Robust Optimization for Empty Repositioning Problems

Alan L. Erera, Juan C. Morales and Martin Savelsbergh

The Supply Chain and Logistics Institute
School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, GA 30332-0205

September 28, 2007

Abstract

We develop a robust optimization framework for dynamic empty repositioning problems modeled using time-space networks. In such problems, uncertainty arises primarily from forecasts of future supplies and demands for assets at different time epochs. The proposed approach models such uncertainty using intervals about nominal forecast values and a limit on the system-wide scaled deviation from the nominal forecast values. A robust repositioning plan is defined as one in which the typical flow balance constraints and flow bounds are satisfied for the nominal forecast values, and the plan is *recoverable* under a limited set of recovery actions. A plan is recoverable when feasibility can be reestablished for *any* outcome in a defined uncertainty set. We develop necessary and sufficient conditions for flows to be robust under this definition for three types of allowable recovery actions. When recovery actions allow only flow changes on inventory arcs, we show that the resulting problem is polynomially solvable. When recovery actions allow limited reactive repositioning flows, we develop feasibility conditions that are independent of the size of the uncertainty set. A computational study establishes the practical viability of the proposed framework.

1 Introduction

Consider a problem faced by most large freight transportation service providers: cost-effective management of empty resources over time. Transporters typically serve load requests that are imbalanced in both time and space. Thus, when a resource such as a container arrives at the destination of a loaded move, there may not be a timely opportunity to match that resource with a new outbound loaded move. To correct imbalance, transporters move resources empty;

planning and executing empty moves that enable future customer demands to be served at low cost is a primary challenge.

Currently, more sophisticated providers address this problem using deterministic flow optimization models over time-space networks. Network nodes are defined at relevant decision points, and connect forward in time with other nodes via arcs that represent management decisions (and their costs) such as holding inventory of empty resources, or repositioning such resources between locations. Next, point forecasts are developed for the expected net supply of resources at some or all of the time-space network nodes, and initial and final resource states are specified. A feasible flow on such a network represents a set of feasible empty management decisions, and network optimization algorithms can be used to find an optimal flow. For problems that can be decomposed by resource, single-commodity minimum cost network flow problems often result, which can be solved very efficiently. In practice, these models are deployed in rolling horizon implementations where a plan is determined for a long planning horizon, but only the decisions in an initial set of time periods are implemented.

A major deficiency of this approach is that there may be significant uncertainty in resource net supply at each time-space node, especially towards the end of the planning horizon. When realized net supply values differ from the point forecasts, repositioning plans created by deterministic models may be highly suboptimal. Importantly, it is likely that some future resource demands will be impossible to satisfy feasibly.

To address this deficiency, this paper proposes a robust optimization approach for repositioning problems, focusing on developing solutions that are immunized against uncertainty in the sense that feasible recovery plans will always exist for any realization in a defined uncertainty set. The approach borrows ideas from both Ben-Tal et al. (2004) and Bertsimas and Sim (2003). Since many future decisions in a repositioning plan can be changed before their execution time, we develop a two-stage planning framework similar to the adjustable robust counterpart (ARC) for linear programming problems (see Ben-Tal et al., 2004). We model forecast uncertainty without probability distributions using intervals about a nominal net supply value at each time-space node, and limit system-wide deviations from nominal values using an uncertainty budget (see Bertsimas and Sim, 2003).

Our robust 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 can simultaneously take their worst-case value. A plan is recoverable if there exists a set of recovery actions (similar to recourse actions in two-stage stochastic programming models) that can transform the plan such that it satisfies flow bounds and balance equalities given the realized net supply values. In this research, we limit the recovery actions to be some predefined subset of the original planning decisions, where the subsets considered include only low-cost decisions. Parameter k can be interpreted as an manager's view on the accuracy of the nominal net supply values: when $k = 0$, the manager has complete confidence in the point forecasts and the problem reduces to the deterministic problem, and when $k = \infty$, the manager has no confidence in their accuracy and solutions must be recoverable for all realizations.

The simplest problem considered herein restricts recovery actions to modify resource flows only on inventory arcs between the same space point in consecutive time periods; this scenario, therefore, corresponds to the case where each location must hedge independently against future uncertainty. We show that this robust problem is polynomially solvable. We also consider recovery actions that allow limited reactive repositioning between locations. For these cases, we develop sets of feasibility conditions whose sizes, while not polynomial, do not grow with the size of the uncertain outcome set. We illustrate the framework using a set of computational experiments based on a representative global container repositioning problem of realistic size. The results provide insight into how different levels of confidence in the nominal net supply values, measured by parameter k, and different degrees of flexibility for performing recovery actions affect the cost of a robust repositioning plan, and therefore the price of robustness.

The main contributions are now summarized:

- This paper develops a robust optimization framework for integer programming problems with equality constraints and right-hand side uncertainty, a common feature of many logistics planning problems. The framework explicitly incorporates the notion of recovery actions to dynamically respond to realizations of the uncertain parameters, transforming initial planned solutions into feasible solutions.

- This paper provides implementations of the proposed robust optimization framework in the context of empty repositioning problems faced by many freight transportation service providers for different sets of allowable recovery actions.

- This paper shows that for the sets of recovery actions considered for the empty repositioning problem, the size of the resulting optimization problems do not depend on the size of the uncertain outcome set, and that for the simplest set of recovery actions the resulting optimization problem can be solved in polynomial time.

- This paper presents a set of computational experiments illustrating the value of the approach in the context of empty repositioning problems and demonstrating the computational viability of the proposed framework.

The remainder of the paper is organized as follows. Section 2 discusses related literature. Section 3 introduces the robust optimization framework. Section 4 applies the robust optimization framework in the context of empty repositioning problems. Finally, Section 5 describes some practical considerations for the use of robust repositioning models, and explores the potential benefits of the approach in the context of the global management of containers.

2 Related Literature

Empty repositioning problems have received much attention by the research community, and the existing literature for problems of this type is extensive and varied. For operational decision-making, dynamic models are most frequently developed and deployed (see Powell, 2003, for an excellent review of dynamic models for transportation operations management). Early important studies of problems in this class assume that decision-makers have complete information for future periods, and deterministic models are developed. Leddon and Wrathall (1967) and Misra (1972) develop models for railcar distribution, and White (1972) for container management. Recent research in this area focuses on solving large-scale problems with time windows, and integrating repositioning decisions with load assignment decisions (see, *e.g.*, Abrache et al., 1999, and Erera et al., 2005). Deterministic models of dynamic repositioning decisions typically result in linear programs or easy-to-solve integer programs. As input, they require only point forecasts of future resource supplies and demands. For these reasons, they remain popular in application with many freight transportation companies.

Approaches that explicitly model uncertainty in this application domain focus primarily on expected cost minimization, beginning with work in Powell (1986) and Powell (1987). Modelling approaches for stochastic empty container management problems are provided in Crainic et al. (1993). Most computational research in this area uses dynamic programming models combined with suboptimal solution procedures, and effective approaches have been proposed for sequentially approximating value functions for multi-stage problems (see, *e.g.*, Frantzeskakis and Powell, 1990, and Cheung and Powell, 1996). More recently, adaptive approaches to approximating nonlinear value functions have been successfully applied to both single commodity and multicommodity problems (see, *e.g.*, Godfrey and Powell, 2002a, Godfrey and Powell, 2002b, and Topaloglu and Powell, 2004).

The model we develop in this paper is similar to two-stage stochastic integer programming models. In such models, initial (first-stage) decisions are planned prior to the realization of uncertain parameters, while second-stage decisions provide recourse opportunities given a realization (for an introduction, see *e.g.*, Birge and Louveaux, 1999). A popular solution approach which has proven quite useful for problems with right-hand side uncertainty is the sample average approximation method (see Kleywegt et al., 2001 and Ahmed and Shapiro, 2002). This approach is practical only if the approximating problem is solvable (usually via a decomposition approach) for a reasonable number of scenarios, which is complicated by integrality restrictions.

Robust optimization is an alternative approach for modeling and solving decision optimization problems given uncertainty. Soyster (1973) is the first work to consider coefficient uncertainty in linear programming formulations, and shows that such uncertainty can be handled by an equivalent linear programming model. The approach, however, is very conservative since it protects feasibility against all potential realizations. More recently, Ben-Tal and Nemirovski (1998) develops a general framework for robust optimization over a convex cone, and Ben-Tal and Nemirovski (2000) specifically considers linear programs with coefficient uncertainty. To control conservatism, these references propose the use of an ellipsoidal uncertainty set that ignores unlikely joint realizations; for linear programming, the resultant robust optimization problem requires solution of a convex optimization problem over a second-order cone. To avoid the necessity of a nonlinear optimization method, Bertsimas and Sim (2004) considers robust linear programming with coefficient uncertainty using a uncertainty set with budgets. In this model, each coefficient is assumed to take a value in a symmetric interval around a nominal value, and a budget parameter for each constraint limits the number of coefficients that can simultaneously take their worst-case value; the resulting robust optimization remains a linear program. Extending this work, Bertsimas et al. (2004) develops robust linear programming problems with coefficient uncertainty sets described by an arbitrary norm.

Robust versions of discrete optimization problems have also received attention. Kouvelis and Yu (1997) develops robust versions of many traditional discrete optimization problems with two different objective functions, one which minimizes the maximum absolute cost under any potential outcome, and another which minimizes the maximum cost difference (or regret) between the robust solution and a reoptimized solution for each outcome. The reference shows that robust forms of many polynomial discrete optimization problems become NP-hard. Bertsimas and Sim (2003) considers robust discrete optimization and network flow problems where parameter uncertainty occurs only in the objective function, and shows that many polynomially solvable (or polynomially approximatable) problems remain so in this case.

Closely related to our work, Atamturk and Zhang (2006) studies network flow and design problems with right-hand side uncertainty using a robust approach. Similar to our approach, this paper develops and analyzes a two-stage approach similar in spirit to the adjustable robust counterpart approach for linear programming developed in Ben-Tal et al. (2004). Also related to our research, Bertsimas and Thiele(2004 and 2006) use a robust optimization framework to address traditional inventory control problems for single point and tree-network supply chain networks that are typically addressed with dynamic programming techniques.

3 Robust Feasibility-Recovery Optimization Framework for Problems with Right-hand-side Uncertainty

Before focusing specifically on empty repositioning flow problems, we first outline a general robust approach for optimization problems with right-hand side uncertainty that considers feasibility recovery actions. The approach applies the adjustable robust counterpart approach developed for uncertain linear programs in Ben-Tal et al. (2004) to mixed-integer programming problems in a specific way useful for rolling-horizon models where ensuring the existence of future feasible solutions is important.

Consider first a nominal optimization problem:

$$\textbf{NP} \quad \min_x \left\{ c^T x : Ax = b,\ x \in X \right\} \tag{1}$$

where c is an n-vector, A is an m by n matrix, b is a deterministic m-vector referred to as the *vector of nominal right-hand side values*, and x is an n-vector of decision variables. Set X is used to describe any additional constraints on the components of x, such as lower and upper bounds as well as integrality.

Now suppose that b may be uncertain, and further that each potential realization is a realization of some random vector $\widetilde{b}$. Given a distribution for $\widetilde{b}$, stochastic approaches for extending **NP** include both chance-constrained programming as well as two-stage stochastic programming with recourse. An alternative is to use a robust approach, planning for worst-case realizations of $\widetilde{b}$. Since it will usually be too conservative to seek solutions that are robust with respect to all potential realizations within the finite support of the distribution of $\widetilde{b}$, one may alternatively consider only realizations in a smaller, user-defined *uncertainty set* $\mathcal{Z} \subset \mathbb{R}^m$. Note that we assume that $b \in \mathcal{Z}$.

