hat{\ell}^+(a)]$, where $\hat{\ell}^-(a) \in [0, \ell(a)]$ and $\hat{\ell}^+(a) \geq 0$ denote the maximum negative and positive deviations from the nominal demand quantity, respectively.

The purpose of the robust model is to ensure that the repositioning plan is *recoverable*, *i.e.*, there exists a set of empty moves (referred to as *recovery actions*) that can alter the repositioning plan obtained prior to the materialization of uncertain demand, so that the altered plan will be feasible with respect to realized loaded demand quantities. It is possible to seek recoverability against all potential demand realizations, including a scenario where all loaded demand quantities simultaneously assume their worst-case values. It is also possible, and more realistic for practical purposes, to pursue a less conservative approach and seek recoverability against only a subset of potential demand realizations. To control the level of conservatism, we follow a method similar to the *budget of uncertainty* idea introduced in Bertsimas and Sim (2003).

To control how much of the demand uncertainty **RP** will hedge against, we define the following limited perturbation set φ_k, where we guarantee recoverability only in cases where a certain fraction of the worst-case deviations are realized:

$$\varphi_k = \{\boldsymbol{\delta} \in \mathbb{Z}^{|\mathcal{A}_R^L|} : \delta(a) \in [-k\hat{\ell}^-(a), k\hat{\ell}^+(a)] \quad \forall a \in \mathcal{A}_R^L\}.$$

Here, $k \in [0, 1]$ denotes the fraction of worst-case deviation from nominal demand for which **RP** will guarantee recoverability. Note that when $k = 0$, every realization is assumed to conform to nominal, and when $k = 1$, the robust model hedges against all potential realizations $\tilde{\boldsymbol{\ell}} \in [\boldsymbol{\ell} - \hat{\boldsymbol{\ell}}^-, \boldsymbol{\ell} + \hat{\boldsymbol{\ell}}^+]$.

For assessing the recoverability of repositioning plans found by **RP**, we first need to define the set of allowed recovery actions. As mentioned in the introduction, the cost of recovery actions will be ignored so that the originally two-stage problem can be reduced to a single integer program. Because the cost of recovery is not going to be explicitly modeled and minimized in the robust

problem, it is crucial that the set of recovery actions consists only of low-cost moves. Given the set of recovery arcs $\mathcal{R} \subseteq \mathcal{A}_R$, the set of recovery actions is defined as:

$$W = \{\mathbf{w} \in \mathbb{Z}^{|\mathcal{A}|} : w(a) \geq 0 \quad \forall a \in \mathcal{R}, \quad w(a) = 0 \quad \forall a \in \mathcal{R}_0\}$$

where $\mathcal{R}_0 = \{(v_0^d, j) \in \mathcal{R} : d \in D, j \in \mathcal{V}\}$ denotes the set of recovery arcs associated with the initial time period. We assume that the decisions associated with the initial period are fixed and therefore cannot be altered with recovery actions, *i.e.*, we assume the resources will have already left their original locations and are en-route to their intended destinations by the time recovery actions are to be taken, and hence cannot be rerouted. Similar recovery sets can be defined if the decisions associated with multiple periods are assumed to be fixed.

Using the definitions of limited perturbation set φ_k and set of recovery actions W, a formulation for **RP** is

$$\begin{aligned} \mathbf{RP}(W,k) \qquad \min_{\mathbf{x} \in \mathbb{Z}_+^{|\mathcal{A}|}} \{\mathbf{c}^T\mathbf{x} : \quad & \mathbf{x} \in X(\boldsymbol{\ell}), \\ & \forall\ \boldsymbol{\ell} + \boldsymbol{\delta} \in \mathcal{Z}_k \quad \exists \mathbf{w} \in W : \mathbf{x} + \mathbf{w} \in X(\boldsymbol{\ell} + \boldsymbol{\delta})\} \end{aligned} \tag{2}$$

where $\mathcal{Z}_k = \{\tilde{\boldsymbol{\ell}} \in \mathbb{Z}_+^{|\mathcal{A}_R^L|} : \ell(a) - k\hat{\ell}^-(a) \leq \tilde{\ell}(a) \leq \ell(a) + k\hat{\ell}^+(a)\ \forall a \in \mathcal{A}_R^L\}$ denotes the set of possible joint demand realizations corresponding to the limited perturbation set φ_k, $X(\boldsymbol{\ell}) = \{\mathbf{x} \in \mathbb{Z}^{|\mathcal{A}|} : \mathbf{A}\mathbf{x} = \mathbf{b}, x(a) \geq \ell(a)\ \forall a \in \mathcal{A}_R^L\}$ denotes the set of feasible flows with respect to flow balance and nominal demand satisfaction constraints, and $X(\boldsymbol{\ell} + \boldsymbol{\delta})$ is the set of feasible flows with respect to flow balance and perturbed demand satisfaction constraints.

We can write **RP** in extended form as

$$\begin{aligned} \mathbf{RP}(W,k) \qquad \min_{\mathbf{x} \in \mathbb{Z}_+^{|\mathcal{A}|}, \mathbf{w} \in \mathbb{Z}_+^{|\varphi_k| \times |\mathcal{A}|}} \{\mathbf{c}^T\mathbf{x} : \quad & \mathbf{x} \in X(\boldsymbol{\ell}), \\ & \mathbf{x} + \mathbf{w}_{\boldsymbol{\delta}} \in X(\boldsymbol{\ell} + \boldsymbol{\delta}) \quad \forall\ \boldsymbol{\delta} \in \varphi_k, \\ & \mathbf{w}_{\boldsymbol{\delta}} \in W \quad \forall\ \boldsymbol{\delta} \in \varphi_k\}. \end{aligned} \tag{3}$$

Since **RP** is mainly aimed at guaranteeing feasible recovery and therefore does not take into consideration the cost of recovery, there is no cost component associated with the set of recovery decisions, $\mathbf{w}$. This feature constitutes the basis for the more tractable alternative robust formulation we propose, which guarantees recoverability not by explicitly deciding on the recovery actions but by enforcing a set of *recoverability constraints* in addition to the constraints of the nominal problem

(**NP**). To further describe the recoverability constraints, it is necessary to define *vulnerability of node sets* and *inbound-closed node sets*.

Definition 1 (Node Set Vulnerability) *For a set of nodes $U \subseteq \mathcal{N}$, the vulnerability $\vartheta(U,k)$ is defined as*

$$\vartheta(U,k) = \sum_{\substack{a=(i,j)\in\mathcal{A}_R^L \\ i\in U, j\in\Delta^{out}(i)\setminus U}} k\hat{\ell}^{+}(a) + \sum_{\substack{a=(i,j)\in\mathcal{A}_R^L \\ i\in\Delta^{in}(i)\setminus U, j\in U}} k\hat{\ell}^{-}(a).$$

Note that when the uncertainty set is φ_k, the vulnerability of node set U represents the maximum excess of loaded outflow from node set U and the maximum shortage of loaded inflow into node set U, multiplied by k.

Definition 2 (Inbound-closed Node Set) *Let $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ denote a network. A set of nodes $C \subseteq \mathcal{N}$ is inbound-closed if there exists no directed path in $\mathcal{G}$ from any node $i \in \mathcal{N} \setminus C$ to any node $j \in C$.*

Let $\mathcal{G}_W = (\mathcal{N}, \mathcal{A}_W)$ represent the recovery network induced by the set of recovery actions W, where $\mathcal{A}_W$ denotes the set of recovery arcs. It is important to observe that in $\mathcal{G}_W$, an inbound-closed node set has no possible incoming recovery flow. Therefore, such sets need to contain enough pooled inventory to guarantee recovery even for worst-case demand realizations. In what follows, we provide the necessary and sufficient conditions for guaranteeing the existence of a recovery flow. The proofs are relatively straightforward adaptations of the original proofs in Erera et al. (2009), but are provided in Appendix A for the sake of completeness.

