Submitted to
manuscript (Please, provide the manuscript number!)

Robust location-transportation problems with integer-valued demand

Ahmadreza Marandi*
Department of Industrial Engineering and Innovation Sciences, Eindhoven University of Technology, The Netherlands
*Corresponding author: a.marandi@tue.nl

Geert-Jan van Houtum
Department of Industrial Engineering and Innovation Sciences, Eindhoven University of Technology, The Netherlands

A Location-Transportation (LT) problem concerns designing a company's distribution network consisting of one central warehouse with ample stock and multiple local warehouses for a long but finite time horizon. The network is designed to satisfy the demands of geographically dispersed customers for multiple items within given delivery time targets. The company needs to decide on the locations of local warehouses and their basestock levels while considering the optimal shipment policies from central or local warehouses to customers.

In this paper, we deal with integer uncertain demands in LT problems to design a robust distribution network. We prove two main characteristics of our LT problems, namely convexity and nondecreasingness of the optimal shipment cost function. Using these characteristics, we show for two commonly used uncertainty sets (box and budget uncertainty sets) that the optimal decisions on the location and the basestock levels of local warehouses can be made by solving a polynomial number of deterministic problems. For a general uncertainty set, we propose a new method, called Simplex-type method, to find a locally robust solution. The numerical experiments show the superiority of our method over using the integer-valued affine decision rules, which is the only available method for this class of problems.

Key words: Location-Transportation problem, Multi-stage discrete robust optimization, Affine decision rule, Simplex-type method

History: March 11, 2020

1. Introduction

An optimal design of a distribution network is extremely important but at the same time complicated, as it involves the trade-off between the customers satisfaction and the costs to deliver the demanded products to the customers within a reasonable time. To design a distribution system, Crainic and Laporte (1997) identify three layers of decisions: strategic (long-term) planning decisions, which are the ones on the locations of the facilities, warehouses, or hubs; tactical (medium-term) planning decisions, which includes the distribution of resources among the locations; and the operational (short-term) planning decisions, which concern how to allocate resources to satisfy the demands. Problems containing all three layers of decisions are called *Location-Transportation (LT) problems.*

Cooper (1972) is the first who studies LT problems and develops an optimization model to find the optimal location of facilities while considering the possibility of shipments from multiple facilities to any customer. Later on, his model has been extended to include extra decisions on the amount of available resources in each facility and to capture different characteristics of the problem; we refer the reader to the review papers by Nagy and Salhi (2007) and Prodhon and Prins (2014) and the references therein.

LT problems arise naturally in many real-life situations. One of the situations concerns e-commerce distribution networks. Currently, the use of e-commerce activities has been dramatically increased and different e-commerce companies have been established with different services. One of the recent services in e-commerce companies, offered for instance by Amazon (Holsenbeck 2018), is a two-hour delivery service in which orders containing specific classes of items are delivered to the customers within two hours after the placement of the order. To be able to offer such services, the company needs a distribution network where the warehouses are in strategic locations each of which contains specific items. Therefore, e-commerce companies offering this service should know where to open warehouses, how many of which items should be stocked in each warehouse, and which warehouse(s) should be assigned to an order.

Another important situation where LT problems arise is in the service logistics of capital goods. Capital goods are used to produce other products or to provide services to the customers. Examples of capital goods are lithography systems, MRI scanners, aircraft, etc. Typically, customers of capital goods need a high system availability, i.e. a high proportion of time that the system is functional. Hence, companies producing capital goods typically provide fast services for the system maintenance (Jiang 2012, Koninklijke Philips N.V. 2019). An important aspect of system maintenance is availability of spare parts, which are required to fix failures in the systems. To meet the high system availability requirements, original equipment manufacturers (OEMs) typically have service logistics networks containing different types of warehouses. To minimize the total cost, it is important for OEMs to optimize the location of warehouses, the amounts of spare parts stocked at each of them, as well as the shipment policies to assign parts to customers.

As the above-mentioned examples reveal, one of the challenges in making integrated decisions is demand uncertainty. All decisions are made to satisfy customers' demands, and the uncertainty in the demand increases the complexity of the problem. Two typically used methodologies to deal with demand uncertainty are stochastic optimization (Klibi et al. 2010) and *robust optimization* (Ardestani-Jaafari and Delage 2017). In this paper, we contribute to the literature concerning robust optimization methodology and show how one can solve real-life LT problems.

In robust optimization, optimal decisions are made such that they are safe-guarded against any realizations of the uncertain parameter in a user-specified set, called *uncertainty set*. The first method in robust optimization, formally introduced by Ben-Tal and Nemirovski (2000), is Static Robust Optimization (SRO), where all decisions are assumed to be taken *before* the uncertain parameter is realized. Such decisions are called "*here-and-now*". In other words, in SRO all decisions on the locations of warehouses, distribution of resources among the warehouses, and the resource allocations to the customers are made such that they are safe-guarded against the realization of the demands in the uncertainty set, and they are made *before* the realization of the customers demand. Baron et al. (2011) compare the optimal decisions on the three layers obtained by solving three problems: (i) a deterministic problem, typically called nominal problem, where the demand is assumed to be the average value of past demands, (ii) a robust problem with a *hyper-box* uncertainty set formulated using SRO, and (iii) a robust problem with a hyper-ellipsoid uncertainty set formulated using SRO. They show that the decisions obtained by solving the robust problems outperform the decisions obtained by solving the nominal problem. For another class of uncertainty sets, called *budget uncertainty set*, Bardossy and Raghavan (2013) study a similar problem excluding the decisions on the tactical planning level and propose an approximate algorithm to solve the robust problem formulated using the SRO method.

The solutions obtained by SRO method are conservative, as the operational planning decisions (the allocation of resources to customers) are taken before the realization of the demand. In reality, however, these decisions are made after the demands from customers are revealed. To capture a more realistic setting and to reduce the conservativeness of obtained solutions using the SRO method, Ben-Tal et al. (2004) propose a new method, called Adjustable Robust Optimization (ARO), where part of the decisions are "here-and-now", while the rest of the decisions are made *after* the realization of the uncertain parameter, hence so-call "*wait-and-see*" decisions.

As ARO is capable of formulating multi-stage real-life situations, it attracts attention and there have been extensive studies on practicability and solvability of robust problems using the ARO method, in general; see, e.g., Ben-Tal et al. (2004), Bertsimas and Goyal (2012), Bertsimas and Bidkhori (2015), Marandi and den Hertog (2018), Bertsimas et al. (2010) and the review papers Gabrel et al. (2014b), Yanıkoğlu et al. (2019).

To apply ARO to LT problems, it is important to notice that after the tactical planning decisions (distribution of resources among the opened warehouses) are made, the operational planning decisions (resource allocations) are taken several times to satisfy the realized demand of customers in different time-slots. For instance, an e-commerce company may replenish the stock of the warehouses once per day but it has to ship products several times from different warehouses to customers during a day to satisfy the two-hour delivery service. Therefore, the LT problems are formulated as multi-stage robust optimization problems. Atamtürk and Zhang (2007) show that even the two-stage LT problems belong to the class of NP-hard problems, and provide a method to approximate them for budget uncertainty sets. Then, Gabrel et al. (2014a) develop an iterative cutting plane algorithm to approximate two-stage LT problems. Simchi-Levi et al. (2018) consider a similar model with a more complicated uncertainty set and derive the optimal resource allocation policy. Ardestani-Jaafari and Delage (2017) is one of the few who analyze multi-stage LT problems. They design three mathematical models to approximate the problems with a budget uncertainty set and develop a row generation algorithm to tackle large-scale problems.

The common assumption made in the literature, which is not always practical for LT problems, is that the demands are nonnegative *real-valued* and the uncertainty set is a *convex set*. Based on this assumption, the existing models for LT problems are developed by (re)formulating the operational planning decisions as the proportion of the customers' demands satisfied by different warehouses. However, in many real-life applications, especially those mentioned above, this assumption cannot be justified. So, clearly such ways of reformulating is not applicable to problems with *integer-valued demands*. To the best of our knowledge, there is no literature on how to deal with multi-stage LT problems with integer-valued demands. So, in this paper, we close the existing gap in the literature by making a three-fold contribution.

First, we show that for a multi-stage LT problem with integer-valued demands, the minimum cost of resource allocations is a convex function in demand (*convexity characteristic*). Using this characteristic we then provide two techniques to solve a multi-stage LT problem: (I) we show that a continuous relaxation of an LT problem can be used to obtain the worst-case optimal value as well as the optimal strategic and tactical decisions. This result implies that the techniques proposed in the literature to solve robust problems by using the ARO method with real-valued demands can be employed to solve multi-stage LT problems with integer-valued demands. (II) we show that, using an enumeration of vertices of the convex hull of the uncertainty set, one can solve multi-stage LT problems to optimality. This means that a multi-stage LT problem with an uncertainty set of which the convex hull contains a polynomial number of vertices is computationally tractable and large-sized problems can be solved efficiently.

Second, we show that for a class of multi-stage LT problems, the minimum cost of resource allocations is nondecreasing in demand. In other words, we mathematically prove that an increase in the demands of different customers for different items results in an equal or higher optimal cost of satisfying the demands (*nondecreasingness characteristic*). Based on this characteristic, we show that under some conditions, a multi-stage LT problem with a hyper-box uncertainty set can be solved using a deterministic LT problem. This result implies that a real-life multi-stage LT problem with a hyper-box uncertainty set can be solved efficiently. Furthermore, under the same conditions, we show that the worst-case optimal value of a multi-stage LT problem can be obtained by replacing the uncertainty set with its subset, which can lead to an improvement regarding the computational complexity. For instance, using this result we show that for a class of multi-stage LT problems with a specific budget uncertainty set, the problem is computationally tractable.

Finally, we design a new method to approximate a multi-stage LT problem with integer-valued demands by obtaining a locally robust solution. Our method is developed using the convexity characteristic and is based on a similar principle as the Simplex algorithm; therefore we call this method a *Simplex-type method*. In each iteration of the Simplex-type method many deterministic LT problems are solved, which facilitates parallel computing and solving large-size problems. In our numerical experiments, we illustrate that the larger the size of a problem instance, the better the performance of the Simplex-type method in comparison to the integer-valued affine decision rules, proposed by Bertsimas and Caramanis (2007).

The structure of the remainder of the paper is as follows. Section 2 provides the description of an LT problem. Section 3 contains the formulation of a multi-stage LT problem and the use of the ARO method. In Section 4, we prove the theoretical results of this paper regarding the convexity and nondecreasingness characteristics, and show how one can use them to find the optimal solutions. In Section 5, we show how to approximate a multi-stage LT problem using the integer-valued affine decision rules, and develop the Simplex-type method. Section 6 contains the numerical experiments of this paper and illustrates the effectiveness of our results in finding a globally or locally robust solution. We conclude the paper in Section 7.

2. Model Description

We consider a company serving many geographically dispersed customers who place integer-valued demands for multiple items. The customers are numbered from 1 to J, and $\mathcal{J}$ denotes the set of customers. The items are numbered from 1 to K, and $\mathcal{K}$ denotes the set of items. We look at a long but finite time horizon ahead and decide about the locations of local warehouses. These locations are chosen from candidate locations $1,\ldots,I$ at the beginning of the time horizon; $\mathcal{I}$ denotes the set of candidate locations. The locations are assumed to be fixed during the whole time horizon. The time horizon is divided into L periods of equal length (e.g., weeks), and the local warehouses are replenished from a central warehouse at the beginning of each period. The corresponding replenishment leadtimes are negligibly small. The location of the central warehouse is given and we assume that the central warehouse has always enough stock for replenishments and to directly satisfy demands from customers. For the central warehouse, we use index 0. Each period consists of short time-slots numbered $1,\ldots,T$; let $\mathcal{T}=\{1,\ldots,T\}$. For each item and customer, demands per period are stationary, and they may have a periodic pattern within a period. This means that the demands during the t^{th} time-slot in any period (t^{th} time-slot of the first period, t^{th} time-slot of the second period, etc.) are independent and belong to the bounded uncertainty set $\mathcal{D}_t \subseteq \mathbb{N}_0^{J\times K}$ with $\mathbb{N}_0$ defined as $\mathbb{N}\cup\{0\}$.