In typical robust optimization problems, one searches for a solution x that remains feasible given any uncertain outcome in $\mathcal{Z}$. However, any feasible solution x to **NP** is infeasible given any non-zero perturbation $\delta \in \mathbb{R}^m$ such that $b + \delta \in \mathcal{Z}$. An alternative approach is to search for a solution x that can be made feasible using *adjustable* variables w (hereafter referred to as recovery actions), following the approach developed in Ben-Tal et al. (2004) for linear programs.

Consider then the following adjustable robust optimization problem:

$$\textbf{ARP} \quad \min_x \left\{ c^T x : Ax = b,\ x \in X,\ \forall\, b + \delta \in \mathcal{Z}\ \ \exists\, w \in W\ :\ Ax + Bw = b + \delta \right\} \tag{2}$$

where B is an m by p matrix and w is a p-vector of recovery action variables that may differ for each realization $b + \delta \in \mathcal{Z}$. Set W does not depend on δ, and represents a set of *recovery constraints* that may place additional restrictions on the recovery actions and should be developed in the context of the application problem.

Formulation **ARP** is a special case of the ARC modeling framework developed in Ben-Tal et al. (2004), but generalized to allow integer variable restrictions through X. It is a special case since **ARP** allows only uncertainty in the right-hand side vector, and additionally limits x to solutions that are feasible with respect to the nominal vector b. Note that **ARP** does not include in its objective the potential costs of the recovery actions w, but rather only ensures that a feasible adjustment of x exists for all realizations. However, by carefully choosing which decisions to make fixed versus adjustable and by judiciously restricting the recovery actions using the constraints W, it is possible in application to appropriately capture the dominant system costs in the objective function $c^T x$ while ignoring the smaller costs of the recovery actions.

In this spirit, it may be useful in many dynamic applications of such a framework to link a recovery action variable w_a to each fixed variable x_a, and to allow no others. If we treat $x + w$ as a transformation of the initial decisions x, we can set $B = A$ and define a *transformable* robust optimization problem as

$$\textbf{TRP} \quad \min_x \left\{ c^T x : Ax = b,\ x \in X,\ \forall\, b + \delta \in \mathcal{Z}\ \ \exists\, w \in W\ :\ Aw = \delta,\ x + w \in X \right\}. \tag{3}$$

Note that in **TRP**, we explicitly provide an additional linkage between the initial decisions and the recovery actions by forcing $x + w \in X$, which may be used to model bounds that apply to both initial and transformed decisions. Again, while it may seem that such a transformable formulation allows the model to change any fixed decision x_a, the user can use W to ensure that only low-cost adjustments to the initial plan are allowed. While it is natural, for example, to force $w_a = 0$ for any decision x_a that is truly fixed when planning, it may also be useful to do so for any high-cost decisions.

To simplify notation for the remainder of the paper, if we define set $H(W, \delta)$ as

$$H(W, \delta) = \{x \mid\ \exists\, w \in W\ :\ Aw = \delta,\ x + w \in X\},$$

then we can write **TRP** as

$$\textbf{TRP}(W, \varphi) \quad \min_x \left\{ c^T x : Ax = b,\ x \in X,\ x \in \bigcap_{\delta \in \varphi} H(W, \delta) \right\}, \tag{4}$$

where $\varphi = \{\delta :\ b + \delta \in \mathcal{Z}\}$. We can write (4) in an extended form as

$$\begin{aligned}
\textbf{TRP}(W, \varphi) \qquad \text{minimize} \qquad & c^T x \\
\text{s.t.} \qquad & Ax = b \\
& x \in X \\
& Aw_\delta = \delta \quad \forall\, \delta \in \varphi \\
& x + w_\delta \in X \quad \forall\, \delta \in \varphi \\
& w_\delta \in W \quad \forall\, \delta \in \varphi
\end{aligned}$$

where w_δ is a recovery vector applied to x given uncertain outcome δ. Observe that the number of decision variables and the number of constraints in the extended formulation of (4) may dramatically increase from (1) when the size of the outcome set φ is large.

4 Robust Empty Repositioning

Consider a transport operator managing a homogeneous fleet of reusable resources using a centralized control. Examples of such resources are containers, railroad cars or trucks. To serve customer requests for loaded moves, the operator assigns an empty resource from a nearby storage depot. When a loaded move is completed, the receiving customer returns the empty resource to a nearby depot. Since the supply and demand of empties at each depot are not balanced over time, it is usually necessary for operators to reposition resources among depots to satisfy loaded move requirements.

The traditional approach for generating a cost-effective empty repositioning plan is to use a deterministic time-expanded network flow formulation over some planning horizon. For example, one large global tank container operator uses such an approach with a six-month planning horizon discretized into weeks (Stolt-Nielsen Transportation Group, 2004). This company operates a set of storage depots, where each depot serves all customer requests arising within a geographic region surrounding the depot. At the beginning of the planning horizon, each depot may have on-hand empty container inventory, as well as inbound in-transit containers to be returned from customers during later weeks. Using this data along with point forecasts of weekly customer-to-customer loaded container demand, the company estimates point forecasts of the expected net supply of containers at each depot during each week of the six-month planning period; negative net supply corresponds to demand for empties. Note that each individual customer forecast generates a demand for empties at some depot for some week, and an associated supply of empties at another depot at a later week when these resources are returned. Feasible depot-to-depot repositioning options and their costs are determined from negotiated rates with ocean carriers, and required repositioning time is derived from the associated vessel schedules. The resultant minimum cost flow model is solved for the entire planning horizon, and the decisions for the first week are implemented. The planning process is repeated weekly.

4.1 The Nominal Repositioning Problem

This deterministic flow model can be considered the nominal repositioning problem. To formalize, let $G = (\mathcal{N}, \mathcal{A})$ be the time-expanded network described above. Assume that the planning horizon includes $\rho + 1$ discrete periods, $\{0, 1, 2, ..., \rho\}$. Let D be the set of depots, and let $V_\tau^d = \{v_0^d, v_1^d, ..., v_\tau^d\}$ for each $d \in D$ be the ordered set of nodes v_t^d representing depot d at time t from time $t = 0$ to time $t = \tau$. Let $V^d = V_\rho^d$, the complete node set for depot d, and $\mathcal{V} = \cup_{d \in D} V^d$. Finally, let b be the vector of net supply forecasts, and let $b(v)$ represent the component of b corresponding to $v \in \mathcal{V}$.

Containers can be held in inventory at a depot from one time period to the next. Therefore, an *inventory arc* (v_t^d, v_{t+1}^d) exists for each $d \in D$ and $0 \leq t < \rho$. Let I be the set of all inventory arcs. A *repositioning arc* (v_t^i, v_{t+h}^j) is defined between depots i and j at time t when available, where $h \geq 1$ is the travel time in periods along this arc. Let R be the set of all repositioning arcs. To complete the network specification, we add to G a sink node s with net supply $b(s) = -\sum_{v_t^d \in V} b(v_t^d)$ and an arc connecting v_ρ^d to s for all $d \in D$. For consistency we add these arcs to set I; for simplicity, node s can be labeled as $v_{\rho+1}^d$ for any $d \in D$. Let $\mathcal{N} = \mathcal{V} \cup \{s\}$ and $\mathcal{A} = I \cup R$.

For each $a \in \mathcal{A}$, let $c(a)$ be the unit cost of flow on arc a. For $a \in R$, $c(a)$ represents the repositioning costs per unit transported, while for $a \in I$, $c(a)$ represents per period holding costs per unit. In virtually all freight transportation settings, the (actual) cost of holding a resource in inventory is much smaller than the cost of moving a resource. Furthermore, since differences in per unit holding costs at different depots are also minor, in many applications holding costs are assumed to be negligible.

The nominal repositioning problem may now be written as:

$$\textbf{NP} \quad \min_x \left\{ c^T x : Ax = b, x \in \mathbb{Z}_+^{|\mathcal{A}|} \right\} \tag{5}$$

where the decision vector x corresponds to the empty container flow on each arc and A is the node-arc incidence matrix implied by G defining the typical network flow-balance constraints $Ax = b$. Let $x(a)$ represent the flow on arc $a \in A$. Note that this formulation is equivalent to (1) where $X = \mathbb{Z}_+^{|\mathcal{A}|}$; we will use this definition of X for the remainder of the paper.

It is well-known that problem (5) can be solved to optimality in polynomial time with standard minimum cost network flow algorithms, or via linear programming. We note that a feasible solution may not exist for (5); well-known techniques can address this problem, but for clarity and simplicity we assume that a feasible solution exists.

4.2 Three Robust Repositioning Problems

Since the nominal net supply $b(v)$ at each node $v \in \mathcal{V}$ may be uncertain, we now apply the robust framework $TRP(W, \varphi)$ to the repositioning problem. Let $\widetilde{b}(v) \in \mathbb{Z}$ be the realized net supply at $v \in \mathcal{V}$. Furthermore, suppose that the decision-maker estimates for each v an interval around $b(v)$ containing all potential realizations of $\widetilde{b}(v)$ for which future feasibility should be protected. Assuming the interval is symmetric, we represent it as

$$\widetilde{b}(v) \in [b(v) - \hat{b}(v), b(v) + \hat{b}(v)] \quad \forall \;\; v \in \mathcal{V}, \tag{6}$$

where $\hat{b} \geq 0$. We assume that the decision-maker always knows with certainty the net supplies in the initial period, and therefore $\hat{b}(v_0^i) = 0$ for each $i \in D$.

To allow further control of the conservatism of our robust repositioning models, we adopt the approach proposed in Bertsimas and Sim (2003) to restrict the potential *joint* realizations of $\widetilde{b}$ using an uncertainty budget. To do so, we define a limited perturbation set φ_k as a function of a budget parameter k as follows:

$$\varphi_k = \left\{ \delta \in \mathbb{Z}^{|\mathcal{N}|} \; : \; \delta(v) = \hat{b}(v) z(v), \sum_{v \in \mathcal{V}} |z(v)| \leq k, \; |z(v)| \leq 1 \; \forall \, v \in \mathcal{V}, \;\; \delta(s) = -\sum_{v \in \mathcal{V}} \delta(v) \right\}, \tag{7}$$

and we assume that each realization of interest is given by $b + \delta$ for some $\delta \in \varphi_k$. Note that the constraint on $\delta(s)$ is a technical condition used only to preserve balance (for this reason, we will ignore $\delta(s)$ for the majority of the discussion to follow). Parameter k specifies the maximum number of net supplies that may simultaneously take on an extreme value in a realization, and thus can be used to control the conservatism of the uncertainty set used by the model. When $k = 0$, the decision-maker is most aggressive assuming that every realization will conform to nominal. When $k \geq |N|$, the decision-maker protects against all potential realizations that fall within the intervals specified by (6).