Let $\mathcal{G}_W(\mathbf{x}, \boldsymbol{\delta})$ represent the recovery network given a repositioning plan $\mathbf{x}$ and a specific perturbation realization $\boldsymbol{\delta} \in \varphi_k$, and $\mathbf{I} \in \mathbb{Z}^{|\mathcal{N}|}$ denote the marginal net inventory of resources available (or needed) in the recovery problem, based on $\mathbf{x}$ and $\boldsymbol{\delta}$.

$$\begin{aligned}
&I(v_0^d) = 0, \quad d \in D \\
&I(v_1^d) = x(v_1^d, v_2^d) + \sum_{i\in\Delta^{in}(v_1^d)} \delta(i, v_1^d) - \sum_{j\in\Delta^{out}(v_1^d)} \delta(v_1^d, j), \quad d \in D \\
&I(v_t^d) = x(v_t^d, v_{t+1}^d) - x(v_{t-1}^d, v_t^d) + \sum_{i\in\Delta^{in}(v_t^d)} \delta(i, v_t^d) - \sum_{j\in\Delta^{out}(v_t^d)} \delta(v_t^d, j), \quad d \in D, 1 < t \le \rho \\
&I(s) = -\sum_{v\in\mathcal{V}} I(v)
\end{aligned}$$

Proposition 1 *A feasible solution* $\mathbf{x}$ *of* **NP** *is recoverable with recovery action set* W *after a perturbation realization* $\boldsymbol{\delta}$ *if and only if there exists a feasible flow in* $\mathcal{G}_W(\mathbf{x}, \boldsymbol{\delta})$.

Proposition 2 *Let* $\mathcal{U}_W$ *denote the collection of inbound-closed node sets in* $\mathcal{G}_W$. *There exists a feasible flow in* $\mathcal{G}_W(\mathbf{x}, \boldsymbol{\delta})$ *if and only if for every set of nodes* $U \in \mathcal{U}_W$

$$\sum_{v \in U} I(v) \geq 0$$

Theorem 1 *A feasible solution* $\mathbf{x}$ *of* ***NP*** *is also feasible for the robust problem* ***RP*** *if and only if for every node set* $U \in \mathcal{U}_{\mathcal{W}}$

$$\sum_{a \in \Delta^{out}(U) \cap \mathcal{A}_I} x(a) \geq \vartheta(U, k). \tag{4}$$

This theorem provides the following alternative formulation for the robust model **RP**:

$$\mathbf{ARP}(W, k) \qquad \min\{\mathbf{c}^T\mathbf{x} : \quad \mathbf{x} \in X(\boldsymbol{\ell}), \sum_{a \in \Delta^{out}(U) \cap \mathcal{A}_I} x(a) \geq \vartheta(U, k) \quad \forall\, U \in \mathcal{U}_W\} \tag{5}$$

Observe that the number of variables in **ARP** is equal to the number of variables in **NP**, whereas **RP** models all recovery actions explicitly and therefore introduces a very large number of variables. Additionally, **ARP** has significantly fewer constraints compared to **RP** when the recovery network is designed pragmatically, *i.e.*, when only a small number of low-cost moves are allowed for recovery flow. It is particularly important to note that the number of recoverability constraints in **ARP** is only dependent on the structure of the recovery network and not on the possible realizations in uncertainty set φ_k. Even though the number of such constraints are directly related to the number of inbound-closed node sets in the recovery network, which may be exponential, it is manageable when the recovery network is kept relatively small.

In transportation applications, empty repositioning decisions are typically made and implemented in a rolling horizon framework, where the optimization model is solved for a given planning horizon but only today's decisions are implemented. Subsequently, model parameters are updated with the implemented decisions and tomorrow's realized demand quantities, and the optimization model can be solved once more to obtain decisions for tomorrow; and so on.

The rolling horizon implementation justifies our two-stage modeling approach to represent a naturally multi-stage setting, where realized demand quantities are revealed at each time period. In

the two-stage robust model, the first stage represents the repositioning decisions made prior to the materialization of demand, and the second stage represents the recovery decisions made after the realized demand quantities are observed. Using this model in a rolling horizon framework provides opportunities to guarantee recoverability every time the horizon is rolled forward and the next period's demand is materialized, much like seeking recourse options in a multi-stage stochastic programming model.

When implementing **ARP** with rolling horizon, it may suffice for practical purposes to guarantee recoverability only for the near future rather than the entire planning horizon, *(i)* as a measure to prevent overconservatism, and *(ii)* since we will have other opportunities to ensure future recoverability after recent decisions are implemented and the horizon is rolled. To demonstrate this idea, consider a time-space network where the travel time between terminals is no more than one period and repositioning is allowed between all pairs of terminals. Then, ensuring recoverability of only the immediate next period's repositioning actions would suffice, since recovery actions for the periods that are further into the future will still be available after the horizon is rolled. Importantly, in this case, even though feasibility is not a concern, looking further into the future for recoverability can provide more cost-effective repositioning plans by taking advantage of cheaper repositioning moves that are available in the future and therefore would be missed by the myopic approach of ensuring recoverability only for the immediate next period. Therefore, it is practical to consider recoverability for a reasonable number of periods into the near future, without being overly myopic or conservative. With this idea, instead of enforcing the recoverability constraints (4) for all node sets that are inbound-closed in $\mathcal{G}_W$, we enforce such constraints for a limited subcollection of inbound-closed node sets. For this, we define a *robustness horizon*, R, and limit the collection of inbound-closed sets $\mathcal{U}_W$ to include only the sets that contain nodes associated with time periods that are no more than R periods into the future. This approach helps in controlling the level of conservatism, as well as limiting the number of recoverability constraints (4) in **ARP** for computational tractability.

To further control the level of conservatism in **ARP**, we define another robustness parameter, N, which represents the maximum number of terminals contributing to an inbound-closed node set. Introducing such a parameter to control conservatism is motivated by the observation that it is not likely in realistic instances for worst-case empty resource deficits to occur in a large number of terminals simultaneously. Therefore, imposing recoverability constraints for the inbound-closed node sets with a large number of terminals is likely to be too conservative for practical purposes, as well as being computationally burdensome. With this idea, we limit $\mathcal{U}_W$ to include only the inbound-closed sets that contain nodes associated with a maximum of N terminals.

To summarize, the level of conservatism associated with **ARP** is controlled by:

(i) $\vartheta(U, k)$, right-hand side of the recoverability constraints (4)

(ii) $\mathcal{U}_W$, collection of inbound-closed node sets in $\mathcal{G}_W$ (the number and structure of the node sets)

Controlling the right-hand side is possible by adjusting the value of the *vulnerability multiplier*, k, which has no effect on the computational burden of building and solving the model. On the other hand, controlling the number and structure of inbound-closed node sets, which is possible by changing the values of robustness parameters R and N, directly affects the computational burden of **ARP**. When not controlled via the robustness parameters, **ARP** can become intractable due to a potentially exponential number of recoverability constraints. However, when the recovery network $\mathcal{G}_W = (\mathcal{N}, \mathcal{A}_W)$ is designed pragmatically, limiting $\mathcal{U}_W$ by robustness parameters R and N can provide tractable **ARP** models.

4 Computational Experiments

In this section, we first describe the testing framework that is devised to analyze the proposed robust optimization methodology, and then present and discuss results of our computational study. To assess the efficiency and performance of the proposed methodology, we use a simulation framework. Furthermore, to evaluate the value of robustness, we conduct computational experiments using the robust model **ARP** in comparison to the nominal model **NP**. In the simulation framework, each day, a set of load requests materializes and an optimization model is solved to determine an empty repositioning plan. Based on this plan, the current day's repositioning moves are dispatched, the net supply quantities are updated based on the dispatched vehicles, and the horizon is rolled forward to start from the next day.

4.1 Problem Instances

The basis for our computational study is a set of instances that are representative of real-life empty repositioning settings. Multiple instances (representing multiple demand realizations) are randomly generated for the same underlying network geography and demand distribution. The inputs to the instance generation process are the geographical locations of a set of terminals and the probabilities

of each terminal being the origin or destination of a load request. With this information, nominal demand quantities and deviations from the nominal values are obtained via Monte Carlo simulation, and then the fleet size and initial resource locations are computed.

In generating the problem instances, we assume that a constant number of load requests materialize each day of the planning horizon. Each load request is the result of a multinomial trial with the origin and destination terminals determined using the origin and destination probabilities associated with each terminal. To generate realistic instances, it is assumed that the transportation carrier operates a hub-and-spoke network, where regional hubs act as consolidation centers for their spokes (non-hub terminals). Load requests are routed so that loads first go from the origin terminal to the origin regional hub, then, if necessary, from the origin regional hub to the destination regional hub, and, finally, from the destination regional hub to the destination terminal. Thus, even though a load request can originate at any terminal and be destined for any terminal, it is routed along the appropriate path in the hub-and-spoke network.