Customers place their orders during a time-slot. At the end of a time-slot, the company decides for each demanded unit of item k by customer j which warehouse satisfies this demanded unit. Next, the part is delivered by a fast shipment. For a unit of item k demanded by customer j, a delivery time target t_{jk}^{max} is given (counting of the delivery time starts at the end of the time-slot). If the unit can be delivered from a sufficiently nearby local warehouse, this target will be met. Otherwise the target is exceeded and a penalty cost is paid per unit of tardyness. Obviously, the local warehouses are limited by their on-hand stocks when satisfying demands. But, it is always possible to satisfy a demand from the central warehouse. We assume, as justified by Klincewicz (1990) and Limbourg and Jourquin (2009), that the shipment cost from a location i to customer j for item k at the end of the t^{th} time-slot is linear in the shipment quantity z_{ijkt}, and the corresponding unit cost is denoted by $\bar{c}_{ijk}$ (≥ 0). This unit cost is built up as follows. Let t_{ij} (≥ 0) be the fast shipment time from location i to customer j. The corresponding delivery cost per unit of item k is given by c^d_{ijk} and forms the first part of $\bar{c}_{ijk}$. The second part is formed by a unit penalty cost c^p_{jk} multiplied with the tardyness when the shipment from location i to customer j is not fast enough to meet the delivery time target t^{max}_{jk}. This leads to the following formula for $\bar{c}_{ijk}$:

$$\bar{c}_{ijk} = c^d_{ijk} + c^p_{jk}\left(t_{ij} - t^{max}_{jk}\right)^+, \qquad \forall i \in \mathcal{I} \cup \{0\}, j \in \mathcal{J}, k \in \mathcal{K},$$

where $(x)^+ = max\{x, 0\}$ for $x \in \mathbb{R}$.

Apart from shipment costs, we also have costs for opening local warehouses, inventory holding costs, and replenishment costs. Let $\bar{F}_i$ be the cost for opening a local warehouse at candidate location i. Because total demands per period are stationary and replenishments are made at the beginning of each period, it is reasonable to assume that the on-hand inventory for each item k at local warehouse i is increased to the same level S_{ik} at the beginning of each period. I.e., we can assume that a basestock policy is followed by each local warehouse for each of the items. We assume that an inventory holding cost h_{ik} is charged for each unit of on-hand stock of item k at local warehouse i at the beginning of each time-slot.

A replenishment cost, which is incurred at the beginning of each period, consists of a fixed cost as well as the transportation cost from the central warehouse to the local warehouses. In the beginning of the time horizon, the replenishment quantity equals the basestock level of local warehouses. The effect of this cost on the total replenishment cost in the whole horizon is very small, as we consider a long time horizon. Therefore, we ignore this effect in our cost computations. From the second period onwards, the total number of goods that are transported from the central warehouse to the local warehouses per period is equal to the total fulfilled demand in the previous period.

In our model, we consider the total fixed cost of replenishing a local warehouse i in the time horizon together with the fixed cost of opening this warehouse, and denote the aggregated cost by F_i. Furthermore, the unit transportation cost of replenishing local warehouse i for item k is

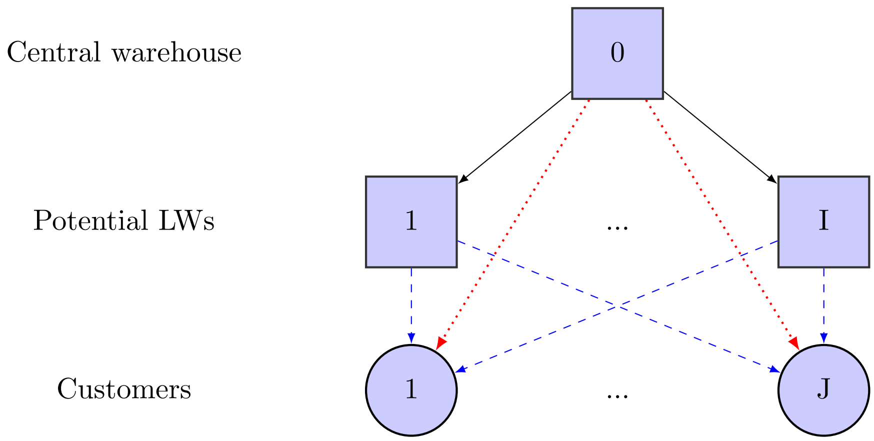


Figure 1 **The schematic illustration of an LT problem with one central warehouse, I candidate locations for the local warehouses (LWs), and J customers. The solid (black), dashed (blue), and dotted (red) arrows are depicted to show the replenishment of the LWs, possible shipments from LWs to each customers, and possbile shipments from central warehouse to each customer, respectively.**

combined with the unit shipment cost from the same warehouse to a customer j, and the aggregated cost is denoted by c_{ijk}.

Figure 1 provides a schematic illustration of the problem considered in this paper. In this figure, the solid black arrows show the replenishments, the blue dashed arrows show the shipments from a potential local warehouse (LW) to the customers, and the red dotted arrows illustrate the shipments from the central warehouse (CW) to the customers.

The company needs to make three types of decisions. The first type of decisions is on whether a local warehouse is opened at the candidate location i at the beginning of the time horizon. We model this decision by a binary variable y_i (with value 1 if the warehouse is opened). The second type of decisions is about the basestock level of the local warehouse i at the beginning of periods, which is denoted by S_{ik}. The final type of decisions is on how the company satisfies demand $d_{jkt\ell}$ from customer j for item k at the end of the t^{th} time-slot in period ℓ. This type of decisions is based on the realization of the demand and hence a function variable $z_{ijkt\ell}(d) \in \mathscr{Z}$ is chosen, where $\mathscr{Z}$ is the space of all functions from $\mathbb{N}_0^{J\times K\times T\times L}$ to $\mathbb{N}_0$.

To design the distribution network, one can analyze only one period. To see that, consider the problem where the time horizon contains L periods. Then, the total cost consists of a fixed cost

Notation	Description
Set	
$\mathcal{J}$	the set of J customers, indexed by j
$\mathcal{I}$	the set of I candidate locations for local warehouses, indexed by i, i'; the central warehouse is denoted by 0
$\mathcal{K}$	the set of K different items, indexed by k
$\mathcal{T}$	the set of T time-slots between two replenishments, indexed by $t, \tilde{t}$
Parameters	
f_i	proportion of the fixed cost of opening the local warehouse i and replenishing it to the number of periods in the time horizon
h_{ik}	unit inventory holding cost of item k in the local warehouse i
c_{ijk}	aggregated unit replenishment cost of item k from central warehouse to the local warehouse i and shipment cost from it to customer j
t_{ij}	delivery time of a shipment from the local warehouse i to customer j
t_{jk}^{max}	delivery time target of the customer j for item k
d_{jkt}	demand from customer j during time-slot t for item k
Variables	
$y_i \in \{0, 1\}$	decision on the i^{th} local warehouse gets opened
$S_{ik} \in \mathbb{N}_0$	the basestock level of item k in the local warehouse i
$z_{ijkt} \in \mathbb{N}_0$	the shipment quantity sent from local\central warehouse i of item k to the customer j at the end of time-slot t

Table 1 **Nomenclature. The term "local warehouse i" refers to the potential local warehouse in the i^{th} candidate location.**

of opening local warehouses at the beginning of the time horizon and fixed cost of replenishing them during the time horizon as well as the holding costs and the shipment costs during the first, second, ..., L^{th} period. Because of the stationary periodic demand, the highest possible total cost (worst-case cost) during the first period is the same as during the second period, ..., and the same as during the L^{th} period. Therefore, the total worst-case cost during the time horizon is equal to the fixed costs, $\sum_{i\in\mathcal{I}} F_i y_i$, and L times the worst-case holding and shipment costs during the first period. Minimizing this total worst-case cost is equivalent to minimizing $\sum_{i\in\mathcal{I}}(F_i/L)\cdot y_i$ plus one time the worst-case holding and shipment costs during the first period. This shows that, without loss of generality, we can reduce the problem of minimizing the worst-case total cost during the time horizon to the problem of minimizing the scaled fixed costs of opening local warehouses plus one time the worst-case holding and shipment costs during one period. Therefore, we remove the ℓ index from our notation and introduce $f_i = F_i/L$ as the scaled fixed costs for location i. The summary of the notation used in this paper is presented in Table 1.

3. Adjustable Robust Optimization Formulation

In this section, we first describe the adjustable robust optimization (ARO) formulation of the problem for the case with only one time-slot in each period ($T=1$). Then, we generalize the formulation to any number of time-slots in a period.

Let us consider the chronological sequence of events that occur during the time horizon. First, the decisions on y_i, $i\in\mathcal{I}$, and the basestock levels S_{ik}, $i\in\mathcal{I}$ and $k\in\mathcal{K}$, are made. Then, during the time-slot, d_{jk1} is realized. Finally, at the end of the time-slot, z_{ijk1}, $i\in\mathcal{I}\cup\{0\}$, $j\in\mathcal{J}$, $k\in\mathcal{K}$, is chosen based on the realized demand. So, given which local warehouses are opened, $y=[y_i]_{i\in\mathcal{I}}$, their basestock levels, $S=[S_{ik}]_{i\in\mathcal{I},k\in\mathcal{K}}$, and the realized demand, $d=[d_{jk1}]_{j\in\mathcal{J},k\in\mathcal{K}}$, the optimal shipment policy can be obtained by solving

$$
\begin{aligned}
Q(y,S,d)=\min_{z(d)} \quad & \sum_{\substack{i\in\mathcal{I}\cup\{0\},\\ j\in\mathcal{J},\\ k\in\mathcal{K}}} c_{ijk} z_{ijk1}(d) && && \text{(1a)}\\
\text{s.t.} \quad & \sum_{j\in\mathcal{J}} z_{ijk1}(d)\le S_{ik}, && \forall i\in\mathcal{I},k\in\mathcal{K}, && \text{(1b)}\\
& \sum_{i\in\mathcal{I}\cup\{0\}} z_{ijk1}(d)=d_{jk1}, && \forall j\in\mathcal{J},k\in\mathcal{K}, && \text{(1c)}\\
& z_{ijk1}(d)\in\mathbb{N}_0, && \forall i\in\mathcal{I}\cup\{0\},j\in\mathcal{J},k\in\mathcal{K}, && \text{(1d)}
\end{aligned}
$$

where $z(d) := [z_{ijk1}(d)]_{\substack{j\in\mathcal{J}\\ i\in\mathcal{I}\cup\{0\}\\ k\in\mathcal{K}}}$, and we refer to it as the vector of shipment policies, which is a function of the realized demand d. The objective function (1a) is the total shipment costs from the central and local warehouses. Constraints (1b) and (1c) guarantee that the shipments from local warehouses do not exceed their stock level and the customers' demands are satisfied, respectively.

Given an uncertainty set $\mathcal{D}_1$, which contains possible realizations of the demand vector d against which the decision maker wants to be safe-guarded, the adjustable robust optimization (ARO) formulation is

$$\min_{y,S} \quad \sum_{i\in\mathcal{I}} f_i y_i + \sum_{\substack{i\in\mathcal{I}\\ k\in\mathcal{K}}} h_{ik}S_{ik} + \max_{d\in\mathcal{D}_1} Q(y,S,d) \tag{2a}$$

$$\text{s.t.} \quad S_{ik} \le M y_i, \qquad \forall i\in\mathcal{I}, k\in\mathcal{K}, \tag{2b}$$

$$S_{ik}\in\mathbb{N}_0, \qquad \forall i\in\mathcal{I}, k\in\mathcal{K},$$

$$y_i\in\{0,1\}, \qquad \forall i\in\mathcal{I},$$

where the objective function (2a) is the total worst-case cost and constraint (2b) makes sure that the basestock levels of the closed local warehouses are zero, with M being a large enough number.