Since the robust framework presented in Section 3 proposes to determine a feasible solution to the nominal repositioning problem (5) that is recoverable for a set of realizations, a concept important for the analysis to follow is the *vulnerability* of a set of nodes:

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

$$\vartheta(V,k) \ = \ \max_{z} \left\{ \sum_{v \in V} \hat{b}(v)\, z(v) \ : \ \sum_{v \in V} |z(v)| \leq k \ , \ |z(v)| \leq 1 \ \forall \ v \in V \right\}.$$

Observe that $\vartheta(V,k)$ corresponds to the maximum aggregate deviation from the nominal values of nodes in V over all demand realizations in φ_k. The vulnerability of a node set V with m members can be determined easily in polynomial time. Suppose $V = \{v_1, v_2, ..., v_m\}$ where $\hat{b}(v_1) \geq \hat{b}(v_2) \geq ... \geq \hat{b}(v_m)$. Then, $\vartheta(V,k) = \sum_{i=1}^{\min(m,k)} \hat{b}(v_i)$. Further, we denote by $\eta(V,k) = \cup_{i=1}^{\min(m,k)} \{v_i\}$ the set of nodes that realize their worst-case value in the determination of $\vartheta(V,k)$.

We now specify several different robust repositioning problems, each defined by a different recovery set W. For each problem, we develop necessary and sufficient conditions for feasible solutions.

4.2.1 The Inventory Robust Repositioning Problem

Suppose that a decision-maker would like each depot to hedge independently against uncertainty using its own inventory. To model this case, let W_1 be the set of recovery vectors w that allow integer flow changes only on inventory arcs:

$$W_1 = \{w \in \mathbb{Z}^{|\mathcal{A}|} \mid w(a) = 0 \quad \forall \ \ a \in R\}.$$

Flow changes on inventory arcs can be interpreted as using resources in inventory to satisfy a larger-than-expected demand (a negative flow change), or adding extra resources to inventory in the event of a larger-than-expected supply (a positive flow change).

We can now define the inventory robust optimization problem using our earlier notation:

$$\textbf{TRP1} = \textbf{TRP}(W_1, \varphi_k).$$

Any repositioning plan x satisfying the feasibility conditions of **TRP1** is called *k-robust inventory feasible*. **TRP1** seeks the minimum cost k-robust inventory feasible solution, and can be written in extended form as

$$
\begin{aligned}
\textbf{TRP1}(W_1, \varphi_k) \qquad & \text{minimize} \qquad && c^T x \\
& \text{s.t.} && Ax = b && (8)\\
& && x \in \mathbb{Z}_+^{|\mathcal{A}|} && (9)\\
& && Aw_\delta = \delta \quad \forall\, \delta \in \varphi_k && (10)\\
& && x + w_\delta \in \mathbb{Z}_+^{|\mathcal{A}|} \quad \forall\, \delta \in \varphi_k && (11)\\
& && w_\delta(a) = 0 \quad \forall\, a \in R, \quad \delta \in \varphi_k && (12)\\
& && w_\delta \in \mathbb{Z}^{|\mathcal{A}|} \quad \forall\, \delta \in \varphi_k && (13)
\end{aligned}
$$

While correct, the above formulation requires for each potential uncertain outcome δ a vector w_δ of decision variables representing the recovery transformation and an associated set of flow balance constraints. Clearly, such a formulation becomes intractable as the size of the outcome set grows. We now show that **TRP1** alternatively can be solved as a minimum cost network flow problem with flow lower bound constraints using only the original flow variables x.

To do so, consider any inventory arc $a_t^d = (v_t^d, v_{t+1}^d) \in I$ and the corresponding node set V_t^d. Given a specific uncertain outcome $\delta \in \varphi_k$, let $\sigma(a_t^d)$ be the cumulative deviation from nominal net supply at depot d by time t:

$$\sigma(a_t^d) = \sum_{v \in V_t^d} \delta(v).$$

Since the vulnerability $\vartheta(V_t^d, k)$ is the maximum cumulative deviation from the nominal net supply values for the nodes of depot d up to the tail node of arc a in any realization, it is clear then that $|\sigma(a_t^d)| \le \vartheta(V_t^d, k)$ for all $\delta \in \varphi_k$.

The relationship between the flow on an inventory arc $x(a_t^d)$ and the vulnerability of V_t^d will determine whether or not a solution is k-robust inventory feasible. This motivates the following definition.

Definition 2 (Weak Arc) *For a given repositioning plan x, an inventory arc $a_t^d = (v_t^d, v_{t+1}^d) \in I$ is a weak arc if*

$$x(a_t^d) < \vartheta(V_t^d, k).$$

Observe that if a_t^d is a weak arc, then the inventory at time t at depot d is not sufficient to protect against every potential uncertain realization in φ_k.

The following theorem now characterizes the set of feasible solutions for **TRP1**:

Theorem 1 *A feasible solution x for the nominal problem* (5) *is also a feasible solution for **TRP1** if and only if*

$$x(a_t^d) \ge \vartheta(V_t^d, k) \;\; \forall \;\; a_t^d = (v_t^d, v_{t+1}^d) \in I. \tag{14}$$

Proof: Given the definition of W_1, for any $\delta \in \varphi_k$ the only transformation vector w that can feasibly satisfy constraints (10), (12), and (13) in **TRP1** is given by

$$w(a) = \sigma(a) \quad \forall \ a \in I. \tag{15}$$

Thus, we focus attention on constraints (11).

To show necessity by contradiction, suppose that there exists a feasible solution x for **TRP1** such that $x(a_t^d) < \vartheta(V_t^d, k)$ for some arc $a_t^d = (v_t^d, v_{t+1}^d)$. Now consider the uncertain outcome $\delta \in \varphi_k$ such that $\delta(v) = -\hat{b}(v)$ for all $v \in \eta(V_t^d, k)$ and $\delta(v) = 0$ for all other $v \in V$. Thus, from (15) note that $w(a_t^d) = -\vartheta(V_t^d, k)$ and thus the transformed flow on arc a_t^d is $x(a_t^d) - \vartheta(V_t^d, k) < 0$ which violates constraint (11). Therefore, x cannot be a feasible solution for **TRP1**.

Sufficiency can also be shown by contradiction. Let x be a feasible solution of (5) satisfying (14). Now, consider uncertain outcome $\delta \in \varphi_k$ such that after applying transformation (15) to x there exists an arc $a_t^d = (v_t^d, v_{t+1}^d) \in I$ such that $x(a_t^d) + w(a_t^d) < 0$. This implies $\vartheta(V_t^d, k) \leq x(a_t^d) < -\sigma(a_t^d)$, which then implies $\delta \notin \varphi_k$. □

Theorem 1 shows that **TRP1** can be solved for any given φ_k by adding a flow lower bound constraint for each inventory arc $a \in I$ to (5). Thus, **TRP1** is polynomially solvable using standard minimum cost network flow algorithms. Also, observe that the lower bound constraints for a specific depot j are independent of the vulnerability of the arcs for any other depot in the system.

It is also important to note that in order to develop the necessary and sufficient conditions in Theorem 1, we need only consider perturbations $\delta \leq 0$. Any positive component in δ implies that more resources than expected are available at some depot at some time, and such an outcome does not act against the interests of a decision-maker attempting to determine a feasible container allocation. In the remainder of this paper, we only consider perturbation vectors $\delta \leq 0$.

4.2.2 The Inventory-Pooling Robust Repositioning Problem

Suppose that container depots can hedge against uncertainty not only using their own inventory but also using inventory at other depots in the system. We use the term *reactive repositioning* to refer to depot-to-depot container repositioning conducted in response to a perturbation from expected net supplies. For a given value of k, a group of depots may be able to jointly hedge against uncertainty with fewer total container resources.

This idea is illustrated for a simple two-depot system in Figure 1 where $k = 1$. The number inside each node corresponds to the nominal net supply value, the number above each arc corresponds to its flow, and the interval above a node determines the range in which δ can take values. Observe that the conditions of Theorem 1 are satisfied for the inventory arcs of depot A, but not for those of depot B. However, if we could reactively reposition a unit of inventory at time 1 from depot A to depot B, then we could recover feasibility for this problem given any realization in φ_1. Since depots A and B share a single container resource in inventory to collectively hedge against uncertainty, such a solution is called an *inventory pooling 1-robust* solution.

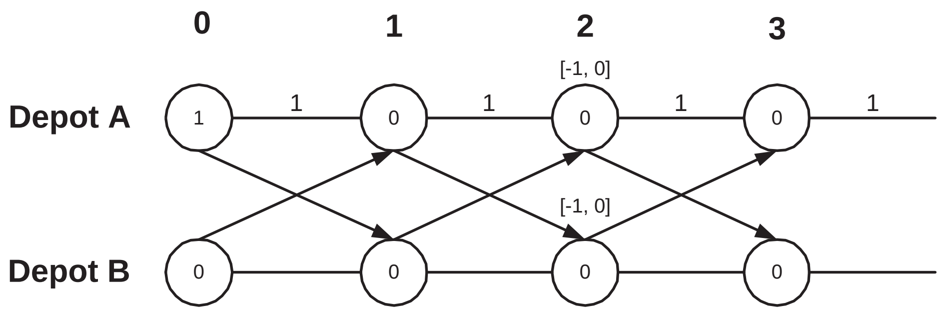

Figure 1: The solution is robust for $k = 1$ if resources in Depot A can be repositioned reactively to Depot B.

We now formally define a recovery set W_2 for the inventory-pooling scenario:

$$W_2 = \{w \in \mathbb{Z}^{|\mathcal{A}|} \mid w(a) \geq 0 \quad \forall \ a \in R, \quad w(a) = 0 \quad \forall \ a = (v_0^j, u) \in R\}.$$

The set W_2 allows any integer flow change on each inventory arc, and non-negative integer flow changes on repositioning arcs. By enforcing non-negativity, we assume that only minor changes are allowed to the initial repositioning plan. Further, we do not allow any reactive flow changes on repositioning arcs that begin in the initial time epoch, since such decisions are assumed to be fixed. If the decision-maker intends to fix decisions for multiple initial time epochs, additional constraints could be added to W_2.

Given W_2, the inventory-pooling robust repositioning problem is:

$$\mathbf{TRP2} = \mathbf{TRP}(W_2, \varphi_k).$$

We can formulate Problem **TRP2** in extended form:

$$\begin{aligned}
\mathbf{TRP2}(W_2, \varphi_k) \ \min \quad & c^T x \\
\text{s.t.} \quad & Ax = b && (16)\\
& x \in \mathbb{Z}_+^{|\mathcal{A}|} && (17)\\
& Aw_\delta = \delta \quad \forall \delta \in \varphi_k && (18)\\
& x + w_\delta \in \mathbb{Z}_+^{|\mathcal{A}|} \quad \forall \delta \in \varphi_k && (19)\\
& w_\delta(a) = 0 \quad \forall a = (v_0^j, u) \in R, \forall \delta \in \varphi_k && (20)\\
& w_\delta(a) \geq 0 \quad \forall a \in R, \forall \delta \in \varphi_k && (21)\\
& w_\delta \in \mathbb{Z}^{|\mathcal{A}|} \quad \forall \delta \in \varphi_k && (22)
\end{aligned}$$

This direct integer programming formulation for **TRP2** may become difficult to solve as the set of realizations defined by φ_k grows large. Therefore, following the approach for the inventory robust repositioning problem, we seek methods for determining an optimal inventory-pooling robust solution that do not rely on enumerating the realization set.

Creating a tight set of necessary and sufficient constraints for a nominal solution x to be robust with respect to any set of allowable recovery actions W requires ensuring that x is in the recoverable set $H(W, \delta)$ for every $\delta \in \varphi_k$. A useful method for testing this condition for the recovery actions is to use the existence conditions for a feasible flow on a properly defined *recovery network*.