Once load requests are generated and routed in the described manner for a large number of days, average loaded demand is calculated for every arc $a \in \mathcal{A}_R^L$ (in the hub-and-spoke network) and that quantity is recorded as the nominal demand quantity, denoted as $\ell(a)$. The maximum positive and negative deviations from nominal values, denoted as $\hat{\ell}^+(a)$ and $\hat{\ell}^-(a)$, are also recorded.

Using the nominal demand quantities, we determine a fleet size that can feasibly satisfy the nominal demand and realistically accommodate some demand uncertainty. To achieve this, we solve a minimum cost network flow problem on a wrapped time-expanded network, which represents the transportation system, for a planning horizon of one week. The minimum fleet size required to satisfy the nominal loaded demand is then increased by a factor to account for demand uncertainty. In order to obtain meaningful results, 30 instances are generated for any given network geography and demand structure, and computational results are presented as averages of these 30 instances.

4.2 Testing Framework

In order to test the performance of the described robust optimization approach for empty repositioning problems, we develop a simulation framework that mimics the daily generation and execution of repositioning plans. For each day of the planning horizon, a time-expanded network is generated, a variant of **ARP** is solved, the repositioning decisions of the current day are executed, and the horizon is rolled to start from the next day. The time-expanded network uses homogeneous time pe-

riods of six hours and covers a workweek, *i.e.*, five working days. Consequently, the time-expanded network has $5 \times 4 \times |D|$ nodes (plus one auxiliary sink node).

The variant of **ARP** that we use in our experiments, **ARP′**, differs from the original **ARP** in allowing corrective action in case the nominal demand satisfaction and recoverability cannot be feasibly attained with the current fleet. **ARP′** explicitly models the option of corrective action, *i.e.*, outsourcing of unmet demand, to ensure feasibility. For this, we introduce decision variables for corrective action, $\mathbf{y}$, to the original **ARP** formulation and assign a sufficiently large penalty cost so that corrective action occurs only when it is unavoidable.

ARP′ is formulated as follows:

$$\mathbf{ARP'}(W,k) \quad \min_{\substack{\mathbf{x}\in\mathbb{R}_+^{|\mathcal{A}|} \\ \mathbf{y}\in\mathbb{R}_+^{|\mathcal{A}_R^L|}}} \{\mathbf{c}^T\mathbf{x} + M\mathbf{1}^T\mathbf{y} : \sum_{a\in\Delta^{out}(i)} x(a) - \sum_{a\in\Delta^{in}(i)} x(a) = b_i \quad \forall i \in \mathcal{N} \tag{6}$$

$$x(a) + y(a) \geq \ell(a) \quad \forall a \in \mathcal{A}_R^L \tag{7}$$

$$\sum_{a\in\Delta^{out}(U)\cap\mathcal{A}_I} x(a) \geq \vartheta(U,k) \quad \forall U \in \mathcal{U}_W\}, \tag{8}$$

where $M > 0$ is a sufficiently large positive number that represents the cost of corrective action.

When **ARP′** is solved for a week of five workdays starting from today, the outcome is an empty repositioning plan and possibly a set of corrective actions. In the simulation framework, the next step is implementing the moves (*i.e.*, served load requests, corrective actions, and empty repositioning) of the first day, which translates into adjusting the net supply values at the nodes impacted by these moves.

After today's moves are executed, the horizon is rolled forward by one day, and tomorrow's demand is observed. In order to roll the horizon, the nodes associated with today's periods are deleted from the time-expanded network, along with the relevant inventory and repositioning arcs. Furthermore, nodes and arcs for one more day are appended to the time-expanded network. Once the network is updated with the rolling horizon logic, we assume that one day has passed and that load requests for the (new) first day are observed. To reflect the load requests that have materialized, the flow lower bounds, $\ell(a)$, of the relevant arcs are changed from nominal to realized demand quantities. An algorithmic summary of the testing procedure is presented in Algorihtm 1.

To assess the value of robustness, we compare the performance of the empty repositioning plans

Algorithm 1 Testing Framework

for $t \in \{1, 2, 3, \ldots, 20\}$ **do**
  Observe real demand of day t
  Generate $\mathcal{G}_t$, the time-expanded network starting with day t
  Solve **ARP′** on $\mathcal{G}_t$
  Implement decisions associated with day t
end for

generated by the robust approach to those generated without robustness consideration. The approach without robustness consideration simply replaces **ARP′** with **NP′** in Algorithm 1, where **NP′** denotes a variant of **NP** that allows corrective action for feasibility.

4.3 Computational Results

We test the performance of various robust and non-robust approaches on geographies with 30, 80, and 100 terminals. Section 4.3.1 focuses on analyzing the impact of the choice of robustness parameters, *i.e.*, the robustness horizon (R), the number of terminals in an inbound closed set (N), and the vulnerability multiplier (k). The 30, 80, and 100-terminal geographies are divided into 3, 5, and 7 sharing groups, respectively, and the sharing groups are designed so that the terminals in sharing groups are within close proximity to each other and the number of terminals in each sharing group is the same (or as close to each other as possible). Section 4.3.2 focuses on the impact of the sharing group configuration, *i.e.*, the number and make-up of the sharing groups, and therefore various sharing group configurations are explored in detail for a 30-terminal geography.

The performance measures that we focus on are service level (fraction of load requests satisfied without any corrective action), total transportation cost incurred by the carrier, and solution time.

In the body of the text, we mainly rely on graphs and figures to present the results of computational experiments, but, for completeness, tables with detailed results are provided in Appendix B.

4.3.1 Value of Robustness

In this section, we concentrate on the benefits of robust, as opposed to nominal, empty repositioning; as well as the impact of level of conservatism on solution quality. All computational experiments reported in this section are conducted with uncertainty sets $[\max\{0, \ell - k\hat{\ell}^-\}, \ell + k\hat{\ell}^+]$, where

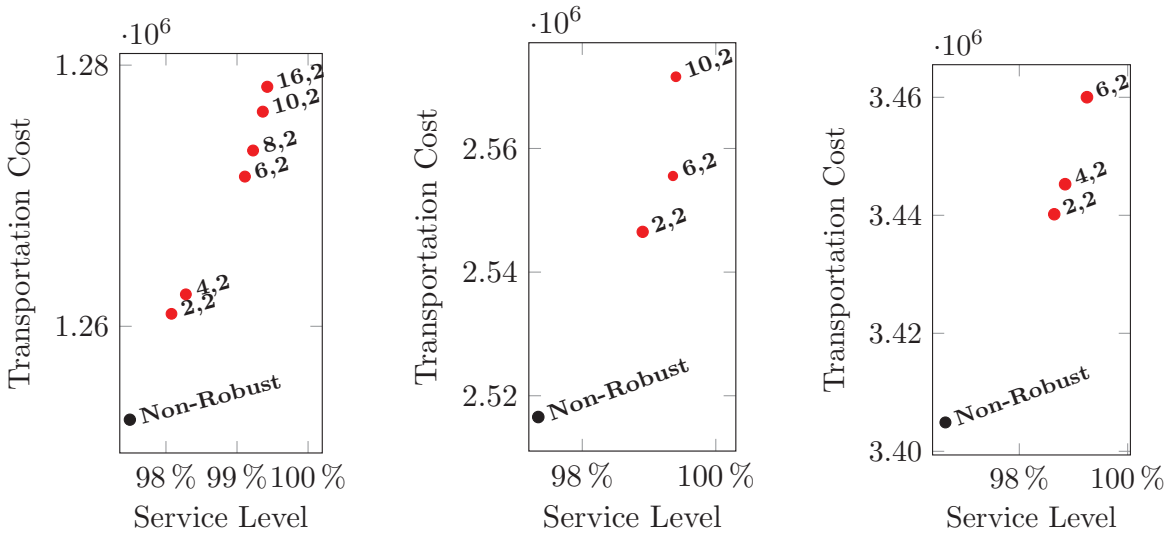
the demand realizations come from a uniform distribution with support $[\max\{0, \ell - \hat{\ell}^-\}, \ell + \hat{\ell}^+]$. Different values for the parameter k, which specifies the fraction of the worst-case deviation from nominal demand that the recoverability constraints are to hedge against, are considered in the test setting. These values are 1, 1/2, 1/3, 1/4. Note that higher values of k correspond to higher levels of protection (*i.e.*, are more conservative).