The formulation of the problem can be easily generalized to T time-slots by considering the holding cost and guaranteeing that the stock level at the beginning of each time-slot is the same as the stock level at the beginning of the previous time-slot minus the shipment quantities to the customers in the previous time-slot. Let us denote by $d_{[t]}\in\mathbb{N}_0^{J\times K\times t}$ the vector of demands realized until the end of time-slot t and let $\mathcal{D}_{[t]} = \prod_{\tilde{t}=1}^{t}\mathcal{D}_{\tilde{t}} \subseteq \mathbb{N}_0^{J\times K\times t}$ be the uncertainty set with respect to $d_{[t]}$, where $\prod$ denotes the Cartesian product. Furthermore, we denote by $z_{[t]}$ the vector containing the shipment quantities from different warehouses (local and central) to customers for different items from the beginning of the period up to the end of time-slot t. Besides, we denote by $0_n\in\mathbb{R}^n$ the vector consisting of all zeros, and we set $z_{[0]}(d_{[0]}) := 0_{JK}$. Then, the ARO formulation of the problem with T time-slots becomes

$$LT(\mathcal{D}_{[T]}) := \min_{y,S} \quad \sum_{i\in\mathcal{I}} f_i y_i + \sum_{\substack{i\in\mathcal{I}\\ k\in\mathcal{K}}} h_{ik}S_{ik} + \max_{d_{[1]}\in\mathcal{D}_{[1]}} Q_1(y,S,d_{[1]},0_{JK}) \tag{3}$$

$$
\begin{aligned}
\text{s.t.} \quad & S_{ik} \le M y_i, && \forall i \in \mathcal{I}, k \in \mathcal{K}, \\
& S_{ik} \in \mathbb{N}_0, && \forall i \in \mathcal{I}, k \in \mathcal{K}, \\
& y_i \in \{0,1\}, && \forall i \in \mathcal{I},
\end{aligned}
$$

where $\mathcal{D}_{[t]} = \prod_{\tilde{t}=1}^{t} \mathcal{D}_{\tilde{t}}$, and for $t = 1, ..., T-1$,

$$
\begin{aligned}
& Q_t\left(y, S, d_{[t]}, z_{[t-1]}(d_{[t-1]})\right) = \\
& \min_{z_t(d_{[t]})} \sum_{\substack{i \in \mathcal{I} \cup \{0\} \\ j \in \mathcal{J} \\ k \in \mathcal{K}}} c_{ijk} z_{ijkt}(d_{[t]}) + \sum_{\substack{i \in \mathcal{I} \\ k \in \mathcal{K}}} h_{ik} \left(S_{ik} - \sum_{j \in \mathcal{J}} \sum_{\tilde{t}=1}^{t} z_{jik\tilde{t}}(d_{[\tilde{t}]}) \right) \\
& \qquad + \max_{d_{t+1} \in \mathcal{D}_{[t+1]}} Q_{t+1}\left(y, S, d_{[t+1]}, z_{[t]}(d_{[t]})\right) \\
& \quad \text{s.t.} \quad \sum_{j \in \mathcal{J}} z_{ijkt}(d_{[t]}) \le S_{ik} - \sum_{j \in \mathcal{J}} \sum_{\tilde{t}=1}^{t-1} z_{jik\tilde{t}}(d_{[\tilde{t}]}), && \forall i \in \mathcal{I}, k \in \mathcal{K}, \\
& \qquad \sum_{i \in \mathcal{I} \cup \{0\}} z_{ijkt}(d_{[t]}) = d_{jkt}, && \forall j \in \mathcal{J}, k \in \mathcal{K}, \\
& \qquad z_{ijkt}(d_{[t]}) \in \mathbb{N}_0, && \forall i \in \mathcal{I} \cup \{0\}, j \in \mathcal{J}, k \in \mathcal{K},
\end{aligned} \tag{4}
$$

and

$$
\begin{aligned}
& Q_T(y, S, d_{[T]}, z_{[T-1]}(d_{[T-1]})) = \\
& \min_{z_T(d_{[T]})} \sum_{\substack{i \in \mathcal{I} \cup \{0\}, \\ j \in \mathcal{J}, \\ k \in \mathcal{K}}} c_{ijk} z_{ijkT}(d_{[T]}) \\
& \quad \text{s.t.} \quad \sum_{j \in \mathcal{J}} z_{ijkT}(d_{[T]}) \le S_{ik} - \sum_{j \in \mathcal{J}} \sum_{t=1}^{T-1} z_{ijkt}(d_{[t]}), && \forall i \in \mathcal{I}, k \in \mathcal{K}, \\
& \qquad \sum_{i \in \mathcal{I} \cup \{0\}} z_{ijkT}(d_{[T]}) = d_{jkT}, && \forall j \in \mathcal{J}, k \in \mathcal{K}, \\
& \qquad z_{ijkT}(d_{[T]}) \in \mathbb{N}_0, && \forall i \in \mathcal{I} \cup \{0\}, j \in \mathcal{J}, k \in \mathcal{K}.
\end{aligned} \tag{5}
$$

Figure 2 shows how the costs functions $Q_t(y, S, d_{[t]}, z_{[t-1]}(d_{[t-1]}))$, $t = 1, ..., T$, can be interpreted as part of the worst-case total cost. In Robust Optimization context, y_i and S_{ik}, $i \in \mathcal{I}$, $k \in \mathcal{K}$, are called "here-and-now" variables since the decision maker should choose their values before the realization of the demand. The other variables are called "wait-and-see", on which the decision maker should decide after the realization of the demand.

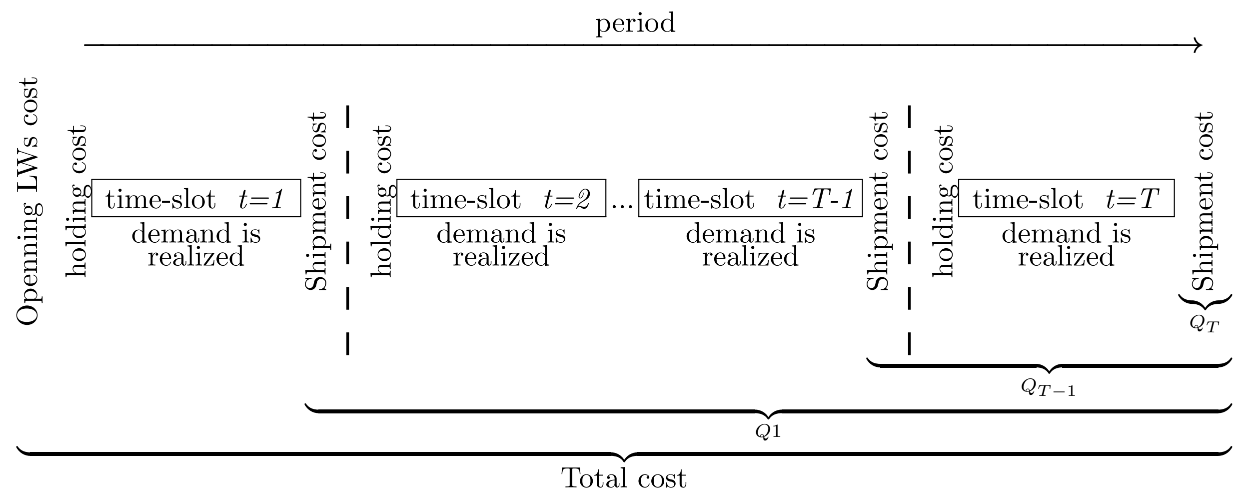


Figure 2 **Illustration of the total cost that is split into the costs in each time-slot. In this figure, "LW" stands for "local warehouse".**

In this paper, we denote the deterministic problem, where the demand vector $d \in \mathbb{N}_0^{J\times K\times T}$ during the time horizon is exactly known at the beginning of the time horizon, by $\underline{Q}(d)$ (see Appendix A for the explicit formulation, and conditions under which the solution is trivial).

4. Exact Methods to Solve the ARO Formulation

The main focus of this paper is on how problem (3) can be solved. In this section, we provide theoretical results with which we prove tractability of (3) for some classes of uncertainty sets. In other words, we show for some classes of uncertainty sets that the ARO formulation (3) can be solved by solving a polynomial number of deterministic LT problems. In the rest of this section, we first show the main theoretical results of the paper and then provide methods to solve (3) exactly using the theoretical results.

4.1. Convexity and nondecreasingness of $Q(y,S,d)$ in d

We first show that the function $Q(y,S,d)$, defined in (1), is convex in d, where the convexity over $\mathbb{N}_0^{JK}$ is defined as follows: for any $d^1, d^2 \in \mathbb{N}_0^{JK}$ and $\lambda \in (0,1)$

$$Q(y,S,\lambda d^1 + (1-\lambda)d^2) \le \lambda Q(y,S,d^1) + (1-\lambda)Q(y,S,d^2), \quad \text{if } \lambda d_1 + (1-\lambda)d_2 \in \mathbb{N}_0^{JK}.$$

Theorem 1. *Let us assume that $T=1$. Also, let $y \in \{0,1\}^I$ and $S \in \mathbb{N}_0^{IK}$ be given. Then, $Q(y,S,d)$, defined in (1), is convex in $d \in \mathbb{N}_0^{JK}$.*

Proof. Let us denote by $\tilde{Q}(y,S,d)$ the continuous relaxation of $Q(y,S,d)$, where constraint (1d) is replaced with $z_{ijk1(d)} \geq 0$. As discussed in Section 10.2 by Bazaraa et al. (2011), any basic feasible solution of $\tilde{Q}(y,S,d)$ corresponds to a totally unimodal submatrix of the coefficient matrix. Therefore, the vertices of the feasible region of $\tilde{Q}(y,S,d)$ are integral if $d \in \mathbb{N}_0^{JK}$. Hence, if $d \in \mathbb{N}_0^{JK}$, then there exists an optimal solution of $\tilde{Q}(y,S,d)$ that is optimal for $Q(y,S,d)$, and we have $\tilde{Q}(y,S,d) = Q(y,S,d)$.

Let $d^1, d^2 \in \mathcal{N}_0^{JK}$, $\lambda \in (0,1)$ such that $\lambda d_1 + (1-\lambda) d_2 \in \mathbb{N}_0^{JK}$. Then, by Lemma EC.2 of Ardestani-Jaafari and Delage (2016), we have

$$\tilde{Q}(y,S,\lambda d^1 + (1-\lambda)d^2) \leq \lambda \tilde{Q}(y,S,d^1) + (1-\lambda)\tilde{Q}(y,S,d^2).$$

Now, since $\lambda d^1 + (1-\lambda)d^2 \in \mathbb{N}_0^{JK}$, we have

$$Q(y,S,\lambda d^1 + (1-\lambda)d^2) \leq \lambda Q(y,S,d^1) + (1-\lambda)Q(y,S,d^2).$$

□

Similar to the proof of Theorem 1, one can show that for any $t \in \mathcal{T}$, $Q_t(y,S,d_{[t]}, z_{t-1}(d_{[t-1]}))$ is convex in $d_{[t]}$. Hence, in each time-slot, the worst-case scenario is among the vertices of the uncertainty set. Therefore, for uncertainty sets with a polynomial number of vertices, problem (3) is tractable.

Another important use of Theorem 1 is in reformulating (3) as an LT problem where the demands as well as the shipment quantities are continuous.

Proposition 1. *A point (y^*, S^*) is an optimal solution of (3) if and only if (y^*, S^*) is the optimal solution of the continuous relaxation of the LT problem, where the integrality restriction on the shipment policies is relaxed and the uncertainty set is replaced with its convex hull.*

Proof. Let us denote the optimal value of the continuous relaxation by $\tilde{LT}(\mathcal{D}_{[T]})$. Theorem 1 implies that replacing $\mathcal{D}_t$ with its set of vertices, denoted by Ω_t, results in an equivalent problem. Let us denote by $\tilde{Q}_t$ the continuous relaxation of Q_t. For any vector d in $\prod_{t=1}^T \Omega_t$, as shown in the proof of Theorem 1, the optimal solution of $\tilde{Q}_t(y, S, d_{[t]}, z_{t-1}(d_{[t-1]}))$ is integer-valued. Hence, we have

$$LT\left(\mathcal{D}_{[T]}\right) \equiv LT\left(\prod_{t=1}^{T}\Omega_t\right) \equiv \tilde{LT}\left(\prod_{t=1}^{T}\Omega_t\right) \equiv \tilde{LT}(conv(\mathcal{D}_{[T]})),$$

where $\equiv$ means the problems have the same optimal value and optimal "here-and-now" solutions, $conv(\mathcal{D}_{[T]})$ is the convex hull of $\mathcal{D}_{[T]}$, and the right most equality holds because of Lemma EC.2 of Ardestani-Jaafari and Delage (2016). □

Proposition 1 asserts that continuous relaxation of (3) can be used to obtain the optimal value as well as the "here-and-now" part of optimal solutions. Therefore, exact methods for ARO problems with continuous uncertain parameters and "wait-and-see" variables can be employed to solve the continuous relaxation of (3); see e.g., Bertsimas et al. (2010), Bertsimas and Bidkhori (2015), Zhen et al. (2018).

Another important property of $Q(y, S, d)$, next to its convexity in d, is that it is nondecreasing in d for a class of problems. We show this in the following theorem.