Let $G_W = (N_W, A_W)$ refer to the recovery network corresponding to allowable recovery action set W. The node set N_W is the same as the node set $\mathcal{N}$ of G. The arc set A_W contains all inventory arcs in I, and all repositioning arcs in R on which recovery flow is permitted to be nonzero by W. In the case of recovery set W_2, G_{W_2} contains only inventory arcs and each repositioning arc departing a depot at time $t > 0$.

To determine whether a repositioning plan x is recoverable using the action set W_2 for a specific realization $\delta \in \varphi_k$, we add appropriate net supplies to the nodes N_{W_2} and search for a feasible flow on G_{W_2}. To do so, let $\mathcal{I}^{x,\delta}(v)$ at each time-space depot node represent the marginal net inventory of resources available (or needed) in the recovery problem given x and δ:

$$\begin{aligned}
\mathcal{I}^{x,\delta}(v_0^d) &= 0 \quad \text{for all } d \in D \\
\mathcal{I}^{x,\delta}(v_1^d) &= x(v_1^d, v_2^d) + \delta(v_1^d) \quad \text{for all } d \in D \\
\mathcal{I}^{x,\delta}(v_t^d) &= x(v_t^d, v_{t+1}^d) - x(v_{t-1}^d, v_t^d) + \delta(v_t^d) \quad \text{for all } d \in D,\ 1 < t \leq \rho
\end{aligned}$$

where we note that $\delta(v_0^d) = 0$ for all $d \in D$.

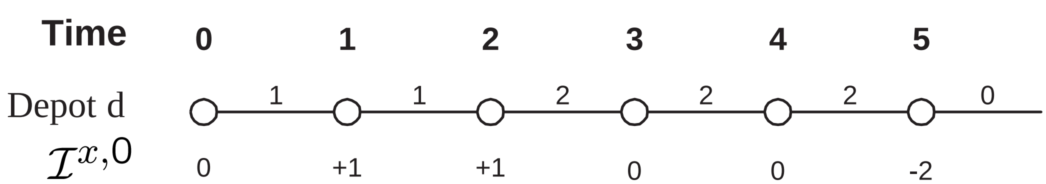

Figure 2: Given the flow on the inventory arcs of depot d, the value of $\mathcal{I}^{x,0}$ indicates that one unit at time 1 is available for reactive repositioning, and that an additional unit becomes available at time 2. The negative value of $\mathcal{I}^{x,0}$ indicates that at least 2 units must be in inventory by time 5. If any units are repositioned out of d, the same number must be repositioned back no later than time 5.

Observe that $\sum_{s=1}^{t} \mathcal{I}^{x,0}(v_s^d)$ corresponds to the actual inventory at time t at depot d. Since x is feasible for the nominal problem, this sum is always nonnegative. Given a nonzero δ, the net inventory at node v_t^d is changed by $\delta(v_t^d)$. Hence, the definition of $\mathcal{I}^{x,\delta}(v)$ models the net inventory availability (or requirement) at each node given the realization. Observe that $\sum_{s=1}^{t} \mathcal{I}^{x,\delta}(v_s^d)$ is not necessarily greater than or equal to 0 for all values of t. A negative value for this expression implies the necessity to reactively reposition units into the depot by time t in order to avoid a container shortage.

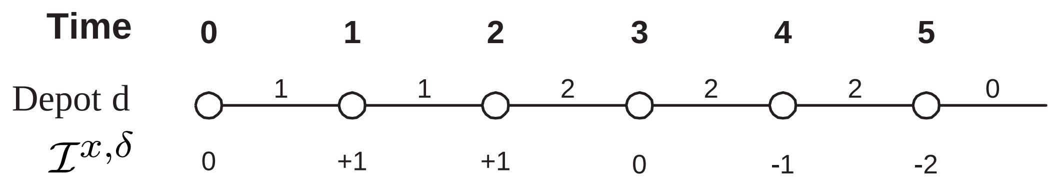

Figure 3: Value of $\mathcal{I}^{x,\delta}$ after perturbation $\delta(v_4^d) = -1$, $\delta(v_1^d) = \delta(v_2^d) = \delta(v_3^d) = \delta(v_5^d) = 0$. At least one unit must be in inventory by time 4 and at least 2 additional units by time 5 to recover feasibility.

Using $\mathcal{I}^{x,\delta}$, we can complete the definition of the recovery network G_{W_2}. Each arc $a \in A_{W_2}$ is given a flow lower bound $\ell(a) = 0$ and upper bound $u(a) = +\infty$. The net supplies d at each node are given by

$$\begin{aligned} d(v_t^d) &= \mathcal{I}^{x,\delta}(v_t^d) \quad \forall\, d \in D,\ t = 0, 1, 2, \cdots, \rho \\ d(s) &= -\sum_{v \in V} d(v) \end{aligned}$$

Let $G_{W_2}(x, \delta)$ refer to the recovery network with net supply vector d defined as above.

Proposition 1 *A feasible solution x for the nominal problem* (5) *belongs to the recoverable set $H(W_2, \delta)$ for a given δ if and only if there exists a feasible flow in $G_{W_2}(x, \delta)$.*

Proof: By construction of the network and its associated net supplies d, a feasible flow in $G_{W_2}(x, \delta)$ defines a set of feasible reactive repositioning decisions and inventory flow changes w restoring the feasibility of x given δ: for arcs $a \in R$, $w(a)$ is simply the flow on the corresponding arc in A_{W_2}, and for arcs $a_t^d = (v_t^d, v_{t+1}^d) \in I$ for $t > 1$, $w(a_t^d)$ is the flow on the arc in A_{W_2} minus $x(a_t^d)$. It is also not difficult to see that a w corresponding to an $x \in H(W_2, \delta)$ can be used to construct a feasible flow in $G_{W_2}(x, \delta)$: for $a \in R$, the flow on the associated arc in A_{W_2} is $w(a)$ and for $a \in I$, the flow is $x(a) + w(a)$. □

We now derive necessary and sufficient conditions for the existence of a feasible flow in general recovery networks $G_W(x, \delta)$, where W allows general flow changes on all inventory arcs $a_t^d = (v_t^d, v_{t+1}^d)$ for $t \geq 1$ and non-negative or zero flow changes on repositioning arcs. To do so, we first introduce two definitions.

Definition 3 (Competing Arc Set) *A set of inventory arcs $S \subseteq I$ is competing if every directed path P in G_W has $|P \cap S| \leq 1$.*

Such arcs essentially compete for inventory to protect against uncertainty, since resources moved to satisfy a need of one arc cannot be later used to satisfy the need of any other in the set since no path for flow exists.

Definition 4 (Inbound-closed Node Set) *A set of nodes $C \subseteq N_W$ is inbound-closed if there exists no directed path P in G_W from any node $i \in N_W \setminus C$ to any node $j \in C$.*

Since no reactive flow paths exist into an inbound-closed node set, no additional resources can be brought into these nodes to satisfy net demand generated in excess of nominal. Thus, each such set must contain enough pooled inventory to hedge against a worst-case outcome. Note that by definition, the set of inventory arcs outbound from an inbound-closed node set are competing.

Now we can define a set of cuts in G_W that characterize feasible flows in $G_W(x, \delta)$. Define

$$\mathcal{U}_W = \{U \subseteq N \mid U \text{ is inbound-closed}\}.$$

Proposition 2 *There exists a feasible flow in $G_W(x, \delta)$ if and only if for every set of nodes $U \in \mathcal{U}_W$*

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

Proof: Let $\Delta^{\text{out}}(U) = \{(v_1, v_2) \in A_W \mid v_1 \in U,\ v_2 \in N_W \setminus U\}$. It is known (see, *e.g.*, Cook et al., 1998) that there exists a feasible flow in $G_W(x, \delta)$ if and only if

$$\sum_{a \in \Delta^{\text{out}}(U)} \ell(a) \leq \sum_{v \in U} d(v) + \sum_{a \in \Delta^{\text{out}}(N \setminus U)} u(a) \qquad \text{for all } U \subseteq N. \tag{23}$$

Consider then any node set $U \subseteq N$ such that $U \notin \mathcal{U}_W$. By definition of $\mathcal{U}_W$, there must exist a $v_1 \in N \setminus U$ and $v_2 \in U$ where $(v_1, v_2) \in A_W$. Since $u((v_1, v_2)) = +\infty$, (23) is always satisfied for such U.

Now consider any node set $U \in \mathcal{U}_W$, and note that because U is inbound-closed, there are no arcs into set U. Since $\ell(a) = 0$ for all $a \in A_W$, (23) reduces to

$$0 \leq \sum_{v \in U} d(v) \quad \text{for all } U \in \mathcal{U}_W.$$

□

The necessary and sufficient conditions in Proposition 2 can be enforced through a set of constraints on the nominal flow variables, as proposed by the following theorem:

Theorem 2 *A feasible solution x of the nominal problem* (5) *is also feasible for* ***TRP2*** *if and only if for every set of nodes $U \in \mathcal{U}_{W_2}$*

$$\sum_{a \in \Delta^{out}(U) \cap I} x(a) \ \geq \ \vartheta(U, k)$$

Proof: Let x be a feasible solution of the nominal problem (5) and $\delta \in \varphi_k$. By definition of $\mathcal{I}^{x,\delta}$ it is clear that

$$\sum_{s=1}^{t} \mathcal{I}^{x,\delta}(v_s^d) = x(v_t^d, v_{t+1}^d) + \sigma((v_t^d, v_{t+1}^d)) \qquad \forall\, d \in D$$

and therefore that

$$\sum_{v \in U} \mathcal{I}^{x,\delta}(v) = \sum_{a \in \Delta^{\text{out}}(U) \cap I} x(a) + \sum_{d \in D} \sigma(a^d)$$

for every $U \in \mathcal{U}_W$, where $a^d \in \Delta^{\text{out}}(U) \cap I$ is the inventory arc for depot d in cut $\Delta^{\text{out}}(U)$. Thus, by Propositions 1 and 2, solution x is feasible for **ROP2** if and only if

$$\sum_{v \in U} \mathcal{I}^{x,\delta}(v) = \sum_{a \in \Delta^{\text{out}}(U) \cap I} x(a) + \sum_{d \in D} \sigma(a^d) \geq 0 \qquad \forall\, U \in \mathcal{U}_{W_2} \tag{24}$$

holds for each δ. But since $\delta \in \varphi_k$, $-\sum_{d \in D} \sigma(a^d)$ can be bounded:

$$-\sum_{d \in D} \sigma(a^d) \leq \vartheta(U, k) \quad \forall\, d \in D.$$

We note that this bound is tight for at least one $\delta \in \varphi_k$ (namely, $\delta(v) = -\hat{b}(v)$ for all $v \in \eta(U, k)$). Thus, condition (24) simplifies to

$$\sum_{a \in \Delta^{\text{out}}(U) \cap I} x(a) \geq \vartheta(U, k) \quad \text{for all } U \in \mathcal{U}_{W_2}.$$

□

As an aside, we note that the conditions in Theorem 2 (and the techniques used to develop the theorem) could also be used for the inventory robust problem **TRP1** given an appropriately-defined recovery network G_{W_1} and set $\mathcal{U}_{W_1}$.