The robustness horizon (R) and the maximum number of terminals contributing to an inbound-closed node set (N) affect solution performance both in terms of solution quality and computational tractability. Therefore, different values for those parameters are considered in our computational experiments, namely 2, 4, 6, 8, 10, 16, and 20 for R and 2, 3, 4 and 5 for N. In the visual representation of results, a marker label (R, N) is representative of a test case where at most R time periods are accounted for in the recoverability constraints, and there are at most N terminals in each inbound-closed node set.

The robust approach, as opposed to a non-robust (*i.e.*, nominal) approach, moves the empty resources to "the right places at the right times" so that the terminals that are more vulnerable to demand uncertainty can be feasibly served with recovery moves. Naturally, while ensuring that the resources are at the right places at the right time, the robust approach moves the resources around more than a nominal approach. It is important to note that in our experimental setting, we place emphasis on improving the service level (and not so much on reducing the total transportation cost), so we set the unit cost of corrective action to be very high. However, it is also possible to set the unit cost of corrective action closer to the unit transportation cost, in order to represent environments where incurring a low total transportation cost is a comparably important goal to improving the service level.

Figure 1 presents total transportation cost and service level for different levels of conservatism on 30-, 80-, and 100-terminal networks. It illustrates the effect of robustness in improving service level while causing a slight increase in total transportation cost. Tables 1, 2, and 3 of Appendix B provide further details and demonstrate that a considerable increase in service level can be achieved with an acceptable increase in computation time when robustness parameters are selected pragmatically. In the discussion to follow, we study the effects of each of the three robustness parameters (R, N, and k) independently.

First, we study the effects of limiting the robustness horizon, R. As seen in Figure 1a, when the values of k and N are kept constant, increasing R results in substantial increase in service level up to a certain point, around $R = 10$. However, as R increases from 10 to 16, the increase in service



(a) 30 Terminals (b) 80 Terminals (c) 100 Terminals

Figure 1: Effect of R on service level and transportation cost ($k = 1/3$, averaged over 30 instances)

level is less significant and it comes at (relatively speaking) greater increase in transportation cost. Furthermore, as we increase R from 16 to 20 (not included in the figure because of difficulty in viewing), service level actually decreases, from 99.43% to 99.40%, contrary to the decreasing trend observed for lower R values. This change in the trend can intuitively be explained with overconservatism, since a robust approach looking too far into the future can accumulate resources at terminals that have high demand in the distant future at the expense of requiring corrective action in the present. Therefore, the value of R must be chosen so that the resulting repositioning plan reflects the service level advantage of robustness and not the adverse effects of overconservatism.

Second, we focus on the effects of limiting N, the maximum number of terminals in an inbound-closed node set. Since increasing N increases the number of inbound-closed node sets (and hence the number of recoverability constraints) exponentially, it was not practical to test for very large values of N, especially in larger geographies. But it can be clearly seen in Figure 2 that increasing N causes a considerable increase in computation time, while not substantially improving the service level. We observed a similar trend in the outcomes of our experiments with $N = 2$ and $N = 3$ in larger geographies. Generally, considering more than two terminals in a recoverability constraint does not improve overall solution performance significantly. This outcome was expected, since we only allow empty repositioning between empty hubs and their spokes and not between spokes. Setting $N = 2$ corresponds to having every spoke with its designated hub in an inbound-closed node set, and allowing more terminals in such sets by increasing N only slightly improves the service level.

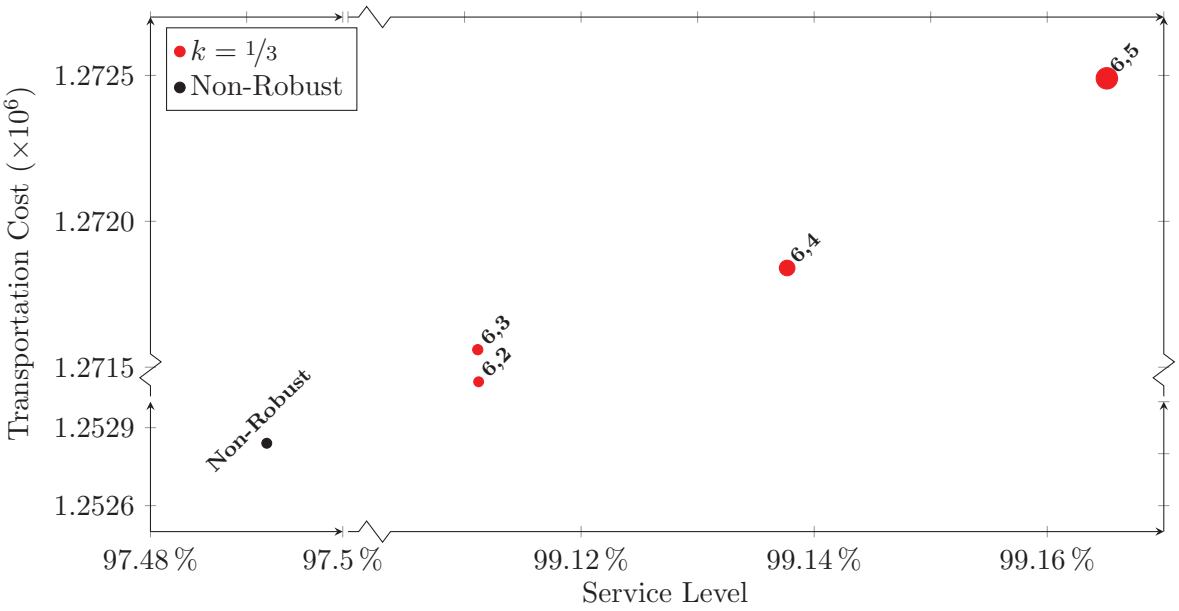
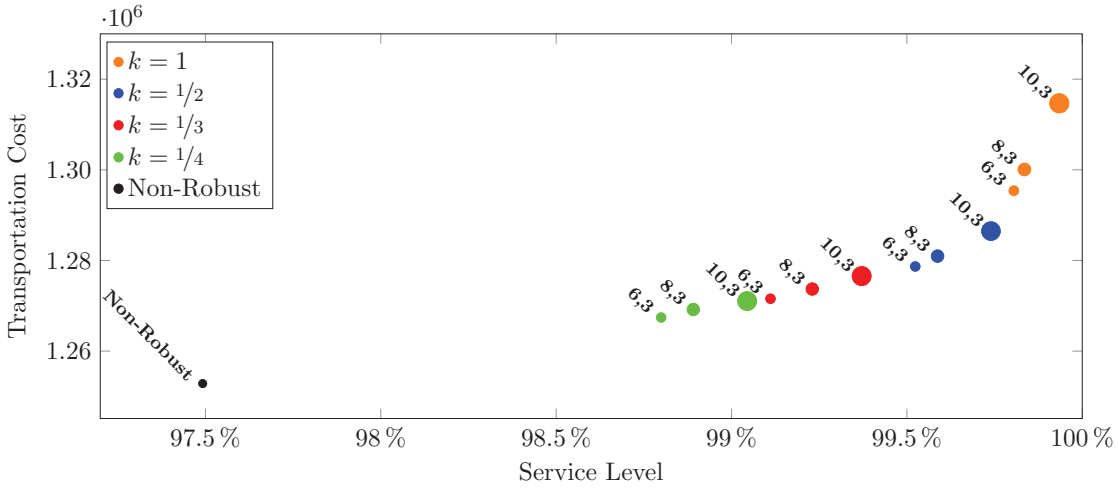


Figure 2: Effect of N in a 30-terminal geography (averaged over 30 instances) (Marker size is proportional to the natural logarithm of total computation time.)