THEOREM 2. *Let us assume that $T = 1$. Let us fix $y_i \in \{0, 1\}$ and $S_{ik} \in \mathbb{N}_0$, for all $i \in \mathcal{I}$, $k \in \mathcal{K}$. For a given uncertainty set $\mathcal{D}_1 \subseteq \mathbb{N}_0^{JK}$, if $d^1, d^2 \in \mathcal{D}$ be such that $d^1 \lneqq d^2$, then $Q(y, S, d^1) \leq Q(y, S, d^2)$.*

Proof. Clearly, due to the ample stock of the central warehouse, (1) is alway feasible. Let us denote by $z^*_{ijk1}(d)$ an optimal shipment policy of (1), given the demand vector d, location variable y_i, and stock level S_{ik}, $i \in \mathcal{I}$, $k \in \mathcal{K}$. By contradiction, let us assume that

$$Q(y, S, d_1) = \sum_{i,j,k} c_{ijk} z^*_{ijk1}(d^1) > \sum_{i,j,k} c_{ijk} z^*_{ijk1}(d^2) = Q(y, S, d_2).$$

We prove the theorem by constructing a new shipment policy to satisfy the demand d^1 with less cost than $z^*_{ijk1}(d^1)$, and reach the contraction with optimality of $z^*_{ijk1}(d^1)$. To this end, let us set

$$\overline{\mathcal{JK}} := \left\{(j, k) : \quad d^1_{jk1} < d^2_{jk1}\right\}.$$

Since $d^1 \lneqq d^2$, $\overline{\mathcal{JK}}$ is not empty. Furthermore, let $\{i_0, i_1, ..., i_I\} := \mathcal{I} \cup \{0\}$, and for any $j \in \mathcal{J}$, $k \in \mathcal{K}$, set

$$\mathcal{I}(j,k) := \left\{ i \in \mathcal{I} \cup \{0\} : \ z^*_{ijk1}(d^2) > 0 \right\}.$$

For any $(j,k) \in \overline{\mathcal{JK}}$ and $\ell = 0, ..., I$, let us set

$$a_{j,k,\ell+1} := \max\left\{ a_{j,k,\ell} - z^*_{i_\ell jk1}(d^2), 0 \right\},$$

and $a_{j,k,0} = d^2_{jk} - d^1_{jk}$. We construct the following solution:

$$z^{new}_{ijk1}(d^1) = \begin{cases} z^*_{ijk1}(d^2) & \text{if } (j,k) \notin \overline{\mathcal{JK}} \\ \max\left\{ z^*_{ijk1}(d^2) - a_{j,k,\ell}, 0 \right\} & \text{if } (j,k) \in \overline{\mathcal{JK}}, i = i_\ell. \end{cases}$$

Clearly, for any $i \in \mathcal{I}$, $j \in \mathcal{J}$, and $k \in \mathcal{K}$, we have $z^{new}_{ijk1}(d^1) \leq z^*_{ijk1}(d^2)$. Hence, constraint (1b) and (1d) are satisfied. Let us set for any $(j,k) \in \overline{\mathcal{JK}}$

$$\tilde{\ell}_{jk} = \min\left\{ \ell : \ \sum_{\bar{\ell}=0}^{\ell} z^*_{i_{\bar{\ell}} jk1}(d^2) \geq d^2_{jk1} - d^1_{jk1} \right\}.$$

In other word, $\tilde{\ell}_{jk}$ is the smallest index where $a_{j,k,\tilde{\ell}_{jk}+1} = 0$. It is straightforward to check that

$$z^{new}_{i_\ell jk1}(d^1) = 0, \ \ \forall \ell < \tilde{\ell}_{jk}.$$

Now, we show that (1c) holds as well. By construction, we have for any $(j,k) \notin \overline{\mathcal{JK}}$

$$\sum_{i \in \mathcal{I} \cup \{0\}} z^{new}_{ijk1}(d^1) = \sum_{i \in \mathcal{I} \cup \{0\}} z^*_{ijk1}(d^2) = d^2_{jk1} = d^1_{jk1},$$

where the last equality is because of the definition of $\overline{\mathcal{JK}}$. For $(j,k) \in \overline{\mathcal{JK}}$, we have

$$\begin{aligned} \sum_{i \in \mathcal{I} \cup \{0\}} z^{new}_{ijk1}(d^1) &= \sum_{\ell = \tilde{\ell}_{jk}}^{I} \max\left\{ z^*_{i_\ell jk1}(d^2) - a_{j,k,\ell}, 0 \right\} \\ &= z^*_{i_{\tilde{\ell}_{jk}} jk1}(d^2) - a_{j,k,\tilde{\ell}_{jk}} + \sum_{\ell = \tilde{\ell}_{jk}+1}^{I} z^*_{i_\ell jk1}(d^2) \\ &= \sum_{\bar{\ell}=0}^{\tilde{\ell}_{jk}} z^*_{i_{\bar{\ell}} jk1}(d^2) - a_{j,k,0} + \sum_{\ell = \tilde{\ell}_{jk}+1}^{I} z^*_{i_\ell jk1}(d^2) \qquad (6) \\ &= \sum_{\ell=0}^{I} z^*_{i_\ell jk1}(d^2) - a_{j,k,0} = d^2_{jk1} - a_{j,k,0} = d^1_{jk1}, \end{aligned}$$

where the equality in (6) is because of the construction of $a_{j,k,\ell}$, $\ell = 1, ..., I$.

Now, since for any $i \in \mathcal{I}$, $j \in \mathcal{J}$, and $k \in \mathcal{K}$, we have $z^{new}_{ijk1}(d^1) \le z^*_{ijk1}(d^2)$, and $\sum_{i,j,k} c_{ijk} z^*_{ijk1}(d^1) > \sum_{i,j,k} c_{ijk} z^*_{ijk1}(d^2)$, we have $\sum_{i,j,k} c_{ijk} z^*_{ijk1}(d^1) > \sum_{i,j,k} c_{ijk} z^{new}_{ijk1}(d^1)$.

So, $z^{new}_{ijk1}(d^1)$, $i \in \mathcal{I} \cap \{0\}$, $j \in \mathcal{J}$, $k \in \mathcal{K}$ is a feasible shipment policy that is cheaper than the optimal policy $z^*_{ijk1}(d^1)$, $i \in \mathcal{I} \cap \{0\}$, $j \in \mathcal{J}$, $k \in \mathcal{K}$, which leads to a contradiction. □

We emphasize that this result holds also for problems with continuous demand where the shipment policies are continuous (with a similar proof). This theorem asserts that $Q(y, S, d)$ is a nondecreasing function in d. In the next section, we show how to use convexity and nondecreasingness of $Q(y, S, d)$ to solve (3) exactly.

4.2. Practicality of convexity and nondecreasingness of $Q(y, S, d)$

In this section, we show how the results in Section 4.1 are employed to solve classes of LT problems. As mentioned, one class of LT problems shown to be tractable is the ones where the uncertainty set has a polynomial number of vertices. In the rest of this section, we show how the results in Section 4.1 imply tractability of LT problems (3) with two typically used uncertainty sets.

LT problems with hyper-box uncertainty sets: A hyper-box uncertainty set is defined as $\mathcal{D}_t = \mathbb{N}_0^{J\times K} \cap \prod_{\substack{j\in\mathcal{J}\\ k\in\mathcal{K}}} [\underline{d}_{jkt}, \overline{d}_{jkt}]$, where $\overline{d}_{jkt} \in \mathbb{N}$, and have been studied extensively in the literature of Robust Optimization problems in general (El-Amine et al. 2017, Yang et al. 2012, Yanıkoğlu et al. 2019, Zhen et al. 2018, Buhayenko and den Hertog 2017) and LT problems in specific (Ardestani-Jaafari and Delage 2017, Solyalı et al. 2015). In the next corollary, we show how our results imply that (3) with a hyper-box uncertainty set is tractable.

Corollary 1. *Let us assume that* $\mathcal{D}_t = \mathbb{N}_0^{J\times K} \cap \prod_{\substack{j\in\mathcal{J}\\ k\in\mathcal{K}}} [\underline{d}_{jkt}, \overline{d}_{jkt}]$, *where* $\overline{d}_{jkt} \in \mathbb{N}$, *and* $c_{ijk} \ge h_{ik}$, $j \in \mathcal{J}$, $k \in \mathcal{K}$, $t \in \mathcal{T}$. *Then, problem* (3) *is equivalent to the deterministic problem* $\underline{Q}(\overline{d})$, *where* $\overline{d} = [\overline{d}_{jkt}]_{\substack{j\in\mathcal{J}\\ k\in\mathcal{K}\\ t\in\mathcal{T}}}$. *In other words,* (y^*, S^*) *is an optimal solution of* (3) *if and only if it is optimal for* $\underline{Q}(\overline{d})$.

Proof. See Appendix B. □

Using this corollary, one can solve LT problems with hyper-box uncertainty sets exactly by solving the equivalent deterministic problem, which can be solved in a matter of minutes for real-life sized problem using existing solvers.

4.2.1. LT problems with budget uncertainty sets Another important class of uncertainty sets is budget uncertainty sets; $\mathcal{D}_t = \prod_{k\in\mathcal{K}} \mathcal{D}_{kt}$, $t \in \mathcal{T}$, where

$$\mathcal{D}_{kt} = \left\{ d_{jkt} \in \mathbb{N}_0 : \; \underline{d}_{jkt} \le d_{jkt} \le \overline{d}_{jkt}, \;\; \forall j \in \mathcal{J}, \; \sum_{j\in\mathcal{J}} d_{jkt} \le \Gamma_{kt} \right\},$$

and where $\Gamma_{kt}, \underline{d}_{jkt}, \overline{d}_{jkt} \in \mathbb{N}_0$, for all $j \in \mathcal{J}$, $k \in \mathcal{K}$, and $t \in \mathcal{T}$. Budget uncertainty sets are introduced by Bertsimas and Sim (2004) to decrease the conservativeness of the robust solutions obtained by using hyper-box uncertainty set, and extensively used (Housni and Goyal 2018, Gabrel et al. 2014a, Atamtürk and Zhang 2007, Ben-Tal et al. 2011).

Using Theorem 2, it can be seen that we can shrink the budget uncertainty set in (3) and only consider the scenarios on the cut. In other words, we have the following corollary.

Corollary 2. *Let us assume that $c_{ijk} \ge h_{ik}$, $j \in \mathcal{J}$, $k \in \mathcal{K}$, $t \in \mathcal{T}$, and $\mathcal{D}_t$, $t \in \mathcal{T}$, is a budget uncertainty set. Then,*

$$LT\left(\prod_{t\in\mathcal{T}} \mathcal{D}_t\right) = \tilde{LT}\left(\prod_{t\in\mathcal{T}} \bar{\mathcal{D}}_t\right)$$

where $\tilde{LT}(.)$ refers to the worst-case objective value of the continuous relaxation of (3), and $\bar{\mathcal{D}}_t = \prod_{k\in\mathcal{K}} \bar{\mathcal{D}}_{kt}$, $t \in \mathcal{T}$,

$$\bar{\mathcal{D}}_{kt} = \left\{ d_{jkt} \in \mathbb{N}_0 : \; \underline{d}_{jkt} \le d_{jkt} \le \overline{d}_{jkt}, \;\; \forall j \in \mathcal{J}, \; \sum_{j\in\mathcal{J}} d_{jkt} = \Gamma_{kt} \right\}. \tag{7}$$

Proof. The result follows analogously to Proof 1 of Corollary 1. □

Corollary 2 assets that the optimal robust decisions on the location of the local warehouses as well as their basestock levels can be made by only being safe-guarded against the demand scenarios in $\prod_{t\in\mathcal{T}} \bar{\mathcal{D}}_t$ instead of $\prod_{t\in\mathcal{T}} \mathcal{D}_t$. In general, it is proved by O'Neil (1971) that the number of vertices of $\bar{\mathcal{D}}_{kt}$ is at most in the order of $\mathcal{O}(J^J)$. Hence, for problems where $\prod_{t\in\mathcal{T}} \bar{\mathcal{D}}_t$ has a polynomial number of vertices, the optimal “here-and-now” decisions can be obtained by solving a polynomial number of deterministic problems.

For a general budget uncertainty set, as proved by Atamtürk and Zhang (2007), problem (3) is intractable. However, one can use Corollary 2 and the efficient algorithm proposed by Lara et al. (2009) to solve (3) to optimality.

Hitherto, we have shown how to solve (3) with hyper-box and budget uncertainty sets. In the next section, we focus on the approximate methods considering general bounded uncertainty sets $\mathcal{D}_t$, $t \in \mathcal{T}$. To do this, we eliminate the equality constraints in (3) and the variables corresponding to the shipments from the central warehouse to the customers, as it is always difficult to deal with equality constraints in an ARO problem (see the discussion by Marandi and den Hertog (2018)). For the explicit formulation of the problem after the elimination, we refer the reader to Appendix C. We emphasize that all the theoretical results in this paper hold for the new formulation.

5. Approximate Methods to Solve the ARO Formulation

Methods in the literature to approximate an ARO problem containing integer wait-and-see variables can be categorized into two classes. The first class contains methods where the approximation is done by restricting the integer wait-and-see variables to be piece-wise constant functions in the uncertain parameter on the uncertainty set (Romeijnders and Postek 2018, Subramanyam et al. 2019, Bertsimas and Dunning 2016, Postek and den Hertog 2016, Zhao and Zeng 2012). The idea behind the methods in this class is to iteratively partition the convex uncertainty set into smaller convex subsets in a disciplined way. Even though we can replace the uncertainty set with its convex hull in the ARO problem (3) after the elimination, finding the explicit formulation of the convex hull of the subsets may be inefficient. Therefore, such methods are not applicable to problem (3).