Using Theorem 2, an alternative integer programming formulation for **TRP2** is:

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax = b \\ & \sum_{a \in \Delta^{\text{out}}(U) \cap I} x(a) \geq \vartheta(U, k) \quad \text{for all } U \in \mathcal{U}_{W_2} \\ & x \in \mathbb{Z}_+^{|\mathcal{A}|} \end{aligned}$$

Observe that the size of the constraint set specifying necessary and sufficient conditions for a feasible solution to **TRP2** is independent of the size of the realization set characterized by φ_k, and that the number of variables in the robust formulation is equal to the number in the nominal problem. However, the formulation requires a separate constraint for each element of $\mathcal{U}_{W_2}$, *i.e.*, for each inbound-closed node set. Fortunately, due to the structure of network, the number of such sets should be relatively small for practical instances; a demonstration is provided in the computational study reported in Section 5. Note that the number of inbound-closed node sets is relatively small because the time-space network has inventory arcs connecting all nodes associated with a specific depot across time and because repositioning travel times are short compared to the planning horizon.

4.2.3 A Restricted Inventory-Pooling Robust Repositioning Problem

Given a set of depots that pool inventory to hedge against uncertainty, a decision-maker may wish to limit further the options considered for reactive repositioning during the planning phase. One approach with practical appeal designates *a priori* those depots that serve only as providers of reactive resources, and those that serve only as recipients. Suppose that D is partitioned into two subsets: depots in D_s can reposition resources reactively to other depots, but do not receive such support, while those in D_r may only receive reactive resources. For example, in a geographic region an operator may have a large hub depot, and many smaller depots. The operator then might include the hub in D_s, and the smaller depots in D_r.

A recovery set W_3 can be specified for this scenario by a simple modification of W_2:

$$\begin{aligned} W_3 = \{w \in \mathbb{Z}^{|\mathcal{A}|} \mid w(a) &\geq 0 \quad \forall \quad a \in R, \\ w(a) &= 0 \quad \forall \quad a \in \{(v_0^j, u) \in R\} \,\cup\, \{(v_t^a, v_{t+h}^b) \in R \mid a \in D_r \text{ or } b \in D_s\}\}, \end{aligned}$$

and a restricted inventory-pooling robust repositioning problem is

$$\mathbf{TRP3} = \mathbf{TRP}(W_3, \varphi_k).$$

While it is not difficult to extend the analysis developed in Section 4.2.2 to determine a tight set of necessary and sufficient conditions defining feasible solutions x for **TRP3**, it turns out that these problems tend to require larger sets of constraints; the intuition behind this result is that fewer available reactive arcs lead to fewer reactive flow paths and therefore more competing arc sets and associated inbound-closed node sets. Since effective solution procedures may therefore require techniques to generate such constraints dynamically, we develop in this section some additional concepts that should prove useful for such techniques.

In the **TRP3** setting, each depot in D_s must not dispatch resources reactively that it will need later to cover its own needs. Recall that given a nominal solution x and a realization $\delta \in \varphi_k$, the inventory available at time t at depot $d \in D_s$ is given by

$$\sum_{s=1}^{t} \mathcal{I}^{x,\delta}(v_s^d) = x(v_t^d, v_{t+1}^d) + \sigma((v_t^d, v_{t+1}^d)) \qquad \forall\, d \in D.$$

Furthermore, any resources that are reactively repositioned from depot d at or before time t will reduce this available inventory. Since this adjusted inventory cannot fall below zero, we define the available container *support* $\mathcal{S}^{x,\delta}(v_t^d)$ at d at time t as the maximum number of resources that can be reactively repositioned from depot d by time t given nominal flow x and uncertain outcome δ, such that no container shortage occurs at d after time t. Mathematically,

$$\mathcal{S}^{x,\delta}(v_t^d) = \begin{cases} 0 & \text{if } t = 0 \\ \min_j\{x(v_j^d, v_{j+1}^d) + \sigma((v_j^d, v_{j+1}^d)) \mid t \leq j \leq \rho\} & \text{otherwise} \end{cases}$$

Note that for a fixed d, $\mathcal{S}^{x,\delta}(v_t^d)$ is a *non-decreasing* function of t. Support at time 0 is defined again to specify that no reactive repositioning is allowed at that time epoch.

Following the approach for **TRP2**, we first define a recovery network G_{W_3} that will be used to develop conditions for the existence of a feasible recovery flow given a nominal problem solution. The network G_{W_3} is specified using the procedure in Section 4.2.2; the arc set A_{W_3} thus contains no repositioning arcs outbound from depots in D_r and none inbound to depots in D_s.

The supply vector d in this case can be specified using the definitions of $\mathcal{I}^{x,\delta}$ and $\mathcal{S}^{x,\delta}$. For nodes associated with depots in D_r, the definition is unchanged. However, for nodes associated with depots in D_s, the net supply is equal to the incremental support available at time t:

$$\begin{aligned} d(v_0^d) &= 0 \quad \forall\, d \in D \\ d(v_t^d) &= \mathcal{I}^{x,\delta}(v_t^d) \quad \forall\, d \in D_r,\ t = 1, \cdots, \rho \\ d(v_t^d) &= \mathcal{S}^{x,\delta}(v_t^d) - \mathcal{S}^{x,\delta}(v_{t-1}^d) \quad \forall\, d \in D_s,\ t = 1, \cdots, \rho \\ d(s) &= -\sum_{v \in N \setminus \{s\}} d(v) \end{aligned}$$

A negative net supply at some node v_t^d where $d \in D_r$ indicates a demand for resources that must be served from inventory, or via reactive repositioning. The net supplies for nodes v_t^d where $d \in D_s$ specify the maximum number of additional units that can be repositioned out of depot d at time t so that no container shortage occurs later in time. Note that by the definition of support, a negative net supply can only occur at such a node when $t = 1$; in this case, there exists no feasible recovery flow.

Let $G_{W_3}(x, \delta)$ represent the recovery network G_{W_3} along with the associated net supply vector d. Again, a feasible flow in this network corresponds one-to-one with a valid recovery vector for a given realization δ.

Proposition 3 *A feasible solution x of the nominal problem (5) is a member of the recoverable set $H(W_3, \delta)$ for a given δ if and only if there exists a feasible flow in $G_{W_3}(x, \delta)$.*

Proof: Parallel to proof of Proposition 1. □

While valid necessary and sufficient conditions for the existence of a feasible flow in $G_{W_3}(x, \delta)$ are given by Proposition 2 using set $\mathcal{U}_{W_3}$, we now develop an explicit representation of these conditions using the special structure of W_3. To do so, we first introduce some additional notation. Given a set of inventory arcs $\alpha \subseteq I$, let $D(\alpha)$ be the set of depots corresponding to α and let $T(\alpha)$ be the set of tail nodes of arcs in α. Furthermore, let $C(\alpha)$ be the set of nodes from which the tail nodes of arcs in α can be reached, *i.e.*, the set of nodes from which there exists a directed path to the tail node of an arc in α. Note that $T(\alpha) \subseteq C(\alpha)$, and that $C(\alpha)$ is an inbound-closed set.

We can determine the existence of a feasible flow in $G_{W_3}(x, \delta)$ by considering only sets of nodes $C(\alpha)$ defined by sets of competing inventory arcs α at depots in D_r, where each arc $a \in \alpha$ has a shortage with respect to δ: $x(a) + \sigma(a) < 0$.

Proposition 4 *There exists a feasible flow in* $G_{W_3}(x, \delta)$ *if and only if*

$$d(v_1^d) = \mathcal{S}^{x,\delta}(v_1^d) \geq 0 \quad \forall\, d \in D_s \tag{25}$$

and

$$\sum_{v \in C(\alpha)} d(v) \geq 0 \tag{26}$$

for all $\alpha \subseteq I$ *where* α *is competing,* $D(\alpha) \subseteq D_r$*, and* $x(a) + \sigma(a) < 0$ *for each* $a \in \alpha$*.*

Proof: From Proposition 2, necessary and sufficient conditions for the existence of a feasible flow in $G_{W_3}(x, \delta)$ are

$$\sum_{v \in U} d(v) \geq 0 \quad \forall\ U \in \mathcal{U}_{W_3} \tag{27}$$

It is now shown that the conditions in the proposition are equivalent to the conditions given by (27).

The necessity of (25) and (26) is clear, since each constraint in the two sets corresponds directly to some set $U \in \mathcal{U}_{W_3}$ for which the expression in (27) must hold. For each $d \in D_s$, (25) corresponds to $U = \{v_0^d, v_1^d\}$ which is clearly inbound-closed by definition of D_s. Each $C(\alpha)$ generating a constraint (26) is also inbound-closed. Further, the arc set $\Delta^{\text{out}}(C(\alpha)) \cap I$ can be shown to be competing. This set is comprised of α and $\overline{\alpha}$, where each $a \in \overline{\alpha}$ is associated with a different depot $d \in D_s$. Therefore, since α is a competing arc set, and no path exists containing arcs both in α and arcs in $\overline{\alpha}$ by definition of $C(\alpha)$, and no path exists containing more than one arc $\overline{\alpha}$ by definition of D_s, $\alpha \cup \overline{\alpha}$ is competing. Thus, constraint (26) corresponding to $C(\alpha)$ also has a corresponding constraint in (27).

We now show the sufficiency of the conditions by showing that if they hold, conditions (27) hold for all $U \in \mathcal{U}_{W_3}$. Consider any $U \in \mathcal{U}_{W_3}$, and let $D_u \subseteq D_r$ be the set of depots d where there exists an arc $a^d \in \Delta^{\text{out}}(U) \cap I$ satisfying $x(a^d) + \sigma(a^d) \geq 0$. Note then that for each $d \in D_u$,

$$\sum_{v \in V^d \cap U} d(v) = x(a^d) + \sigma(a^d) \geq 0. \tag{28}$$

We claim first then that the conditions for U in (27) are redundant with those for $\widetilde{U} = U \setminus \bigcup_{d \in D_u} (V^d \cap U)$ in this case. Clearly this is true if $\widetilde{U} = \emptyset$. If $\widetilde{U} \neq \emptyset$, note that $\widetilde{U} \in \mathcal{U}_{W_3}$ since the subset $\Delta^{\text{out}}(\widetilde{U})$ of competing arcs $\Delta^{\text{out}}(U) \cap I$ is competing, and since no outbound reactive repositioning arcs exist in A_{W_3} from any depot $d \in D_u$ the set $\widetilde{U}$ remains inbound-closed. Suppose first that $\widetilde{U}$ contains only nodes associated with depots in D_s. In this case, conditions (25) guarantee that $\sum_{s=1}^{t} d(v_s^d) \geq 0$ for any $0 \leq t \leq \rho$ by definition of $\mathcal{S}^{x,\delta}$, and that therefore for any such $\widetilde{U}$, (27) holds. Along with (28), this in turn implies that (27) is satisfied for U.