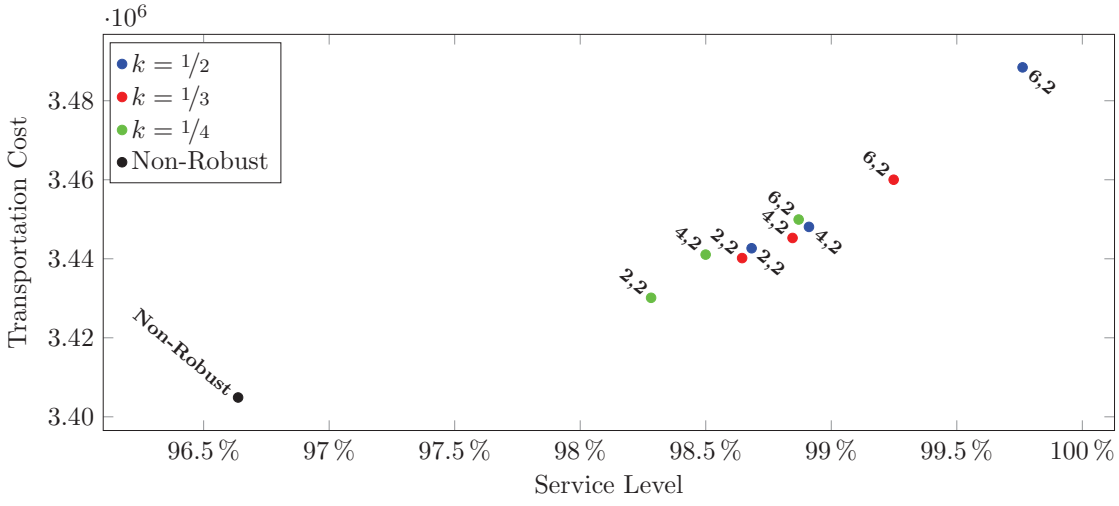
Finally, the impact of vulnerability multiplier, k, which dictates how wide the robust uncertainty interval is compared to the underlying demand distribution, on solution quality is investigated. Examining Figure 3, it is clear that on average, as k increases from $1/4$ to $1/2$, service level increases without causing the total transportation cost to increase by much and without a noticeable increase in computation time. However, as can be seen especially in the 30-terminal geometry, increasing k to 1, *i.e.*, hedging against the worst-case demand realizations, can result in only slightly better service levels, but at significantly higher total transportation costs. Furthermore, in some individual test instances, we observe lower service levels than the non-robust approach when hedging against worst-case demand realizations by setting $k = 1$. Similar to increasing R excessively, this is due to overconservatism. When $k = 1$, the repositioning plans are protected against all uncertainty, which means that the plan can be recovered even when all future loaded demand is realized at the worst-case values. Generally, this much conservatism is unnecessary for practical purposes, and it generates solutions that are even more vulnerable to uncertainty than those produced by a non-robust approach, by accumulating resources at certain terminals for potential future need at the cost of a reduced service level at other terminals today.

In our computational experiments where *(i)* about 10% of the terminals act as empty hubs, *(ii)* the maximum travel time between an empty hub and a designated spoke and between two empty hubs are 3 and 10 periods, respectively, and *(iii)* uncertain demand quantities follow a uniform distribu-

tion, we observe that the proposed robust methodology with robustness parameters $(R, N) = (6, 2)$ and $k = 1/3$ produces high-quality repositioning plans in acceptable computation times, *i.e.*, about 250 seconds for instances with 30 terminals, about 4,300 seconds for instances with 80 terminals, and about 10,400 seconds for instances with 100 terminals. In a similar manner, robustness parameters that would provide the most desirable outcome in terms of service level, total transportation cost, and computational burden can be determined for different geographies and network parameters.



(a) 30-terminal geography



(b) 100-terminal geography

Figure 3: Effect of k (averaged over 30 instances) (Marker size is proportional to total computation time.)

4.3.2 Sharing Group Configuration

In this section, we concentrate on the impact of sharing group configuration on service level and total transportation cost. More specifically, we investigate the effects of *(i)* the number and size of the sharing groups; and *(ii)* the volume of nominal inbound and outbound loaded demand of sharing groups. All computational experiments reported in this section are conducted on a 30-terminal geography with either 3, 5, or 7 sharing groups, where the number of terminals in each sharing group is the same (or as close to each other as possible), and each sharing group is either balanced, in terms of inbound and outbound loaded demand, or imbalanced. The different sharing group configurations are shown in Figure 4.

First, we examine the effects of sharing group balance in terms of inbound and outbound loaded demand. An imbalanced sharing group configuration has some sharing groups that dominantly act as the origin or the destination of load requests, whereas in a balanced configuration all sharing groups have roughly equal inbound and outbound load requests. It can be clearly seen in Figure 5 that the adverse effects of uncertainty (lower service level and excessive empty repositioning) are observed more strongly in a system with an imbalanced sharing group configuration (for both robust and non-robust empty repositioning plans). A balanced sharing group configuration provides an extra layer of risk pooling, so that within each sharing group, terminals with lower-than-expected net loaded demand can provide empty resources to those with higher-than-expected net loaded demand. This way, *(i)* higher service levels can be achieved with fewer corrective actions, as resources are more likely to be available within the sharing group when needed, and *(ii)* less empty repositioning occurs between sharing groups, as sharing groups are more self-sufficient in terms of demand and supply of empty resources.

Next, we focus on how the number and size of sharing groups affect service level and total transportation cost. Figure 6 reveals patterns that can be explained with risk pooling and average travel distance within and between sharing groups. When the number of sharing groups increases from 3 to 5, the larger number of empty hubs eases the demand intensity on each hub, hence lessening the adverse effects of demand uncertainty (in both robust and non-robust empty repositioning plans). It is also worth noting that a sharing group configuration with 3 sharing groups naturally has sharing groups that cover a larger geographic area, and thus longer spoke-hub distances and therefore higher empty repositioning costs within a sharing group. When the number of sharing groups increases from 5 to 7, the decrease in sharing group size causes a decrease in the risk pooling within the sharing groups, and, as a consequence, an increase in empty repositioning between sharing groups. This leads to *(i)* greater total transportation cost, as the travel distances between