The second class contains methods that restrict integer wait-and-see variables to be (piece-wise) affine functions; see e.g., Bertsimas and Caramanis (2007), and Lappas and Gounaris (2016). In this section, we first provide new insights for this class of methods and contribute to it and then provide a new method to approximate the class of ARO problems that has the convexity characteristic.

5.1. Integer affine decision rule

The state-of-the-art method to approximate an ARO problem with continuous wait-and-see variables is based on affine decision rules (Ben-Tal et al. 2004) and piece-wise affine decision rules (Chen and Zhang 2009). The idea behind affine decision rules is to restrict the wait-and-see variables to be affine in the uncertain parameters. This type of approximations is known to perform well on the continuous relaxation of (3), where the integrality restriction over the demand and the shipment policies are relaxed (see, e.g., Ardestani-Jaafari and Delage (2017)). However, such approximations are not applicable to (3). To see this, let us fix $i \in \mathcal{I}$, $j \in \mathcal{J}$, $k \in \mathcal{K}$, $t \in \mathcal{T}$, and restrict $z_{ijkt}(d_{[t]})$ to be affine in $d_{[t]}$, hence $z_{ijkt}(d_{[t]}) = u_{jikt} + \left\langle V^{jikt}, d_{[t]} \right\rangle$, where $u_{jikt} \in \mathbb{R}$, $V^{jikt} \in \mathbb{R}^{J\times K\times t}$, and $\langle A, B\rangle = \sum_{h=1}^{H} \sum_{\ell=1}^{L} \sum_{n=1}^{N} A_{h\ell n} B_{h\ell n}$, for $A, B \in \mathbb{R}^{H\times L\times N}$. Clearly, there is no guarantee that the value of $u_{jikt} + \left\langle V^{jikt}, d_{[t]} \right\rangle$ is integer even with an integer-valued demand vector $d_{[t]}$.

To resolve this issue, Bertsimas and Caramanis (2007) propose to restrict the wait-and-see variables to be affine functions with integer-valued coefficient vectors and constant terms. More specifically, based on their approximation, $z_{ijkt}(d_{[t]})$ is restricted to be in the form of $u_{jikt} + \left\langle V^{jikt}, d_{[t]} \right\rangle$, where $u_{jikt} \in \mathbb{N}_0$, $V^{jikt} \in \mathbb{N}_0^{J\times K\times t}$. This method guarantees that the resulting policy generates only integer values and provides an affine decision rule for an integer wait-and-see variable, which we call integer ADR. In the following proposition, we show that this is the only way of restricting an integer wait-and-see variable to be affine in the integer uncertain parameter.

Proposition 2. *The function $f : \mathbb{Z}^n \to \mathbb{Z}^m$ is affine if and only if $f(x) = Ax + b$, where $A \in \mathbb{Z}^{m\times n}$, and $b \in \mathbb{Z}^m$.*

Proof. The "if" part is trivial. We will proof the "only if" part.

Let function $f : \mathbb{Z}^n \to \mathbb{Z}^m$ be affine. Then, f is clearly a summation of a linear function $g : \mathbb{Z}^n \to \mathbb{R}^m$ and a constant $b \in \mathbb{R}^m$. Since g is linear, then $g(0_n) = 0_m$, where 0_n is the origin of $\mathbb{Z}^n$. Therefore,

$$b = f(0) - g(0) = f(0) \in \mathbb{Z}^m.$$

This implies that $g:\mathbb{Z}^n\to\mathbb{Z}^m$. Now, let $x\in\mathbb{Z}^n$. Then, $x=\sum_{i=1}^n x_ie_i$, where $e_i\in\mathbb{R}^n$ is a vector of all zeros except in the i^{th} entry, which is one. Since g is a linear function, we have

$$g(x)=\sum_{i=1}^n x_ig(e_i)=\underbrace{\begin{bmatrix} g(e_1)\ \dots\ g(e_n)\end{bmatrix}}_{A:=}\begin{bmatrix} x_1\\ \vdots\\ x_n\end{bmatrix}.$$

Since the range of g is a subset of $\mathbb{Z}^m$, so $A\in\mathbb{Z}^{m\times n}$, which completes the proof. □

Proposition 2 states that any integer-valued affine policy in an integer uncertain parameter with an uncertainty set containing standard simplex should be in the form of the discrete ADR.

To be able to use the conventional reformulation after applying the discerete ADR, the uncertainty set needs to be convex. Similar to the proof of Theorem 1, one can show that the uncertainty set can be replaced by its convex hull after applying the discrete ADR, as the worst-case scenarios are among the vertices. Therefore, one can approximate (3) in three steps: first, the equality constraints need to be eliminated, then the discrete ADR can be applied to the reformulated problem to construct the static robust approximation, and finally, the uncertainty set can be replaced with its convex hull in order to use the conventional methods in Robust Optimization to solve the static robust approximation (see, e.g., Gorissen et al. (2015) for how we can solve the approximation).

5.2. A Simplex-type method to obtain a lower bound

In this paper, we assume that the uncertainty set is bounded. Hence, it has finite number of vertices and its convex hull is a polytope (see, e.g, Theorem 6 by Dey and Morán (2013)). So, we let the convex hull to be in the form

$$\mathcal{D}=\{d:\in\mathbb{R}^{JKT}: Ad=b,\ \ d\geq 0\},$$

where $A\in\mathbb{R}^{m\times JKT}$ and $b\in\mathbb{R}^m$. In this section, we propose a new method to generate a lower bound on $LT(\mathcal{D}_{[T]})$, inspired by the simplex method (see Chapter 3 of the book by Bazaraa et al. (2011)). As we have seen in the proof of Proposition 2, the worst-case scenario occurs in a vertex of $\mathcal{D}$. So, we randomly start from a vertex and check the objective values of the deterministic LT problems with the demand vector being the ones adjacent to the selected vertex. In case of an improvement in the objective value, we move to that vertex and continue the procedure until we see that no improvement can be achieved by moving to the adjacent vertices. In this case, we are in a local optimum.

To start from a random vertex, we randomly generate a vector $e \in [-1,1]^{JKT}$ and solve the optimization problem $\min_{d\in\mathcal{D}} e^T d$. Let us denote by d^0 the optimal vertex of the linear programming problem. Then, similar to the simplex method, we denote by B and N the basis and non-basis sub-matrices of A corresponding to d^0. Then, increasing any non-basis variable x_k to

$$\frac{\bar{b}_r}{y_{rk}} = \min_{i=1,\dots,m} \left\{ \frac{\bar{b}_i}{y_{ik}} : \; y_{ik} > 0 \right\},$$

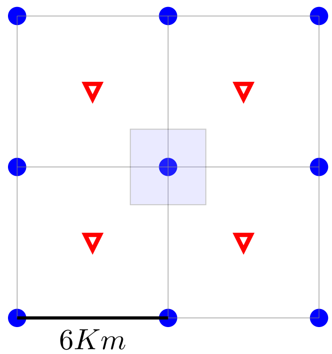
and updating the basis variables x_{B_i} to $\bar{b}_i - \frac{\bar{b}_r}{y_{rk}}$, where $\bar{b} = B^{-1}b$, results in moving to an adjacent vertex. Due to the similarity of this method to Simplex, we refer to it as the *Simplex-type* algorithm.

6. Numerical Experiment

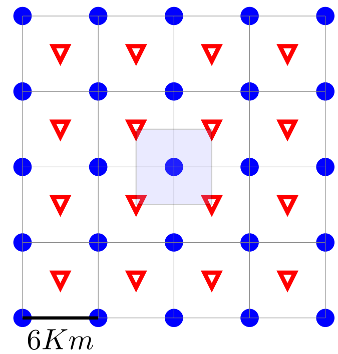
In this section, we illustrate the effectiveness of the exact solution method, proposed in Section 4, as well as the approximate method, proposed in Section 5, in solving robust LT problems. The numerical results of this work were carried out on the Dutch national e-infrastructure with the support of SURF Cooperative. More specifically, we used a virtual machine featuring 8 processors 2.30 GHz and 20.00 GB RAM running Julia 1.0.3 (Bezanson et al. 2017). We use JuMPeR 0.6 (see Chapter 6 of Dunning (2016)) to reformulate the static robust optimization problems into a mixed integer linear programming (MILP) problem, and use JuMP 0.18.5 (Dunning et al. 2017) to pass MILP problems to IBM ILOG CPLEX 12.7.1. To have a fair comparison, the time report in this section contains the time needed to construct the model, pass it to the solver, and the time taken by the solver to solve the optimization problem. For all the methods, the time limit of 7200 seconds has been considered.

6.1. Testbed

To compare different solution methods, we randomly generate two types of LT instances to optimally design the distribution network of an e-commerce company. The decisions are in finding the



(a) Small-sized instances



(b) Medium-sized instances

Figure 3 **Illustration of the square cities considered in the numerical experiments. The blue dots are the customers and the red triangles are the candidate locations to open distribution centers. The light blue** $6Km \times 6Km$ **square shows the center of the city.**

optimal locations of warehouses and their inventory of each product while considering the optimal shipment policies to the customers. The types of instances include small-sized instances generated for a $12Km \times 12Km$ square city with $J = 9$ customers and one product valued €50, and medium-sized instances generated for a $24Km \times 24Km$ square city with $J = 25$ customers and two products valued €50 and €200. The candidate locations to open warehouses are the middle points of the $6Km \times 6Km$ squares. Figure 3 illustrates the locations of the customers (blue circles) and candidate locations to open warehouses (red triangles). Each location has the two-dimensional coordinate based on the grid. For example, in the small-sized instances the coordinate of the customer in the south-west is $\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]$, and the candidate location close to it has the coordinate $\left[\begin{smallmatrix}3\\3\end{smallmatrix}\right]$.

The company offers a same-day delivery service. It specifies that customers can place their orders in the following time-slots of a day: from 20:00 at the previous evening up to 10:00, 10:00 upto 12:00, 12:00 upto 14:00, 14:00 upto 16:00, 16:00 upto 18:00, and 18:00 upto 20:00; i.e., $T = 6$. An order placed in a time-slot has to be delivered within an hour after the end of the time-slot ($t_{jk}^{max} = 1h$). In case of tardiness, a unit penalty of 20% of the product value per hour delay is paid to the customer. Thus, $c_{jk}^{p} = 0.2 v_k$, where v_k is the product value.

To ship the orders to the customers from the central warehouse, the company is charged by a third party at the fixed cost of €50 per item ($c_{0jk} = €50$, for customer j per item of product

k). One way of reducing the shipment cost is by opening warehouses in the city. The cost of opening a warehouse is $69,350\frac{€}{\text{year}}$ in a location outside of the center of the cities (highlighted by light blue square in Figure 3), and $142,350\frac{€}{\text{year}}$ in the center of the the large city. Moreover, the inventory holding cost per item of each product is the same in different warehouses and equals 20% of the product value per year. So, the unit inventory holding cost for location i and product k is $h_{ik} = \frac{0.2v_k}{365\times 6}$, per time-slot, where v_k is the value of product k.

To ship the items from the warehouses in the city to the customers, the company plans to use couriers at the cost of $9\frac{€}{hour}$ who travel at an average speed of $18\frac{Km}{hour}$. Per customer order, a separate trip is made by a courier. As the city is designed in blocks, the couriers can only travel to the East, West, North, or South. Therefore, to calculate the distance between the warehouse in coordinate $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ and the customer in coordinate $\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$, Manhattan distance is used, which is defined as

$$Dist\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}\right) = |x_1 - y_1| + |x_2 - y_2|. \tag{8}$$

To replenish each warehouse at the end of each time-slot, the company uses a third party that charges a daily fixed cost of €10 (c_i^{rep}, $i \in \mathcal{I}$) and a unit transit cost of €0.5 (c_{ik}^{rep}, $i \in \mathcal{I}$, $k \in \mathcal{K}$). Therefore, the scaled total fixed cost for warehouses inside the center of the large city is

$$f_i = \frac{142,350 + 3650}{365} = 400 \quad \left(\frac{€}{day}\right),$$

and for the warehouses outside the center is

$$f_i = \frac{69,350 + 3650}{365} = 200 \quad \left(\frac{€}{day}\right).$$

Furthermore, the unit shipment cost from location i to customer j for product k per hour is

$$c_{ijk} = \underbrace{\frac{2 \times 9 \times Dist_{ij}}{18}}_{\text{courier cost}} + \underbrace{0.2v_k\left(\frac{Dist_{ij}}{18} - 1\right)^+}_{\text{tardiness penalty cost}} + \underbrace{0.5}_{\text{replenishment cost}} \quad \left(\frac{€}{item}\right),$$

where $Dist_{ij}$ is the Manhatan distance between location i and customer j. A summary of the parameters considered in the testbed is presented in Table 2. The company wants to use the

Parameter (units)	small-sized instances	medium-sized instances
I	4	16
J	9	25
K	1	2
T	6	6
f_i ($\frac{€}{day}$)	200	200, 400 (the ones in the center)
v_k ($\frac{€}{item}$)	50	50, 200
h_{ik} ($\frac{€}{item}$)	$9\times 10^{-5}v_k$	$9\times 10^{-5}v_k$
$Dist_{ij}$ (Km)	Manhatan distance defined in (8)	Manhatan distance defined in (8)
c_{ijk} ($\frac{€}{item}$)	$Dist_{ij}+0.2v_k\left(\frac{Dist_{ij}}{18}-1\right)^{+}+0.5$	$Dist_{ij}+0.2v_k\left(\frac{Dist_{ij}}{18}-1\right)^{+}+0.5$
c_{0jk} ($\frac{€}{item}$)	50	50

Table 2 Parameters used in the generated instances.

demand data on the last 100 days to make the decisions. To construct the demand data, we first fix $\hat{d}\in\{1,2,3,4\}$. Then, we randomly generate the demand rate $\tilde{d}\in[0,\hat{d}]^{JKT}$ from a uniform distribution and construct 100 data points $data_P$ from the Poisson distribution with rate $\tilde{d}$, $Pois(\tilde{d})$. We also construct another 100 data points $data_{\beta Bin}$ from the JKT dimensional beta binomial distribution with $2\hat{d}$ trials and both shape parameters being 0.25. We consider the demand data to be $[\frac{data_P+data_{\beta Bin}}{2}]$, where [.] is the rounding function. We have chosen this procedure to avoid having a demand vector with a specific distribution.