Finally, suppose instead that $\widetilde{U}$ contains nodes for each depot in some set $\widetilde{D}_r \subseteq D_r$, in addition perhaps to nodes for some depots in D_s. Let $\alpha \subset \Delta^{\text{out}}(\widetilde{U}) \cap I$ where $D(\alpha) = \widetilde{D}_r$. Clearly, α is competing, and $x(a) + \sigma(a) < 0$ for each $a \in \alpha$ by the definition of $\widetilde{U}$. Further, $C(\alpha) \subseteq \widetilde{U}$, where any additional nodes in $\widetilde{U}$ must be associated with depots in D_s. Since $d(v_t^d) \geq 0$ for $d \in D_s$ by (25) and the definition of $\mathcal{S}^{x,\delta}$, if (26) holds for $C(\alpha)$ then (27) holds for $\widetilde{U}$, and finally (28) implies further that (27) is also satisfied for U. □

Proposition 4 can now be used to specify necessary and sufficient conditions for a feasible nominal repositioning plan x to be a feasible solution to **TRP3**. To do so, we first identify sets of arcs, denoted *vulnerable*, that require a constraint (26) to protect against a joint container shortage that will arise for at least one uncertain realization $\delta \in \varphi_k$:

Definition 5 *Given a feasible solution x to the nominal problem* (5)*, a set of inventory arcs $\alpha \subset I$ that are competing in G_{W_3} and where $D(\alpha) \subseteq D_r$ is* vulnerable *if there exists a $\delta \in \varphi_k$ such that $x(a) + \sigma(a) < 0$ for all $a \in \alpha$.*

Let $\Psi(x) = \{\alpha \subset I \mid \alpha \text{ is vulnerable}\}$. If α is vulnerable then there is $\delta \in \varphi_k$ such that $x(a) + \sigma(a) \leq -1$ for each $a \in \alpha$. Therefore, if α is vulnerable then

1. Each arc $a \in \alpha$ is weak, and
2. $\sum_{a\in\alpha} x(a) + |\alpha| \ \leq \ \vartheta(\hat{V}(\alpha), k)$.

where $\hat{V}(\alpha) = \bigcup_{a\in\alpha} \hat{V}(a)$, and $\hat{V}(a) = V_t^d$ for $a = (v_t^d, v_{t+1}^d) \in I$.

To guarantee the feasibility of x by Proposition 4, it is necessary to ensure that (26) is satisfied only for vulnerable arc sets α. To do so, we introduce the concept of *layer inequalities*. Let $D'(\alpha) \subseteq D \setminus D(\alpha)$ be the set of depots with nodes from which an arc in α can be reached. Let $t_\alpha^d = \max\{t : \ v_t^d \in C(\alpha)\}$ for all $d \in D'(\alpha)$. Finally, let

$$I^d(\alpha) = \left\{(v_t^d, v_{t+1}^d) \in I \ \mid \ t \geq t_\alpha^d\right\} \quad \forall \, d \in D'(\alpha).$$

Definition 6 *A layer of a vulnerable $\alpha \in \Psi(x)$, denoted $\theta(\alpha) \subset \bigcup_{d\in D'(\alpha)} I^d(\alpha)$, is a set of inventory arcs where*

$$|\theta(\alpha) \cap I^d(\alpha)| = 1 \quad \forall \, d \in D'(\alpha).$$

Each layer of α, therefore, contains one inventory arc from each $d \in D'(\alpha)$ departing at some time greater than or equal to t_α^d. An example of a layer is shown in Figure 4.

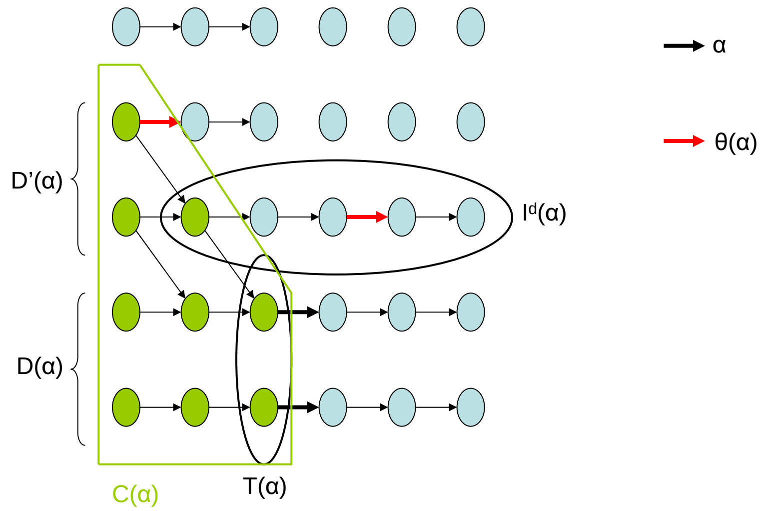

Figure 4: An example layer for a given vulnerable arc set α.

Let $\Theta(\alpha) = \{\theta(\alpha) \subset I \mid \theta(\alpha) \text{ is a layer of } \alpha\}$. Given α and a layer $\theta(\alpha)$, let

$$\widetilde{U}(\alpha, \theta(\alpha)) \;=\; \hat{V}(\alpha) \cup \hat{V}(\theta(\alpha))$$

Definition 7 *Given a feasible solution x to the nominal problem* (5)*, a vulnerable set $\alpha \in \Psi(x)$, and a layer $\theta(\alpha) \in \Theta(\alpha)$, let*

$$\sum_{a\in\alpha} x(a) + \sum_{a\in\theta(\alpha)} x(a) \geq \vartheta\left(\widetilde{U}(\alpha, \theta(\alpha)), k\right)$$

be the layer $\theta(\alpha)$ inequality.

We are now ready for the main theorem in this subsection:

Theorem 3 *A feasible solution x for the nominal problem* (5) *is also a feasible solution for **TRP3** if and only if*

$$x(a) \geq \vartheta(V_t^d, k) \quad \forall \;\; a = (v_t^d, v_{t+1}^d) \in I \text{ where } d \in D_s$$

and for every vulnerable set $\alpha \in \Psi(x)$, the layer $\theta(\alpha)$ inequality is satisfied for each $\theta(\alpha) \in \Theta(\alpha)$.

Proof: Let x be a feasible solution of the nominal problem (5). By Propositions 3 and 4, x is also a feasible solution for **TRP3** if and only if conditions (25) and (26) are satisfied for all $\delta \in \varphi_k$.

First we consider conditions (25). By the definition of $\mathcal{S}^{x,\delta}$, these conditions become

$$\min_{1\leq j\leq\rho} \left\{x(v_j^d, v_{j+1}^d) + \sigma((v_j^d, v_{j+1}^d))\right\} \geq 0 \quad \forall\, d \in D_s \quad \forall\, \delta \in \varphi_k,$$

which are equivalent to the following sets of constraints for each $d \in D_s$:

$$x(v_t^d, v_{t+1}^d) \geq -\sigma((v_t^d, v_{t+1}^d)) \quad \forall\, t \in \{1, \cdots, \rho\} \quad \forall\, \delta \in \varphi_k. \tag{29}$$

For any $\delta \in \varphi_k$,

$$-\sigma((v_t^d, v_{t+1}^d)) \leq \vartheta(V_t^d, k)$$

where we note that this bound is tight for at least one $\delta \in \varphi_k$. Therefore (29) simplifies to

$$x(v_t^d, v_{t+1}^d) \geq \vartheta(V_t^d, k) \quad \forall\, t \in \{1, \cdots, \rho\},$$

which must hold for all $d \in D_s$. Thus,

$$x(a) \geq \vartheta(V_t^d, k) \quad \forall\, a = (v_t^d, v_{t+1}^d) \in I \text{ where } d \in D_s.$$

Now consider conditions (26). Using the definition of vulnerable arcs, these conditions become

$$\sum_{v \in C(\alpha)} d(v) \geq 0 \quad \forall\, \alpha \in \Psi(x), \tag{30}$$

which again by Proposition 3 must hold for all $\delta \in \varphi_k$. The sum can be rewritten by partitioning the depots into those associated with the vulnerable arcs and those providing reactive repositioning support to the vulnerable arcs, and simplified as follows:

$$\begin{aligned}
\sum_{v \in C(\alpha)} d(v) &= \sum_{d \in D(\alpha)} \sum_{v \in C(\alpha) \cap V^d} d(v) + \sum_{d \in D'(\alpha)} \sum_{v \in C(\alpha) \cap V^d} d(v) \\
&= \sum_{d \in D(\alpha)} \sum_{v \in C(\alpha) \cap V^d} \mathcal{I}^{x,\delta}(v) + \sum_{d \in D'(\alpha)} \sum_{s=1}^{t_\alpha^d} \left(\mathcal{S}^{x,\delta}(v_s^d) - \mathcal{S}^{x,\delta}(v_{s-1}^d) \right) \\
&= \sum_{a \in \alpha} x(a) + \sum_{d \in D(\alpha)} \sigma(a^d) + \sum_{d \in D'(\alpha)} \mathcal{S}^{x,\delta}(v_{t_\alpha^d}^d) \\
&= \sum_{a \in \alpha} (x(a) + \sigma(a)) + \sum_{d \in D'(\alpha)} \min_{a \in I^d(\alpha)} \{x(a) + \sigma(a)\}.
\end{aligned}$$

where a^d is the inventory arc for depot d in α. Thus, the condition for a specific α in (30) can now be replaced by a set of inequalities, one for each layer $\theta(\alpha) \in \Theta(\alpha)$:

$$\sum_{a \in \alpha} (x(a) + \sigma(a)) + \sum_{a \in \theta(\alpha)} (x(a) + \sigma(a)) \geq 0.$$

Rewriting yields

$$\sum_{a \in \alpha} x(a) + \sum_{a \in \theta(\alpha)} x(a) \geq -\sum_{a \in \alpha} \sigma(a) - \sum_{a \in \theta(\alpha)} \sigma(a). \tag{31}$$

Since $\delta \in \varphi_k$, we have a tight bound of the right-hand side of (31) using $\vartheta(\widetilde{U}(\alpha, \theta(\alpha)), k)$, therefore

$$\sum_{a \in \alpha} x(a) + \sum_{a \in \theta(\alpha)} x(a) \geq \vartheta(\widetilde{U}(\alpha, \theta(\alpha)), k)$$

which is the layer $\theta(\alpha)$ inequality. $\square$

Theorem 3 allows **TRP3** to be formulated as the following integer program:

$$\begin{aligned}
\min \quad & c^T x \\
\text{s.t.} \quad & Ax = b \\
& x(a) \geq \vartheta(V_t^d, k) \quad \forall\, a = (v_t^d, v_{t+1}^d) \in I, \;\; d \in D_s \\
\sum_{a\in\alpha} x(a) + \sum_{a\in\theta(\alpha)} & x(a) \geq \vartheta(\tilde{U}(\alpha, \theta(\alpha)), k) \quad \forall\, \alpha \in \Psi(x), \;\; \forall \theta(\alpha) \in \Theta(\alpha) \\
& x \in \mathbb{Z}_+^{|\mathcal{A}|}
\end{aligned}$$

Note that the layer inequality constraints are specified for each vulnerable set α, and that the vulnerable arc sets depend on the solution x. If, alternatively, the layer inequality constraints are specified for each competing arc set α in G_{W_3} regardless of x (where $D(\alpha) \subseteq D_r$), the formulation remains valid and the dependence on x has been removed. However, Theorem 3 indicates that a given solution x can be checked for feasibility to **TRP3** using potentially far fewer constraints. This suggests that cutting plane algorithms, in which layer inequality constraints are added as necessary during solution, may be appropriate for solving **TRP3**. Future research may explore computational approaches along these lines.

4.3 Alternative Uncertainty Sets

Although the results developed above have assumed perturbation sets of the form specified by (7), it turns out that several different types of "budget" uncertainty sets may be used without changing the fundamental results. We now show that different budget uncertainty sets lead only to different specifications of the vulnerability ϑ of a node set V. In this section, let $\vartheta(V, \varphi)$ represent the vulnerability of node set V given an uncertainty set represented by the perturbation set φ.