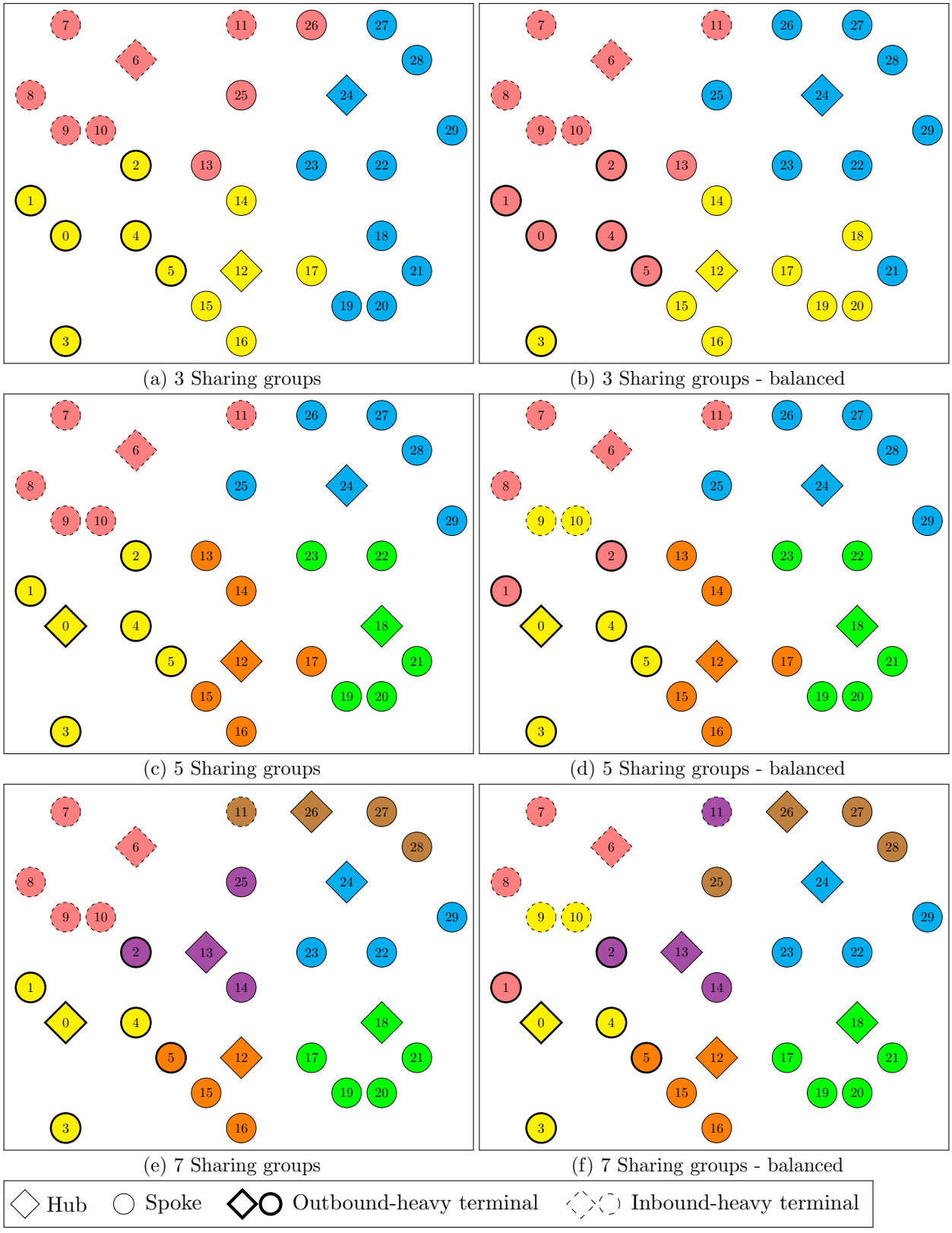


Figure 4: Sharing group configurations for 30-terminal geography

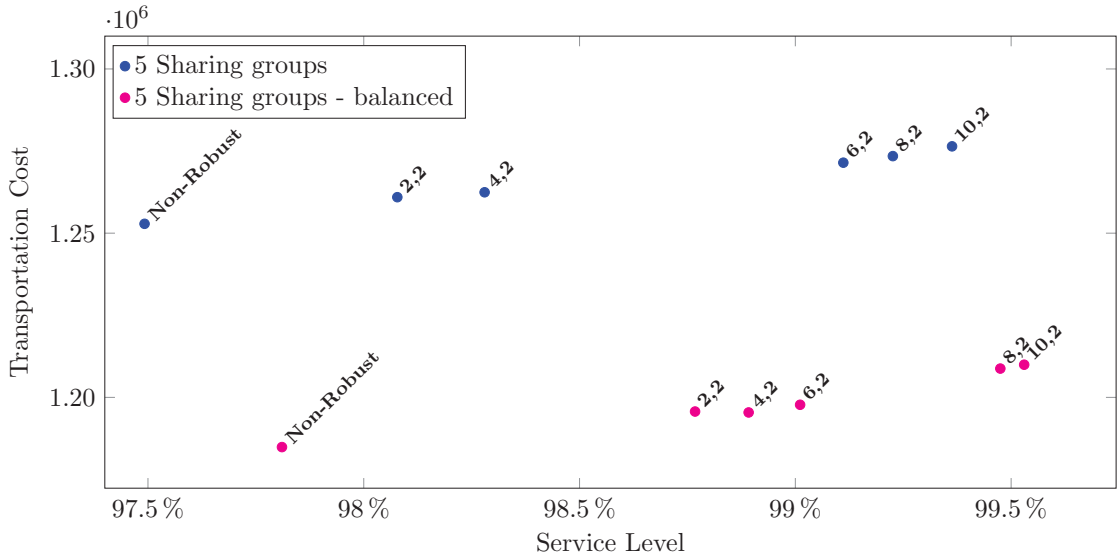


Figure 5: Effect of inbound/outbound loaded demand balance (30 terminals, $k = 1/3$, averaged over 30 instances)

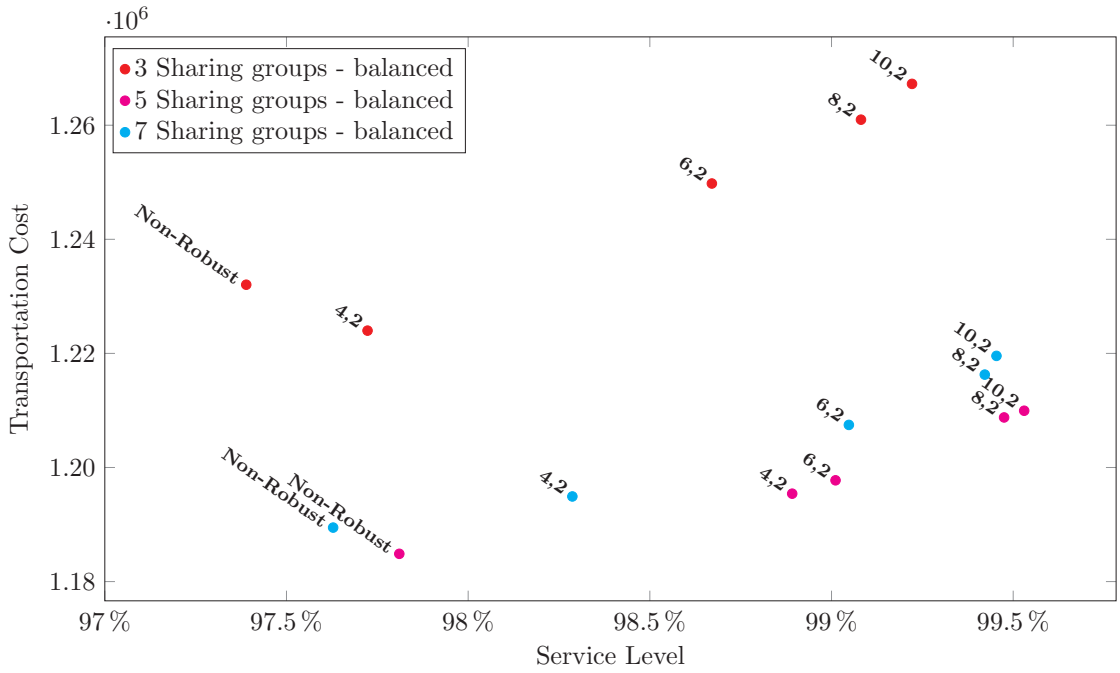


Figure 6: Effect of sharing group size (30 terminals, $k = 1/3$, averaged over 30 instances)

sharing groups are relatively long compared to those within sharing groups, and *(ii)* lower service levels, as the longer intergroup travel times can cause empty resources to arrive too late.

For every geography and demand intensity, sharing group configurations with appropriately sized

and balanced (in terms of inbound and outbound loaded demand) groups can be designed in order to take advantage of pooling benefits.

5 Conclusion

We have introduced a rolling horizon framework, which features a robust optimization model, for managing empty repositioning of resources in the presence of future demand uncertainty. The proposed robust model finds repositioning plans that are feasible for nominal demand quantities and recoverable (by using a predefined set of recovery actions) for every possible demand realization, or a subset of demand realizations. By using this two-stage robust model in a rolling horizon framework, we accurately represent the multi-stage nature of a typical transportation carrier's day-to-day operations. In order to pragmatically control the level of conservatism of the robust model, we defined model-specific robustness parameters.

We demonstrated the value of this framework with a comprehensive computational study, which includes problem instances with various network and fleet sizes, robustness parameters, and recovery action sets. Even though in our computational study, we concentrate on instances with *(i)* hub-and-spoke networks (and hub-to-spoke recovery actions), *(ii)* relatively small fleets, and *(iii)* relatively low demand imbalance, and prioritize service level over other performance measures (such as incurred transportation costs), the robust rolling horizon framework can be used with appropriate network and robustness parameters to accurately represent numerous network structures, demand distributions, and objectives. The computational results illustrate that when the robustness parameters are selected pragmatically, this framework can improve service level significantly compared to a nominal approach, while causing only a slight increase in transportation cost. Robustness parameters also help in keeping the formulation tractable, as the robust model tends to have a prohibitively large number of recoverability constraints when it aims for recoverability against all possible demand realizations.

Finally, we investigated how various sharing group configurations can affect the quality of the repositioning plans in terms of operational and computational simplicity. By designing the sharing group configuration in a way that balances expected transportation volumes within and between sharing groups, we were able to improve service level and reduce transportation cost in our test instances.

Appendix A Proofs of Propositions and Theorems

Proof (Proposition 1): By construction of the network and the associated net supply vector $\mathbf{I}$, a feasible flow in $\mathcal{G}_W(\mathbf{x}, \boldsymbol{\delta})$ defines a set of recovery moves $\mathbf{w} \in \mathbb{Z}^{|\mathcal{A}|}$ that would recover the feasibility of $\mathbf{x}$ after the realization of $\boldsymbol{\delta}$. For repositioning arcs $a \in \mathcal{A}_R$, $w(a)$ would be the flow on arc a minus $(x(a) + \delta(a))$, and for inventory arcs $a \in \mathcal{A}_I$, $w(a)$ is simply the flow on the corresponding arc in $\mathcal{G}_W(\mathbf{x}, \boldsymbol{\delta})$. Also, a feasible flow on $\mathcal{G}_W(\mathbf{x}, \boldsymbol{\delta})$ can be constructed using a feasible solution $\mathbf{x}$ that can be recovered using recovery actions $\mathbf{w}$ after perturbation $\boldsymbol{\delta}$. For $a \in \mathcal{A}_R$, the flow is $x(a) + \delta(a) + w(a)$; and for $a \in \mathcal{A}_I$, the flow is $w(a)$. ∎

Proof (Proposition 2): See Erera et al. (2009), Proposition 2.