6.2. LT instances with hyper-box uncertainty set

Based on the dataset, we construct the uncertainty set using the empirical distribution of the demand of the customers. This is because all the results in the literature (Ning and You 2018, Campbell and How 2015) are designed for continuous uncertain parameters and not applicable to a LT problem studied in this paper. More precisely, for a given $\alpha\in[0,1]$, and $t\in\mathcal{T}$, we assume that the convex hull of the uncertainty set $\mathcal{D}_t$ is formulated as

$$conv(\mathcal{D}_t):=\left\{d_{kt}\in\mathbb{R}^J\,\middle|\,0\le d_{jkt}\le\bar{d}^{\alpha}_{jkt},\quad\forall j\in\mathcal{J},k\in\mathcal{K}\right\},\tag{9}$$

where $\bar{d}^{\alpha}_{jkt} \in \mathbb{N}$ is the upper bounds of the $(1-\alpha)$ confidence interval derived based on the empirical distribution of d_{jkt} for $j \in \mathcal{J}$, $k \in \mathcal{K}$, $t \in \mathcal{T}$. To see the impact of α in the final solution, we vary its value in $\{0.6, 0.65, 0.70, ..., 0.95\}$.

Corollary 1 shows that the optimal location of warehouses as well as their basestock policies can be obtained by solving the deterministic problem $\underline{Q}\left(\bar{d}\right)$, where $\bar{d} = \left[\bar{d}^{\alpha}_{jkt}\right]_{j\in\mathcal{J},\ k\in\mathcal{K},\ t\in\mathcal{T}}$, whereas the integer affine-decision rule (IADR) provides, in general, an upper bound on the worst-case cost. To apply the IADR, we first restrict the shipment policy from location i to customer j for product k at time-slot t to be dependent on the demands of different customers for different products in the time-slots up to t, as discussed by Ardestani-Jaafari and Delage (2017) for the continuous problem. In other words, for any $i \in \mathcal{I}$, $j \in \mathcal{J}$, $k \in \mathcal{K}$, and $t \in \mathcal{T}$, we check the performance of approximating $z_{ijkt}(d)$ by $\left\langle V_{jikt}, d_{[t]}\right\rangle + u_{jikt}$, where $V_{jikt} \in \mathbb{N}_0^{K\times J\times t}$, $u_{jikt} \in \mathbb{N}$, and $d_{[t]}$ is a vector containing all the demands up to the time-slot t. We find that such restriction results in extremely large and hence computationally expensive MILP problems, which are not possible to construct within the given time limitation. Therefore, from now on we consider the dependency of the shipment quantity from local warehouse i to customer j for product k at time-slot t on d_{jkt}. In other words, for any $i \in \mathcal{I}$, $j \in \mathcal{J}$, $k \in \mathcal{K}$, and $t \in \mathcal{T}$, we approximate $z_{ijkt}(d)$ by $V_{jikt}d_{jkt} + u_{jikt}$, where $V_{jikt} \in \mathbb{N}$ and $u_{jikt} \in \mathbb{N}$.

6.2.1. Results for small-sized instances For this class of instances, the solutions obtained by IADR coincide with the optimal solutions obtained by solving the deterministic problem $\underline{Q}\left(\bar{d}\right)$. Figure 4 illustrates total worst-case cost of the small-sized instances. As expected, two observations can be made from Figure 4. First, one can observe that the worst-case cost is increasing with respect to $\hat{d}$, which happens because increasing $\hat{d}$ results in an increase in the aggregated demand of customers. The second observation is that for a given $\hat{d}$, an increase in α results in an increase in the worst-case cost. This is because an increase in α can be interpreted as an increase in the proportion of aggregated worst-case demand satisfied by the company. Hence, the higher α the more demand needs to be met in the worst-case scenario, which results in a higher cost.

To have a better understanding of how an increase in α impacts the solution, we have illustrated the worst-case solution of the LT instances in Figure 5 for two values of α with $\hat{d}$ being 2, where

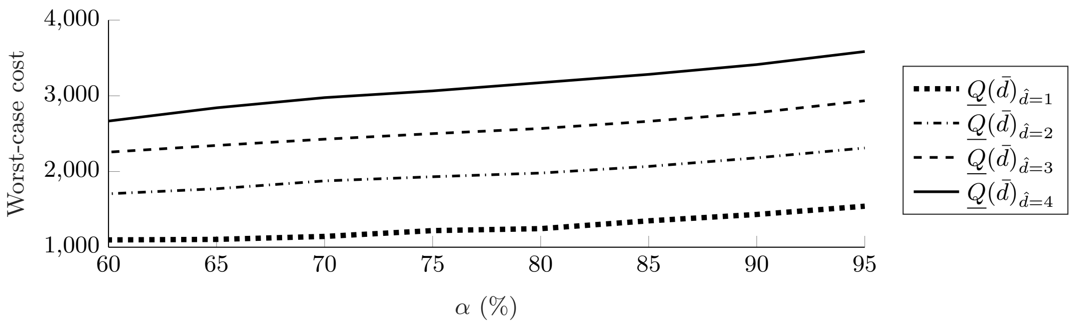
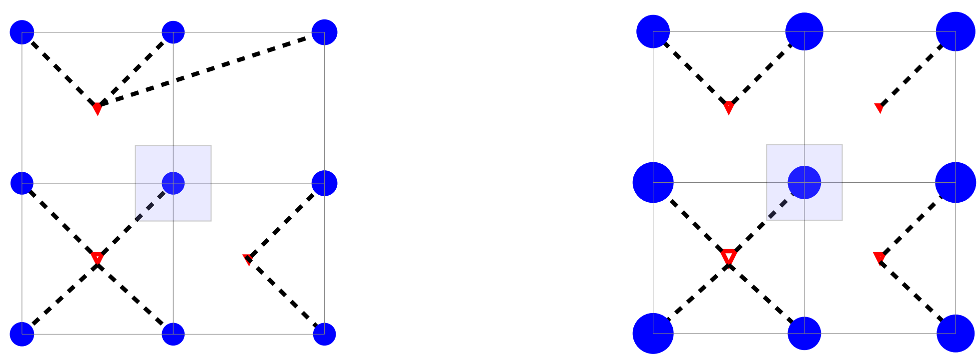


Figure 4 **The optimal worst-case cost obtained for the small-sized instances with hyper-box uncertainty sets. $\underline{Q}\left(\bar{d}\right)$ refers to the deterministic LT problem $\underline{Q}\left(\bar{d}\right)$, which provides the same worst-case cost as IADR.**



(a) $\alpha = 0.6$ (b) $\alpha = 0.9$

Figure 5 **Illustration of solutions for the small-sized instances with $\hat{d} = 2$ and a hyper-box uncertainty set for two values of α. The sizes of circles and triangles correspond to the aggregated worst-case demand for each customer over time and the total basestock level of each warehouse, respectively.**

sizes of triangles represent the total basestock policies of opened warehouses to satisfy the worst-case demand. In this figure, the sizes of circles corresponds to the worst-case aggregated demand of each customer in a day in the uncertainty set. Besides, the dashed lines show the optimal resource allocations to satisfy worst-case customers' demands.

As one can see, an increase in α results in an increase in the aggregated worst-case demand of each customer. For $\alpha = 0.6$, the demands are low and hence the courier costs to deliver the product from warehouses to customers are low and thus three warehouses are used to service the customers. However, for $\alpha = 0.9$, the increase in these costs are such that it is more beneficial for the company to open the forth warehouse and store more items. Hence, an increase in α results in an increase in the total number of items stored in warehouses.

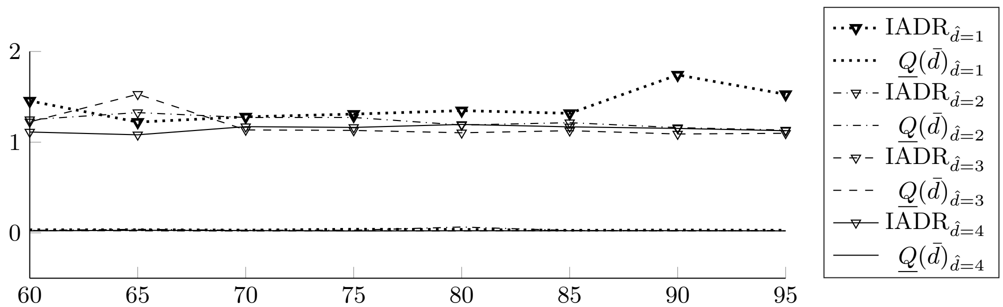


Figure 6 **The time needed to solve the small-sized instances with hyper-box uncertainty sets. IADR and $\underline{Q}(\bar{d})$ refer to the integer-valued affine decision rule and the deterministic LT problem, respectively.**

Next to the comparison of the objective values, the time needed to solve the problems with IADR or $\underline{Q}(\bar{d})$ is important. Figure 6 provides the comparison of times needed for both methods. As one can see, the deterministic LT problem $\underline{Q}(\bar{d})$ can be solved much faster than the MILP problem formulated using IADR, which has many more variables and constraints. In particular, to reformulate the static robust optimization problem after applying IADR to an MILP problem, one first needs to introduce two extra integer variables V_{ijkt} and u_{ijkt} per candidate location i, customer j, time-slot t, and product k. Then, the robust counterpart of each constraint needs to be formulated using extra constraints and variables. Hence, the size of the MILP problem is much higher than the deterministic LT problem $\underline{Q}(\bar{d})$. Therefore, it is almost 6 times faster to obtain the optimal solution using $\underline{Q}(\bar{d})$ instead of using IADR.

6.2.2. Results for Meduim-sized instances For this class of instances, the MILP problems formulated after applying the IADR have huge sizes and it is not possible to construct the models within the time limit (7200 seconds). So, for this class of instances, it is not possible to use IADR, and we only illustrate the worst-case cost obtained by the deterministic problem in Figure 7. Similar to the observations for the small-sized instances, one can see that the worst-case cost is increasing in $\hat{d}$ as well as α.