1. *Maximum Scaled Deviation Per Depot*: Rather than limiting the maximum scaled deviation from the nominal net supply for all depots jointly, a decision-maker may wish to limit each depot's deviation separately. Assuming for simplicity that each depot has an identical budget k, the alternative perturbation set for this case is given by $\varphi_k^D = \{\delta \in \mathbb{Z}^{|\mathcal{N}|} \;:\; \delta(v) = \hat{b}(v)z(v),\, \sum_{v\in V^d} |z(v)| \leq k \;\forall\; d \in D,\, |z(v)| \leq 1 \;\forall\; v \in \mathcal{V}, \;\; \delta(s) = -\sum_{v\in\mathcal{V}} \delta(v)\}$. In this case, the vulnerability of a set $V \subseteq \mathcal{V}$ of nodes can be determined by

$$\vartheta(V, \varphi_k^D) \;=\; \max_z \left\{ \sum_{v\in V} \hat{b}(v)\, z(v) \;:\; \sum_{v\in V\cap V^d} |z(v)| \leq k \;\; \forall\, d \in D \;,\; |z(v)| \leq 1 \;\forall\; v \in V \right\},$$

a simple optimization problem similar to that given by (8) which can be solved by summing for each depot d the k largest values of $\hat{b}(v)$ for nodes $v \in V \cap V^d$, and then adding the sums for all depots.

2. *Telescoping Maximum Scaled Deviation*: One limitation in applying uncertainty set φ_k to problems with time-space nodes is that as the value of k is increased, the method may generate very conservative decisions in the first few planning periods since a large proportion of the demands may simultaneously take on their worst case values. This limitation can be avoided by using a telescoping uncertainty set. Let $\kappa = \{k_1, k_2, ..., k_\rho\}$ be a vector of budget parameters, $k_t \leq k_{t+1}$, where parameter k_t represents the maximum number of time-space net supplies that may take on their worst-case value by time period t. If we define $V_t = \cup_{d \in D} V_t^d$, the perturbation set for this case is given by $\varphi_k^T = \{\delta \in \mathbb{Z}^{|\mathcal{N}|} \ : \ \delta(v) = \hat{b}(v)z(v), \ \sum_{v \in V_t} |z(v)| \leq k_t \ \forall \ t \in \{1, ..., \rho\}, \ |z(v)| \leq 1 \ \forall \ v \in \mathcal{V}, \ \ \delta(s) = -\sum_{v \in \mathcal{V}} \delta(v)\}$. Then, the vulnerability of $V \subseteq \mathcal{V}$ can be determined by

$$\vartheta(V, \varphi_k^T) \ = \ \max_z \left\{ \sum_{v \in V} \hat{b}(v) \, z(v) \ : \ \sum_{v \in V \cap V_t} |z(v)| \leq k_t \ \ \forall \ t \in \{1, ..., \rho\} \ , \ |z(v)| \leq 1 \ \forall \ v \in V \right\}.$$

This optimization problem is also easy to solve, given that the nodes in V are sorted by non-decreasing order of $\hat{b}(v)$: set $z(v_t^d) = 1$ for the largest value of $\hat{b}(v_t^d)$, as long as $\sum_{v \in V \cap V_\tau} |z(v)| \leq k_\tau$ for $\tau \geq t$, then proceed to the next largest value of $\hat{b}(v)$ and repeat. Note that it would also be simple to construct such a telescoping maximum scaled deviation uncertainty set where the maximums were applied per depot rather than system-wide.

3. *Maximum or Telescoping Maximum Absolute Deviation*: Finally, while budget uncertainty sets that limit the maximum total scaled deviation from nominal have the benefit of being independent of the vectors b and $\hat{b}$, there may be cases where the decision-maker would rather limit the maximum absolute deviation from nominal for which he would like to plan; in fact, generating an appropriate value for such a statistic may be relatively simple from an analysis of past forecast accuracy. In the non-telescoping case, such a perturbation set could be represented by $\varphi_k^A = \{\delta \in \mathbb{Z}^{|\mathcal{N}|} \ : \ \delta(v) = \hat{b}(v)z(v), \ \sum_{v \in \mathcal{V}} \hat{b}(v)|z(v)| \leq k \ , \ |z(v)| \leq 1 \ \forall \ v \in \mathcal{V}, \ \ \delta(s) = -\sum_{v \in \mathcal{V}} \delta(v)\}$. In this case, the vulnerability of $V \subseteq \mathcal{V}$ can be determined simply by

$$\vartheta(V, \varphi_k^A) \ = \ \min \left\{ k, \sum_{v \in V} \hat{b}(v) \right\}.$$

A telescoping set and vulnerability could be similarly defined.

4.4 Joint Realizations and Conservatism

Lastly, it is important to note that by applying the TRP framework to repositioning problems using a perturbation set φ_k, the operator protects future feasibility for every realization $b + \delta$ with $\delta \in \varphi_k$. This may be overly conservative, since this set of realizations may be a superset of the possible realizations of the net supply vector $\widetilde{b}$ given that the components $\widetilde{b}(v)$ may not be independent.

Randomness in $\widetilde{b}$ may result from several sources. Examples include: (1) more (or less) customer demand on individual customer origin-destination lanes, (2) late (or early) arrival of forecasted customer demand, (3) late (or early) return of empty resources by customers, and (4) travel time for moves of loaded resources. While each source may clearly lead to variation in the components of $\widetilde{b}$, it is also clear that the changes in these components may often be negatively correlated. For example, if more demand materializes along a specific lane than expected, the result will be a negative perturbation in net supply at the origin depot and a corresponding positive perturbation at the destination depot at a later time epoch. Changes of the latter three types may actually induce correlated perturbations at even larger sets of nodes.

Although this is the case, it should be clear that ignoring such correlations by considering *all* realizations specified by the perturbation sets φ_k can only be more conservative than necessary, since the correlated perturbations are members of the set. Whether or not this leads to overly conservative solutions depends on the application context.

As a final note, it may be possible to refine the methodology presented here to explicitly handle uncertainty sets with component-wise dependence. For example, it may be possible in the **TRP2** framework to develop a better (smaller) estimate of the true potential vulnerability of a node set $U \in \mathcal{U}_W$ given negative correlations between the realized perturbations at nodes included in the set. We believe that this is an interesting avenue for additional research.

5 Using the Robust Repositioning Models in Practice

Although the sets of constraints required to specify the robust repositioning problems **TRP2** and **TRP3** grow large as the number of spatial locations grows, many real-world repositioning problems can be decomposed geographically such that the approaches proposed in this paper should be tractable. Furthermore, since recovery costs are ignored, it may be better that the user not include certain high-cost repositioning moves as allowable reactive moves.

To illustrate these ideas, we will consider example problem instances based on those that arise for a major tank container fleet operator. The company moves loads globally using ocean transportation, and faces significant loaded flow imbalance. Container depots are located near seaports worldwide, and can be naturally grouped into regions: *e.g.*, Southeast Asia, East Coast North America, Northern Europe, etc. While depots within a region might pool inventory to hedge against uncertainty, it might not be practical to allow reactive sharing across regional boundaries. This scenario can be modelled using a simple modification of **TRP2** or **TRP3** where inter-regional repositioning arcs are included in the nominal problem network G, but excluded from the specification of G_W. It should be clear, then, that the conditions in Theorems 2 and 3 decompose by region, leading to much smaller sets of constraints guaranteeing feasibility. Such decomposition also could allow the manager to specify certain regions where complete reactive pooling of the type specified by **TRP2** is allowed, and other regions where the restricted reactive pooling of **TRP3** is used.

The idea of partitioning a service area into regions is incorporated in the computational study now described. The computational experiment demonstrates that realistic instances of the empty repositioning problem can be solved effectively with the proposed robust optimization framework, and provides insight into how different levels of conservatism (determined by parameter k) and different degrees of flexibility for performing reactive repositioning (determined by the recovery constraints) affect the cost of a repositioning plan and therefore the price of robustness.

Consider now an example problem setting representative of those found in the tank container industry. Twenty depots, each located nearby a seaport of global importance, were chosen to be part of the network where the tank container operator provides transportation services. The depots were partitioned into eight geographic regions as described in Table 1. Transportation times between seaports were determined using a published schedule of port-to-port sailing for a large ocean carrier, in which the largest time corresponded to 5 weeks between ports in the east coast of North America and some ports in Asia. Transportation costs were assumed to be proportional to transportation times.

Region	Depots
1. South East Asia	Singapore, Port Kelang
2. East Asia	Hong Kong, Shanghai, Busan
3. Japan	Kobe, Tokyo
4. Northern Europe	Southampton, Rotterdam, Hamburg
5. Southern Europe	Algeciras, Gioia Tauro
6. North America West	Los Angeles, San Francisco, Seattle
7. North America East	New York, Norfolk, Savannah
8. South America	Buenos Aires, Rio de Janeiro

Table 1: Regions and depots in the computational test network.

In order to generate point forecasts of the net expected supply for each time period for each depot (*i.e.*, the vector of nominal values), a total of $3,000$ customer requests for loaded containers were randomly generated uniformly over a time period of 57 weeks. Each request requires a number of containers, which was randomly generated uniformly from the set $\{1, 2, 3, 4, 5\}$. Each request has an associated origin depot and destination depot (the determination of which is described in the following paragraph), where containers are assumed to be sourced from and returned to, respectively. Aggregating across all requests, the weekly net inflow of empty containers was calculated for each depot for each week; this value was then used as the corresponding point forecast $b(v)$ at each node v. Uncertain intervals were then constructed assuming fluctuations of up to 10% of nominal value: $\hat{b}(v) = \lceil 0.10 b(v) \rceil$.

To incorporate geographic trade imbalances into the test problem, the origin region and destination region of each request was randomly generated using the probabilities given in Table 2. Within in a region, a specific depot is randomly selected with equal probability. For this distribution data, regions in North America and Europe are on average net sources of empty containers, while regions in Asia and South America correspond to net sinks.

		Conditional probability of destination region							
Region	Probability of Origin	1	2	3	4	5	6	7	8
1	0.15	0	0.10	0.10	0.20	0.15	0.20	0.20	0.05
2	0.35	0.05	0	0.05	0.15	0.10	0.30	0.30	0.05
3	0.10	0.05	0.05	0	0.15	0.10	0.30	0.30	0.05
4	0.10	0.10	0.20	0.10	0	0.10	0.20	0.25	0.05
5	0.05	0.10	0.25	0.15	0.05	0	0.15	0.25	0.05
6	0.05	0.10	0.30	0.15	0.20	0.15	0	0.00	0.10
7	0.10	0.10	0.30	0.15	0.20	0.15	0.00	0	0.10
8	0.10	0.05	0.05	0.10	0.15	0.15	0.20	0.30	0

Table 2: Origin-destination distribution information for loaded demands between regions for computational test.

To avoid beginning and ending effects created by this approach for generating time-space net supplies, we truncated the problem horizon. The first 9 weeks and the final 8 weeks were eliminated from the initial 57 weeks of data, resulting in an instance with a 40 week planning horizon. The size of the container fleet was set at 600. Initial inventories of containers at each depot were determined proportional to the probability of a demand originating in its corresponding region.