Proof (Theorem 1): Let $\mathbf{x}$ be a feasible solution to **NP** and $\boldsymbol{\delta} \in \varphi_k$ be a perturbation vector. Then, by definition on $\mathbf{I}$,

$$\sum_{v \in U} I(v) = \sum_{a \in \Delta^{out}(U) \cup \mathcal{A}_I} x(a) - \sum_{\substack{i \in U \\ j \in \mathcal{N} \setminus U}} \delta(i,j) + \sum_{\substack{i \in \mathcal{N} \setminus U \\ j \in U}} \delta(i,j)$$

for every $U \in \mathcal{U}_W$. Thus, by Propositions 1 and 2, $\mathbf{x}$ is feasible for **RP** if and only if

$$\sum_{v \in U} I(v) = \sum_{a \in \Delta^{out}(U) \cup \mathcal{A}_I} x(a) - \sum_{\substack{i \in U \\ j \in \mathcal{N} \setminus U}} \delta(i,j) + \sum_{\substack{i \in \mathcal{N} \setminus U \\ j \in U}} \delta(i,j) \geq 0 \qquad \forall U \in \mathcal{U}_W \tag{9}$$

holds for each $\boldsymbol{\delta}$. Since $\boldsymbol{\delta} \in \varphi_k$, $\sum_{\substack{i \in U \\ j \in \mathcal{N} \setminus U}} \delta(i,j) - \sum_{\substack{i \in \mathcal{N} \setminus U \\ j \in U}} \delta(i,j)$ can be bounded:

$$\sum_{\substack{i \in U \\ j \in \mathcal{N} \setminus U}} \delta(i,j) - \sum_{\substack{i \in \mathcal{N} \setminus U \\ j \in U}} \delta(i,j) \leq \sum_{\substack{i \in U \\ j \in \mathcal{N} \setminus U}} kl^+(i,j) - \sum_{\substack{i \in \mathcal{N} \setminus U \\ j \in U}} -kl^-(i,j) = \vartheta(U,k)$$

Note that this bound is tight for at least one $\boldsymbol{\delta} \in \varphi_k$ (namely, $\delta(a) = l(a) + kl^+(a)$ for each $a \in \Delta^{out}(U)$, and $\delta(a) = l(a) - kl^-(a)$ for each $a \in \Delta^{in}(U)$).

Thus, condition (9) simplifies to

$$\sum_{a \in \Delta^{out}(U) \cup \mathcal{A}_I} x(a) \geq \vartheta(U,k) \qquad \forall U \in \mathcal{U}_W. \qquad \blacksquare$$

Appendix B Detailed Computational Results

Table 1: Data associated with Figure 1a

Robustness Characteristics	Service Level	Transportation Cost	Solution Time (s)
Non-Robust	97.493%	1,252,846.4	261.294
Robust; $(R, N) = (2, 2)$	98.078%	1,260,953.9	248.504
Robust; $(R, N) = (4, 2)$	98.281%	1,262,446.1	250.722
Robust; $(R, N) = (6, 2)$	99.112%	1,271,464.1	253.241
Robust; $(R, N) = (8, 2)$	99.226%	1,273,461.1	252.775
Robust; $(R, N) = (10, 2)$	99.364%	1,276,436.6	253.428
Robust; $(R, N) = (16, 2)$	99.426%	1,278,341.0	256.996
Robust; $(R, N) = (20, 2)$	99.397%	1,277,732.0	263.060

Table 2: Data associated with Figure 1b

Robustness Characteristics	Service Level	Transportation Cost	Solution Time (s)
Non-Robust	97.342%	2,516,557.9	5,374.510
Robust; $(R, N) = (2, 2)$	98.905%	2,546,511.5	5,244.896
Robust; $(R, N) = (6, 2)$	99.359%	2,555,547.3	4,282.854
Robust; $(R, N) = (10, 2)$	99.403%	2,571,598.7	4,492.900

Table 3: Data associated with Figure 1c

Robustness Characteristics	Service Level	Transportation Cost	Solution Time (s)
Non-Robust	96.638%	3,404,887.9	9,824.288
Robust; $(R, N) = (2, 2)$	98.645%	3,440,173.0	10,241.292
Robust; $(R, N) = (4, 2)$	98.846%	3,445,266.4	10,369.630
Robust; $(R, N) = (6, 2)$	99.249%	3,460,009.5	10,385.561

Table 4: Data associated with Figure 2

Robustness Characteristics	Service Level	Transportation Cost	Solution Time (s)
Non-Robust	97.493%	1,252,846.4	261.294
Robust; $(R, N) = (6, 2)$	99.112%	1,271,464.1	253.241
Robust; $(R, N) = (6, 3)$	99.112%	1,271,560.5	313.371
Robust; $(R, N) = (6, 4)$	99.138%	1,271,845.5	6,595.112
Robust; $(R, N) = (6, 5)$	99.166%	1,272,501.4	176,915.330

Table 5: Data associated with Figure 3a

Robustness Characteristics	k	Service Level	Transportation Cost	Solution Time (s)
Non-Robust	N/A	97.493%	1,252,846.4	261.294
Robust; $(R, N) = (6, 3)$	1	99.805%	1,295,388.6	317.826
Robust; $(R, N) = (8, 3)$	1	99.836%	1,300,076.7	410.635
Robust; $(R, N) = (10, 3)$	1	99.935%	1,314,678.7	624.784
Robust; $(R, N) = (6, 3)$	1/2	99.524%	1,278,676.7	311.738
Robust; $(R, N) = (8, 3)$	1/2	99.588%	1,280,963.3	406.185
Robust; $(R, N) = (10, 3)$	1/2	99.74%	1,286,487.1	608.169
Robust; $(R, N) = (6, 3)$	1/3	99.112%	1,271,560.5	313.371
Robust; $(R, N) = (8, 3)$	1/3	99.231%	1,273,698.2	403.862
Robust; $(R, N) = (10, 3)$	1/3	99.371%	1,276,555.2	617.180
Robust; $(R, N) = (6, 3)$	1/4	98.8%	1,267,420.2	310.658
Robust; $(R, N) = (8, 3)$	1/4	98.892%	1,269,179.9	405.012
Robust; $(R, N) = (10, 3)$	1/4	99.045%	1,271,038.8	615.430

Table 6: Data associated with Figure 3b

Robustness Characteristics	k	Service Level	Transportation Cost
Non-Robust	N/A	96.638%	3,404,887.9
Robust; $(R, N) = (2, 2)$	1/2	98.683%	3,442,661.4
Robust; $(R, N) = (4, 2)$	1/2	98.912%	3,448,070.3
Robust; $(R, N) = (6, 2)$	1/2	99.762%	3,488,445.0
Robust; $(R, N) = (2, 2)$	1/3	98.645%	3,440,173.0
Robust; $(R, N) = (4, 2)$	1/3	98.846%	3,445,266.4
Robust; $(R, N) = (6, 2)$	1/3	99.249%	3,460,009.5
Robust; $(R, N) = (2, 2)$	1/4	98.283%	3,430,121.1
Robust; $(R, N) = (4, 2)$	1/4	98.5%	3,441,054.5
Robust; $(R, N) = (6, 2)$	1/4	98.871%	3,449,934.1

Table 7: Data associated with Figure 5

Empty Network Structure	Robustness Characteristics	Service Level	Transportation Cost
5 Sharing groups	Non-Robust	97.493%	1,252,846.4
	Robust; $(R, N) = (2, 2)$	98.078%	1,260,953.9
	Robust; $(R, N) = (4, 2)$	98.281%	1,262,446.1
	Robust; $(R, N) = (6, 2)$	99.112%	1,271,464.1
	Robust; $(R, N) = (8, 2)$	99.226%	1,273,461.1
	Robust; $(R, N) = (10, 2)$	99.364%	1,276,436.6
5 Sharing groups - balanced	Non-Robust	97.811%	1,184,865.1
	Robust; $(R, N) = (2, 2)$	98.768%	1,195,691.8
	Robust; $(R, N) = (4, 2)$	98.892%	1,195,410.0
	Robust; $(R, N) = (6, 2)$	99.012%	1,197,751.9
	Robust; $(R, N) = (8, 2)$	99.475%	1,208,772.9
	Robust; $(R, N) = (10, 2)$	99.531%	1,209,949.8