To have a clear understanding of the worst-case solutions of the instances in this class, we illustrate the solutions for two values of α with $\hat{d}$ being 2 in Figure 8. As one can see, in none of the solutions, there is a warehouse opened in the center of the city. This is because the company

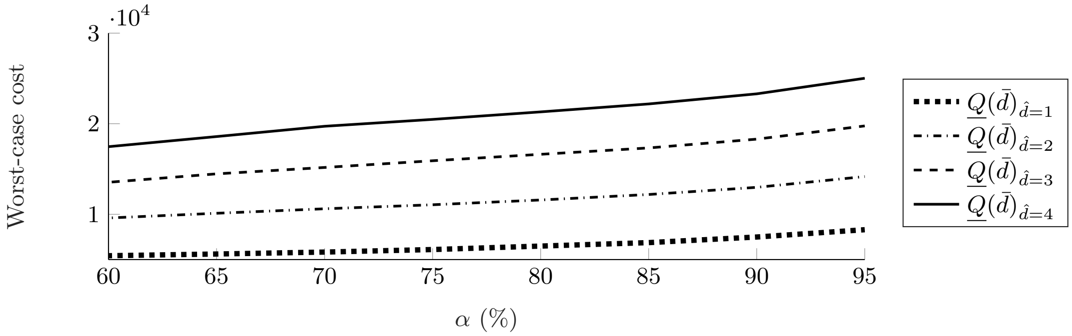
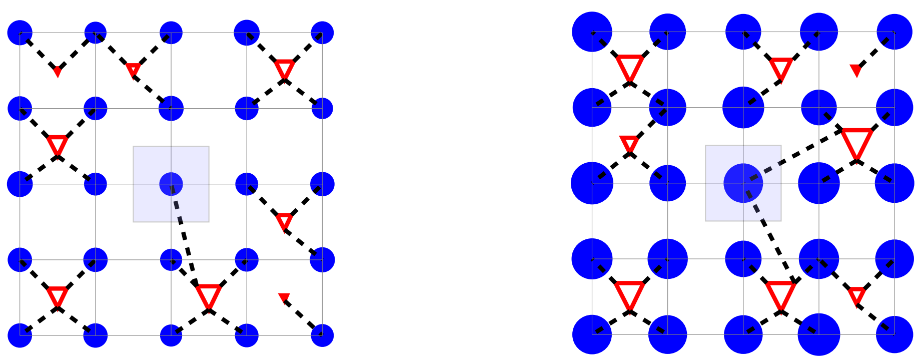


Figure 7 **The optimal worst-case cost obtained for the medium-sized instances with hyper-box uncertainty sets, where $\underline{Q}\left(\bar{d}\right)$ refers to the deterministic LT problem.**



(a) $\alpha = 0.6$

(b) $\alpha = 0.9$

Figure 8 **Illustration of the worst-case solution of the medium-sized instances with $\hat{d} = 2$ and the hyper-box uncertainty set with $\alpha = 0.6$ and $\alpha = 0.9$.**

can service the customers inside and outside of the center by opening warehouses located outside of the center. It is also worth mentioning that for these instances, the warehouses are located such that it is possible to serve all customers in the worst-case scenario within the promised delivery time of 1 hour and hence no tardiness penalty paid to the customers.

An important observation is that there is no linear dependency between α and the basestock policy of each warehouse. In other words, increasing α does not imply increasing the basestock policies of opened warehouses. More precisely, increasing α from 0.6 to 0.9 results in rearranging the resources between different opened warehouses. This is mainly due to the changes in the optimal allocation of resources to customers.

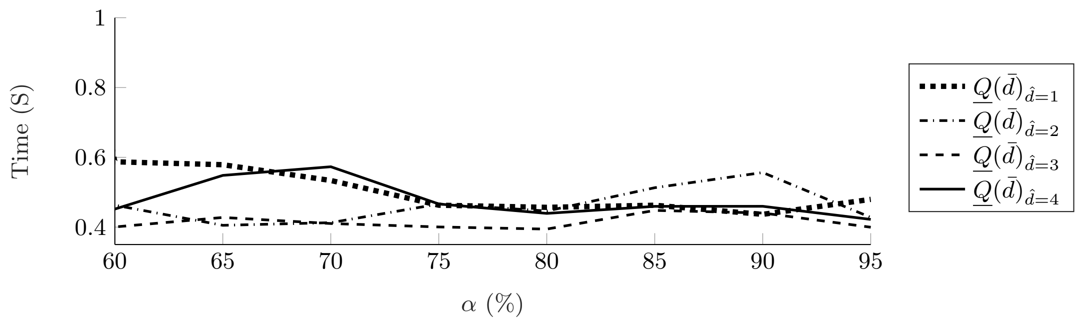


Figure 9 **The time needed to solve the medium-sized instances with hyper-box uncertainty sets, where $\underline{Q}(\bar{d})$ refers to the deterministic LT problem.**

Even though for this class of instances, applying IADR is not practical, the worst-case solutions are obtained relatively fast using deterministic LT problem $\underline{Q}(\bar{d})$. Figure 9 provides the time taken to construct the model and solve $\underline{Q}(\bar{d})$ for different values of $\hat{d}$ and α.

6.3. LT instances with budget uncertainty set

Similar to the construction of the hyper-box uncertainty set, we use empirical distributions to construct budget uncertainty sets. More specifically, we use the set

$$conv(\mathcal{D}_t) := \left\{ d_{kt} \in \mathbb{R}^J \;\middle|\; \begin{array}{ll} 0 \le d_{jkt} \le \bar{d}^{\alpha}_{jkt}, & \forall j \in \mathcal{J}, k \in \mathcal{K} \\ \sum_{\substack{j\in\mathcal{J}\\ k\in\mathcal{K}}} d_{jkt} \le \bar{d}^{\alpha}_t, & \end{array} \right\}, \tag{10}$$

as the convex hull of the budget uncertainty set, where $\bar{d}^{\alpha}_{jkt}, \bar{d}^{\alpha}_t \in \mathbb{N}$ are the upper bounds of the $(1-\alpha)$ confidence intervals derived based on the empirical distribution of d_{jkt} and $\sum_{\substack{j\in\mathcal{J}\\ k\in\mathcal{K}}} d_{jkt}$, respectively, for $j \in \mathcal{J}$, $k \in \mathcal{K}$, $t \in \mathcal{T}$.

As stated in Corollary 2, one can solve the LT instances by enumerating the vertices of the uncertainty set. Table 3 shows the number of vertices of the set constructed in (10) for small-sized instances.

$\hat{d}$	1		2		3		4	
α	0.6	0.9	0.6	0.9	0.6	0.9	0.6	0.9
Number of vertices	10^9	2×10^{14}	7×10^9	2×10^{14}	9×10^{10}	3×10^{14}	4×10^9	10^{14}

Table 3 **Number of vertices of $\prod_{t\in\mathcal{T}} conv(\mathcal{D}_t)$ for small-sized instances with different values of α and $\hat{d}$**

As one can see in Table 3, the enumeration of the vertices of the budget uncertainty set is not computationally efficient to get the exact optimal solution. Therefore, we use the upper bounds generated using the integer-valued affine decision rule (IADR), the lower bounds generated by the Simplex-type method, defined in Section 5, and the lower bounds generated using the adversarial approach proposed by Hadjiyiannis et al. (2011). Under the adversarial approach, the lower bound is generated by limiting the uncertainty set to a finite set of scenarios. The set of scenarios are generated by fixing the solution obtained by IADR and finding the worst demand for each constraint. In what follows, we denote the upper bounds by $IADR$, the lower bounds constructed using Simplex-type method by SM, and the lower bounds constructed using the adversarial approach by AA. In the implementation of the Simplex-type method, to increase the chance of obtaining the global optimum, we run the method for 3 randomly selected initial vertices.

Moreover, as Corollary 2 asserted, the uncertainty set can be reduced to the scenarios in the cut (7). In the numerical experiments, we investigate on whether the reduced uncertainty set would result in improvements in the quality of the obtained solutions. We use the index $_{\text{cut}}$ to refer to the methods applied to the problem with the cut uncertainty set.

6.3.1. Results for small-sized instances To compare the methods, we illustrate the gap between the upper bound obtained by applying IADR and the lower bounds obtained by the Simplex-type method and adversarial approach. Figure 10 shows the gap for different values of $\hat{d}$ and α. We emphasize that applying the IADR to the instances with the budget uncertainty set and the ones with the cut result in the same objective value.

Even though in most cases the Simplex-type method outperforms the adversarial approach with respect to the quality of the obtained solutions, the performance of the Simplex-type method seems to be dependent on the value of $\hat{d}$. For $\hat{d} \in \{1,2\}$, the obtained solutions have a gap of $2-4\%$ while for $\hat{d} \in \{3,4\}$ the gap is reduced to $0.05-0.2\%$. The main reason is that an increase in $\hat{d}$ results in a decrease in the number of vertices of $\mathcal{D}_4$. Therefore, in the Simplex-type method the worst-case scenario(s) can be obtained when $\hat{d}$ increases.

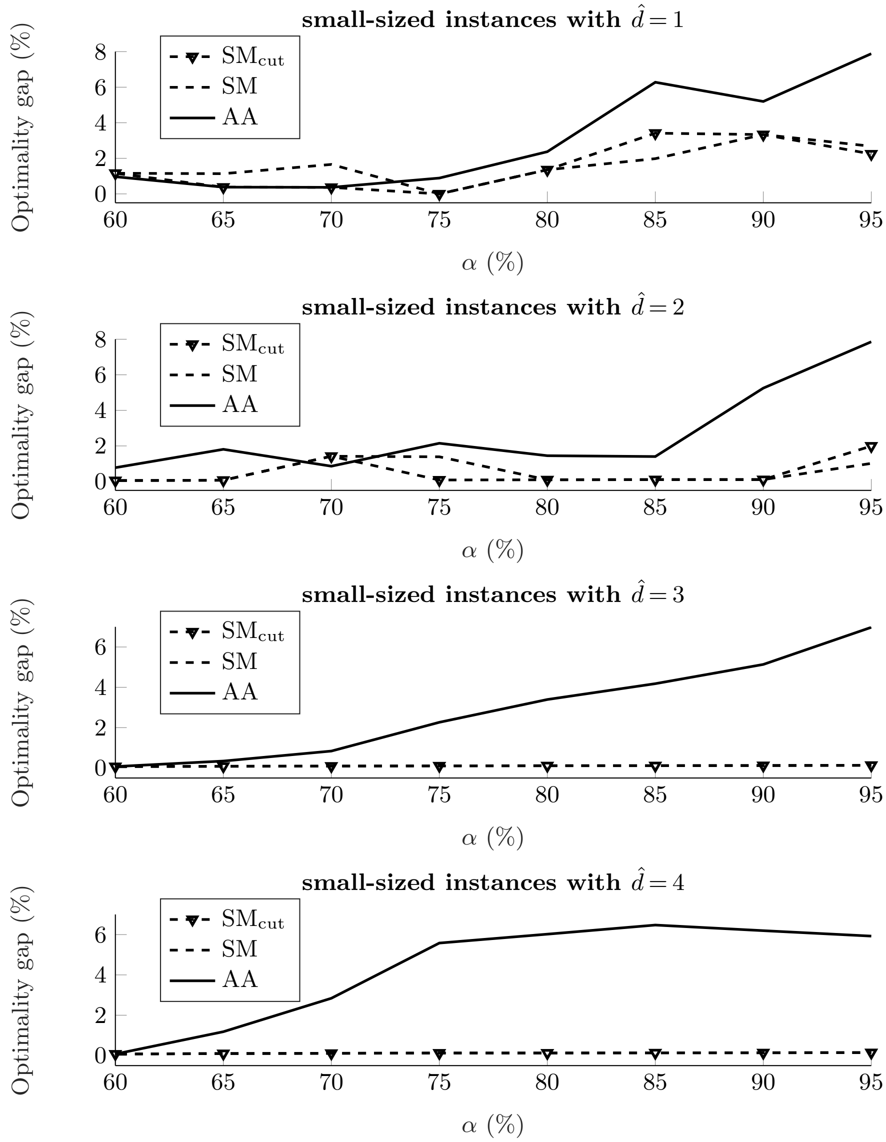


Figure 10 **Comparison of the gap between upper bounds obtained by IADR and the lower bounds obtained by SM and AA for small-sized instances with budget uncertainty set.**

Another important observation is that the cut (7) does not necessarily provide an improvement in the quality of the obtained solutions. Despite the quality of the solutions, restricting the uncertainty set to the cut has the advantage of finding a local solution faster, but still much slower than the time needed to obtain a solution using IADR and AA. Figure 11 shows the time needed for each method to get terminated. As one can see, the time needed for AA to be terminated is much less than the time needed for Simplex-type method. This is due to the fact that the Simplex-type