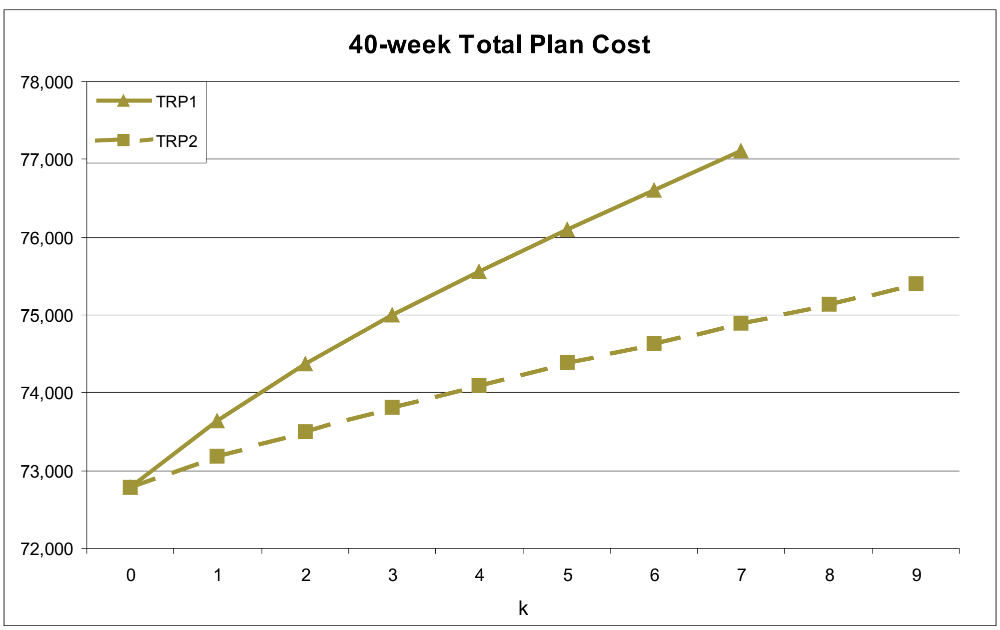

Figure 5: Total plan cost by fleet size and value of control parameter k for **TRP1** and **TRP2**.

The instance was solved using **TRP1** and **TRP2**, where for the latter, reactive repositioning was only allowed between depots in the same region. Control parameter k was varied from 0 (*i.e.*, solution to the nominal problem) to 9. Figure 5 summarizes cost results.

Interestingly, there are no feasible solutions for **TRP1** for values of k greater than 7. Note also that the total 40-week plan cost for **TRP2** increases only 3.6% from $k = 0$ to $k = 9$.

All solutions were obtained using CPLEX 9.0 with default parameter values on a PC with a 1.6 GHz processor and 1Gb of memory. In all cases it took fewer than 4 seconds of computation time to instantiate and solve an instance.

Lastly, to examine how decisions change when using a robust repositioning approach, we examine the planned average inventory levels at depots that result for different levels of k for both **TRP1** and **TRP2**. To avoid any bias resulting from our choice of initial inventory locations, a 2-phase approach was employed for this analysis. In Phase I, we solve the full 40-week problem, and assume that decisions for the first 12 weeks of the plan are implemented. Based on these decisions, we update the nominal net supply values for weeks 13 through 17 (recall that the maximum transportation time is 5 weeks). Then, in Phase II, we solve a 28-week instance corresponding to weeks 13 through 40. Our inventory analysis is based on the results of the 28-week repositioning instance solved in Phase II.

Figures 6 and 7 summarize average inventory by region for different levels of k for **TRP1** and **TRP2**, respectively; note that these figures include regions that are net demanders of empty containers. Not unexpectedly, observe that in both cases, average inventory per region increases as the value of k increases. At $k = 0$, which corresponds to the nominal problem, little inventory is kept at these depots; on the other hand, for values of $k > 0$, safety-stock inventory starts being consider in the repositioning plan in order to hedge against uncertainty.

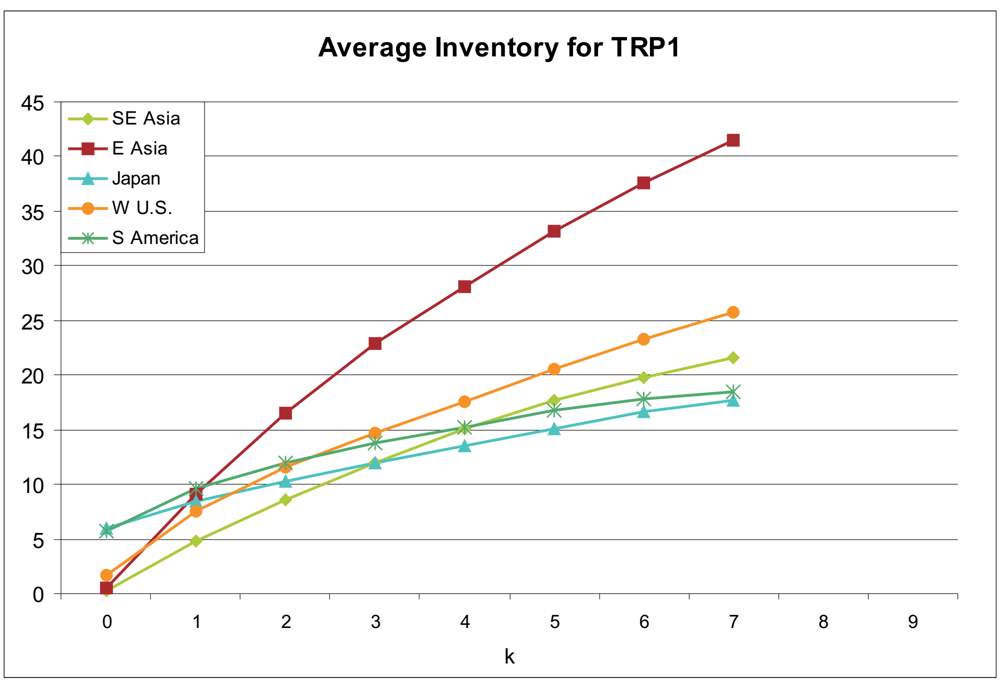

Figure 6: Average inventory per region by value of control parameter k for **TRP1** given a fleet size of 600 containers.

The effect of inventory pooling can be observed by contrasting Figures 6 and 7. When reactive repositioning is allowed between depots in the same region, the plan is recoverable with respect to the same level of uncertainty, defined by parameter k, with far fewer containers of inventory per region.

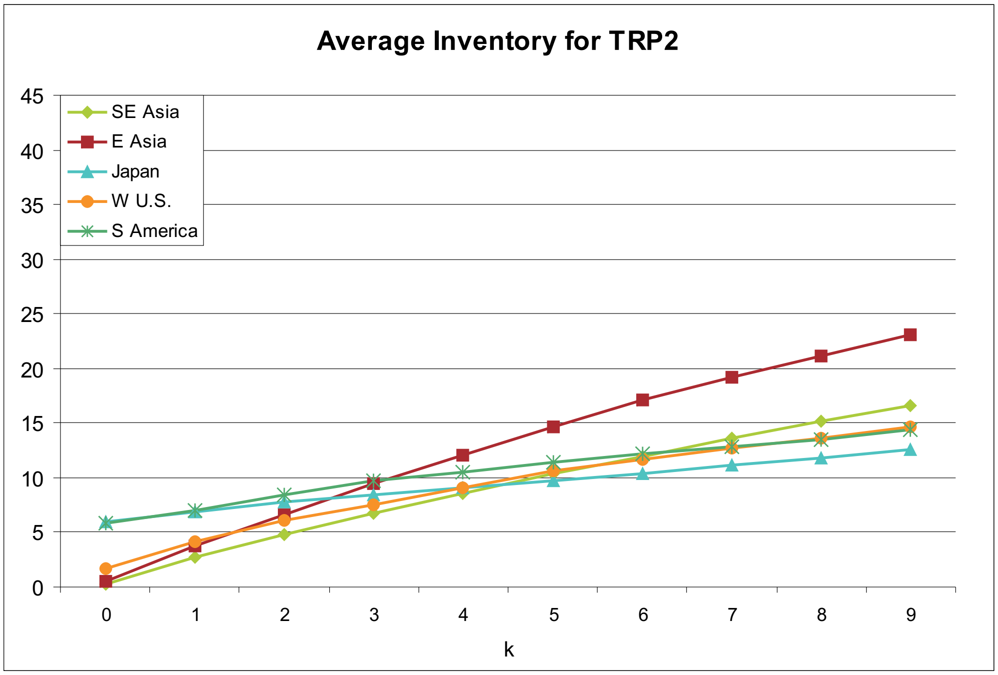

Figure 7: Average inventory per region by value of control parameter k for **TRP2** given a fleet size of 600 containers.

References

J. Abrache, T. G. Crainic, and M. Gendreau. A new decomposition algorithm for the deterministic dynamic allocation of empty containers. Technical Report CRT-99-49, Centre de Recherche sur les Transport, 1999.

S. Ahmed and A. Shapiro. Two-stage robust network flow and design under demand uncertainty. *Optimization Online*, http://www.optimization-online.org, 2002.

A. Atamturk and M. Zhang. Two-stage robust network flow and design under demand uncertainty. *Operations Research*, to appear, 2006.

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

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

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

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

D. Bertsimas and M. Sim. The price of robustness. *Operations Research*, 52:35–53, 2004.

D. Bertsimas and A. Thiele. A robust optimization approach to supply chain management. In *Proceedings of the 10th International Integer Programming and Combinatorial Optimization (IPCO) Conference*, pages 86–100, 2004.

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

D. Bertsimas, D. Pachamanova, and M. Sim. Robust linear optimization under general norms. *Operations Research Letters*, 32:510–516, 2004.

J.R. Birge and F. Louveaux. *Introduction to Stochastic Programming*. Springer, New York, 1999.

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

W.J. Cook, W.H. Cunningham, W.R. Pulleyblank, and A. Schrijver. *Combinatorial Optimization*. John Wiley and Sons, New York, 1998.

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

A.L. Erera, J.C. Morales, and M.W.P. Savelsbergh. Global intermodal tank container management for the chemical industry. *Transportation Research, Part E*, 41:551–566, 2005.

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

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

G. Godfrey and W.B. Powell. An adaptive dynamic programming algorithm for single-period fleet management problems II: Multiperiod travel times. *Transportation Science*, 36:40–54, 2002b.

A. Kleywegt, A. Shapiro, and T. Homem de Mello. The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12:479–502, 2001.

P. Kouvelis and G. Yu. *Robust Discrete Optimization and Its Applications*, volume 14 of *Nonconvex Optimization and Its Applications*. Kluwer Academic, Dordrecht, Germany, 1997.

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

S. Misra. Linear programming of empty wagon disposition. *Rail International*, 3:151–158, 1972.

W.B. Powell. Dynamic models of transportation operations. In S. Graves and T. A. G. de Tok, editors, *Handbooks in Operations Research and Management Science: Supply Chain Management*, pages 677–756. Elsevier, Amsterdam, 2003.

W.B. Powell. A stochastic formulation of the dynamic vehicle allocation problem. *Transportation Science*, 20:117–129, 1986.

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

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

Stolt-Nielsen Transportation Group. Private communication. 2004.

H. Topaloglu and W.B. Powell. Dynamic programming approximations for stochastic, time-staged integer multicommodity flow problems. *Informs Journal on Computing*, to appear, 2004.

W. White. Dynamic transshipment networks: an algorithm and its application to the distribution of empty containers. *Networks*, 2:211–236, 1972.