Table 8: Data associated with Figure 6

Empty Network Structure	Robustness Characteristics	Service Level	Transportation Cost
3 Sharing groups - balanced	Non-Robust	97.39%	1,232,041.1
	Robust; $(R, N) = (4, 2)$	97.724%	1,223,992.0
	Robust; $(R, N) = (6, 2)$	98.671%	1,249,781.1
	Robust; $(R, N) = (8, 2)$	99.082%	1,260,972.9
	Robust; $(R, N) = (10, 2)$	99.222%	1,267,246.9
5 Sharing groups - balanced	Non-Robust	97.811%	1,184,865.1
	Robust; $(R, N) = (4, 2)$	98.892%	1,195,410.0
	Robust; $(R, N) = (6, 2)$	99.012%	1,197,751.9
	Robust; $(R, N) = (8, 2)$	99.475%	1,208,772.9
	Robust; $(R, N) = (10, 2)$	99.531%	1,209,949.8
7 Sharing groups - balanced	Non-Robust	97.629%	1,189,464.5
	Robust; $(R, N) = (4, 2)$	98.287%	1,194,920.6
	Robust; $(R, N) = (6, 2)$	99.048%	1,207,463.9
	Robust; $(R, N) = (8, 2)$	99.422%	1,216,283.2
	Robust; $(R, N) = (10, 2)$	99.454%	1,219,565.2

References

Atamtürk, A. and Zhang, M. (2007). Two-stage robust network flow and design under demand uncertainty. *Operations Research*, 55(4):662–673.

Ben-Tal, A., Golany, B., Nemirovski, A., and Vial, J.-P. (2005). Retailer-supplier flexible commitments contracts: A robust optimization approach. *Manufacturing Service Operations Management*, 7(3):248–271.

Ben-Tal, A., Goryashko, A., Guslitzer, E., and Nemirovski, A. (2004). Adjustable robust solutions of uncertain linear programs. *Mathematical Programming*, 99(2):351–376.

Ben-Tal, A. and Nemirovski, A. (1998). Robust convex optimization. *Mathematics of Operations Research*, 23(4):769–805.

Ben-Tal, A. and Nemirovski, A. (2000). Robust solutions of linear programming problems contaminated with uncertain data. *Mathematical Programming*, 88(3):411–424.

Bertsimas, D. and Sim, M. (2003). Robust discrete optimization and network flows. *Mathematical Programming*, 98(1-3):49–71.

Bertsimas, D. and Sim, M. (2004). The price of robustness. *Operations research*, 52(1):35–53.

Bertsimas, D. and Thiele, A. (2006). A robust optimization approach to inventory theory. *Operations Research*, 54(1):150–168.

Chen, Y. and Iyengar, G. (2012). A new robust cycle-based inventory control policy.

Cheung, R. and Chen, C. (1998). A two-stage stochastic network model and solution methods for the dynamic empty container allocation problem. *Transportation science*, 32(2):142–162.

Cheung, R. and Powell, W. (1996). An algorithm for multistage dynamic networks with random arc capacities, with an application to dynamic fleet management. *Operations Research*, 44(6):951–963.

Chung, B. D., Yao, T., Xie, C., and Thorsen, A. (2010). Robust optimization model for a dynamic network design problem under demand uncertainty. *Networks and Spatial Economics*, 11(2):371–389.

Crainic, T., Gendreau, M., and Dejax, P. (1993). Dynamic stochastic models for the allocation of empty containers. *Operations Research*, 41(1):102–126.

Di Francesco, M., Crainic, T. G., and Zuddas, P. (2009). The effect of multi-scenario policies on empty container repositioning. *Transportation Research Part E: Logistics and Transportation Review*, 45(5):758–770.

Di Francesco, M., Lai, M., and Zuddas, P. (2013). Maritime repositioning of empty containers under uncertain port disruptions. *Computers & Industrial Engineering*, 64:827–837.

Du, Y. and Hall, R. (1997). Fleet sizing and empty equipment redistribution for center-terminal transportation networks. *Management Science*, 43(2):145–157.

Erera, A., Morales, J., and Savelsbergh, M. (2009). Robust optimization for empty repositioning problems. *Operations Research*, pages 1–32.

Frantzeskakis, L. F. and Powell, W. B. (1990). A successive linear approximation procedure for stochastic, dynamic vehicle allocation problems. *Transportation Science*, 24(1):40–57.

Gabrel, V., Lacroix, M., Murat, C., and Remli, N. (2014). Robust location transportation problems under uncertain demands. *Discrete Applied Mathematics*, 164:100–111.

Godfrey, G. A. and Powell, W. B. (2002a). An adaptive dynamic programming algorithm for dynamic fleet management, I: Single period travel times. *Transportation Science*, 36(1):21–39.

Godfrey, G. A. and Powell, W. B. (2002b). An adaptive dynamic programming algorithm for dynamic fleet management, II: Multi-period travel times. *Transportation Science*, 36(1):40–54.

Hall, R. W. and Zhong, H. (2002). Decentralized inventory control policies for equipment management in a many-to-many network. *Transportation Research Part A: Policy and Practice*, 36(10):849–865.

Kouvelis, P. and Yu, G. (1997). *Robust Discrete Optimization and Its Applications*. Kluwer Academic, Dordrecht, Germany.

Laporte, G., Marín, A., Mesa, J. A., and Perea, F. (2011). Designing robust rapid transit networks with alternative routes. *Journal of Advanced Transportation*, 45:54–65.

Leddon, C. and Wrathall, E. (1967). Scheduling empty freight car fleets on the Louisville and Nashville railroad. In *Second International Symposium on the Use of Cybernetics on the Railways*, pages 1–6, Montreal, Canada.

Lee, S. and Moon, I. (2020). Robust empty container repositioning considering foldable containers. *European Journal of Operational Research*, 280:909–925.

Long, Y., Chew, E. P., and Lee, L. H. (2015). Sample average approximation under non-i.i.d. sampling for stochastic empty container repositioning problem. *OR Spectrum*, 37:389–405.

Long, Y., Lee, L. H., and Chew, E. P. (2012). The sample average approximation method for empty container repositioning with uncertainties. *European Journal of Operational Research*, 222(1):65–75.

Misra, S. (1972). Linear programming of empty wagon disposition. *Rail International*.

Powell, W. (1986). A stochastic model of the dynamic vehicle allocation problem. *Transportation Science*, 20(2):117–129.

Powell, W. (1987). An operational planning model for the dynamic vehicle allocation problem with uncertain demands. *Transportation Research Part B: Methodological*, 21(3):217–232.

Powell, W. and Carvalho, T. (1998). Dynamic control of logistics queueing network for large-scale fleet management. *Transportation Science*, 32(2):90–109.

See, C.-T. and Sim, M. (2010). Robust approximation to multiperiod inventory management. *Operations Research*, 58(3):583–594.

Shu, J. and Song, M. (2014). Dynamic container deployment: Two-stage robust model, complexity, and computational results. *INFORMS Journal on Computing*, 26(1):135–149.

Solyalı, O., Cordeau, J.-F., and Laporte, G. (2012). Robust inventory routing under demand uncertainty. *Transportation Science*, 46(3):327–340.

Solyalı, O., Cordeau, J.-F., and Laporte, G. (2016). The impact of modeling on robust inventory management under demand uncertainty. *Management Science*, 62(4):1188–1201.

Soyster, A. (1973). Convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations research*, 21(5):1154–1157.

von Westarp, A. G. and Schinas, O. (2016). A fuzzy approach for container positioning considering sustainable profit optimization. *Transportation Research Part E: Logistics and Transportation Review*, 92:56–66.

White, W. (1972). Dynamic transshipment networks: An algorithm and its application to the distribution of empty containers. *Networks*.

Zhang, G., Smilowitz, K., and Erera, A. (2011). Dynamic planning for urban drayage operations. *Transportation Research Part E: Logistics and Transportation Review*, 47(5):764–777.