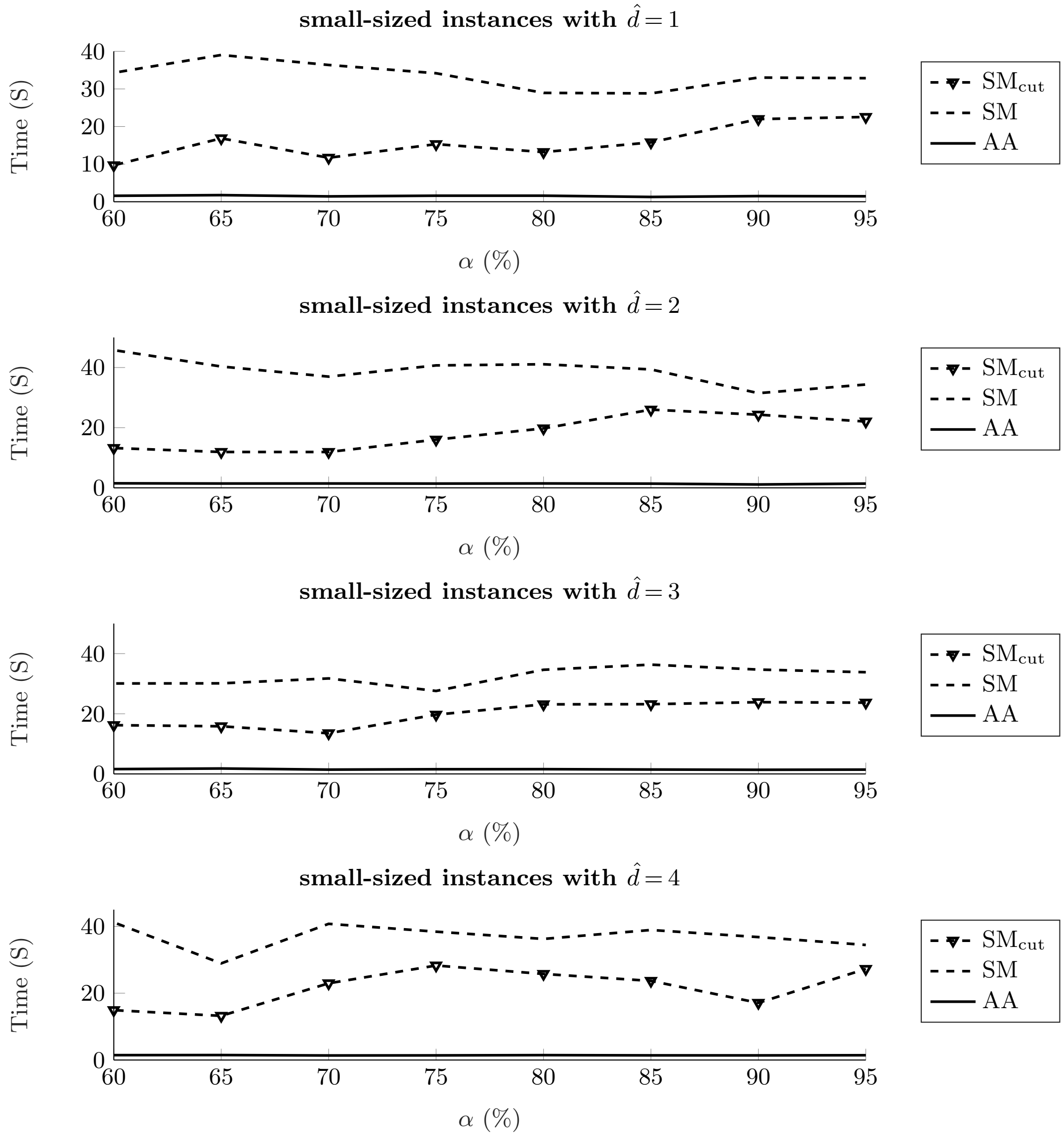


Figure 11 **Time comparisons between different methods for small-sized instances with budget uncertainty set.** AA **and** SM **refer to the adversarial approach and the Simplex-type method, respectively.**

method solves so many deterministic LT problems but IADR solves a medium-sized MILP and many small-sized MILP.

6.3.2. Results for medium-sized instances For this class of instances, IADR is not applicable as it results in an "out-of-memory" error. Therefore, the Simplex-type method is the only approach that can be used to find a solution to the LT instances. For the sake of space, we only

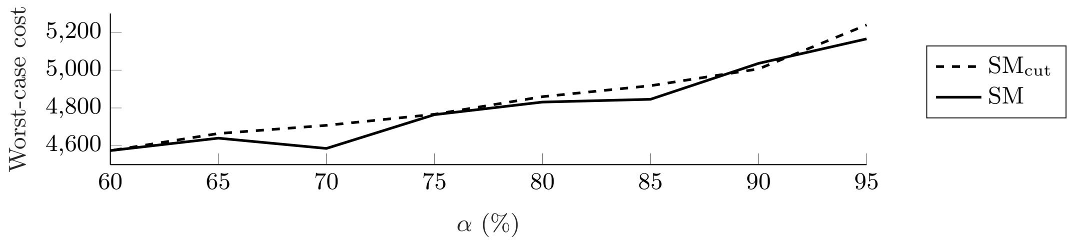


Figure 12 **Comparison of the worst-case cost obtained by Simplex-type method applied to the instances with $\hat{d} = 1$ and the budget uncertainty set.**

present the results of applying the Simplex-type method to instances with $\hat{d} = 1$. For the other values of $\hat{d}$, similar results are obtained.

Figure 12 illustrates the quality of the solutions obtained by the Simplex-type method applied to the instances with the budget uncertainty set and the ones with the cut (7). As one can see, the quality of the solutions are close to each others but the ones obtained using the cut uncertainty set (7) has slightly better quality. This is because when the Simplex-type method starts from a random vertex point of the original set (10) it takes some iterations to reach the cut (7), which is known to contain the worst-case scenario. As the size of the MILP is large, the time limitation may terminate the method before it reaches the cut and hence the quality of the obtained solution is worse than the one obtained using the cut.

Figure 13 illustrates the time taken by the Simplex-method to reach a solution. Even though in most of the instances, the solution is obtained much faster when using the cut uncertainty set compared to using the original set, in two of the cases the method is terminated as it reaches the time limitation, which is due to having many degenerate basis points. As moving from such vertices to others is known to be challenging in the Simplex method, it is also a challenge for our method.

7. Conclusion

In this paper, we have analyzed Location-Transportation (LT) problems with integer demands, consisting of three layers of decisions on locations of facilities or warehouses, distribution of resources among the locations, and allocations of resources to customers. We have dealt with integer demand uncertainty in the LT problems using adjustable robust optimization.

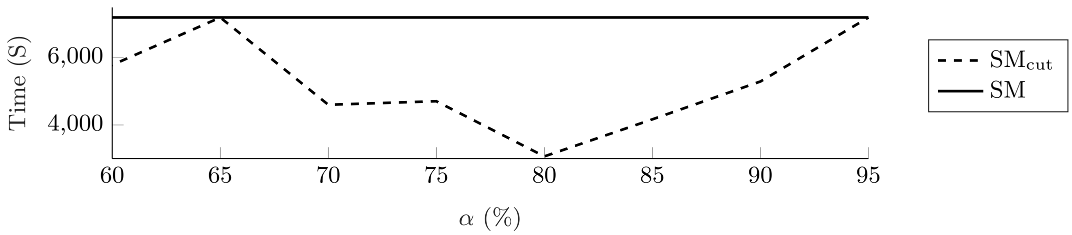


Figure 13 Time comparison between Simplex-type method applied to the instaces with original budget uncertainty set and the ones with cut uncertainty set.

We have shown that the optimal shipment cost from locations of facilities to customers, $Q(y,S,d)$, is a convex function in demand d. Using this characteristic, we have proved that the LT problem can be relaxed to a problem where the "wait-and-see" variables are continuous and the uncertainty set is a polyhedron. Then, we have shown that under some conditions, $Q(y,S,d)$ is a nondecreasing function in demand d, which leads to tractability of a class of LT problems with box and budget uncertainty sets. For a general LT problem, we have designed a new method, called Simplex-type method, to find a locally robust solution, which is motivated by the Simplex method.

We have executed numerical experiments to compare the solution obtained by our method with the ones obtained by the existing method in the literature. The experiments have been set up to design distribution networks of a hypothetical e-commerce company in two different cities. The experiments have shown that our results provide a new state-of-the-art method to solve such problems with a hyper-box uncertainty set. For a budget uncertainty set, the Simplex-type method has provided solutions of a better quality but slower than the existing method for small-size instances. For larger instances, the Simplex-type method has been the only method that could provide a solution to the LT instances.

References

Ardestani-Jaafari A, Delage E (2016) Robust optimization of sums of piecewise linear functions with application to inventory problems. *Operations Research* 64(2):474–494.

Ardestani-Jaafari A, Delage E (2017) The value of flexibility in robust location–transportation problems. *Transportation Science* 52(1):189–209.

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

Bardossy MG, Raghavan S (2013) Robust optimization for the connected facility location problem. *Electronic Notes in Discrete Mathematics* 44:149 – 154.

Baron O, Milner J, Naseraldin H (2011) Facility location: A robust optimization approach. *Production and Operations Management* 20(5):772–785.

Bazaraa MS, Jarvis JJ, Sherali HD (2011) *Linear programming and network flows* (John Wiley & Sons).

Ben-Tal A, Chung BD, Mandala SR, Yao T (2011) Robust optimization for emergency logistics planning: Risk mitigation in humanitarian relief supply chains. *Transportation Research Part B: Methodological* 45(8):1177 – 1189.

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

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

Bertsimas D, Bidkhori H (2015) On the performance of affine policies for two-stage adaptive optimization: a geometric perspective. *Mathematical Programming* 153(2):577–594.

Bertsimas D, Caramanis C (2007) Adaptability via sampling. *46th IEEE Conference on Decision and Control*, 4717–4722.

Bertsimas D, Dunning I (2016) Multistage robust mixed-integer optimization with adaptive partitions. *Operations Research* 64(4):980–998.

Bertsimas D, Goyal V (2012) On the power and limitations of affine policies in two-stage adaptive optimization. *Mathematical Programming* 134(2):491–531.

Bertsimas D, Iancu DA, Parrilo PA (2010) Optimality of affine policies in multistage robust optimization. *Mathematics of Operations Research* 35(2):363–394.

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

Bezanson J, Edelman A, Karpinski S, Shah V (2017) Julia: A fresh approach to numerical computing. *SIAM Review* 59(1):65–98.

Buhayenko V, den Hertog D (2017) Adjustable robust optimisation approach to optimise discounts for multi-period supply chain coordination under demand uncertainty. *International Journal of Production Research* 55(22):6801–6823.

Campbell T, How JP (2015) Bayesian nonparametric set construction for robust optimization. *2015 American Control Conference (ACC)*, 4216–4221 (IEEE).

Chen X, Zhang Y (2009) Uncertain linear programs: Extended affinely adjustable robust counterparts. *Operations Research* 57(6):1469–1482.

Cooper L (1972) The transportation-location problem. *Operations Research* 20(1):94–108.

Crainic TG, Laporte G (1997) Planning models for freight transportation. *European Journal of Operational Research* 97(3):409 – 438.

Dey SS, Morán DA (2013) Some properties of convex hulls of integer points contained in general convex sets. *Mathematical Programming* 141(1):507–526.

Dunning I, Huchette J, Lubin M (2017) JuMP: A modeling language for mathematical optimization. *SIAM Review* 59(2):295–320.

Dunning IR (2016) *Advances in robust and adaptive optimization: algorithms, software, and insights*. Ph.D. thesis, Massachusetts Institute of Technology.

El-Amine H, Bish EK, Bish DR (2017) Robust postdonation blood screening under prevalence rate uncertainty. *Operations Research* 66(1):1–17.

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

Gabrel V, Murat C, Thiele A (2014b) Recent advances in robust optimization: An overview. *European Journal of Operational Research* 235(3):471–483.

Gorissen BL, hsan Yankolu, den Hertog D (2015) A practical guide to robust optimization. *Omega* 53:124 – 137.

Hadjiyiannis MJ, Goulart PJ, Kuhn D (2011) A scenario approach for estimating the suboptimality of linear decision rules in two-stage robust optimization. *50th IEEE Conference on Decision and Control and European Control Conference*, 7386–7391.

Holsenbeck KF (2018) Everything you need to know about prime now. `https://www.amazon.com/primeinsider/tips/prime-now-qa.html`.

Housni OE, Goyal V (2018) On the optimality of affine policies for budgeted uncertainty sets. *arXiv:1807.00163*.

Jiang H (2012) *Central warehouse planning within the service parts network of ASML*. Master's thesis, Eindhoven University of Technology, the Netherlands.

Klibi W, Lasalle F, Martel A, Ichoua S (2010) The stochastic multiperiod location transportation problem. *Transportation Science* 44(2):221–237.

Klincewicz JG (1990) Solving a freight transport problem using facility location techniques. *Operations Research* 38(1):99–109.

Koninklijke Philips NV (2019) Philips comprehensive on-site service agreement.

Lappas NH, Gounaris CE (2016) Multi-stage adjustable robust optimization for process scheduling under uncertainty. *AIChE Journal* 62(5):1646–1667.

Lara C, Flores JJ, Calderon F (2009) On the hyperbox–hyperplane intersection problem. *INFOCOMP* 8(4):21–27.

Limbourg S, Jourquin B (2009) Optimal rail-road container terminal locations on the European network. *Transportation Research Part E: Logistics and Transportation Review* 45(4):551–563.

Marandi A, den Hertog D (2018) When are static and adjustable robust optimization problems with constraint-wise uncertainty equivalent? *Mathematical Programming* 170(2):555–568.

Nagy G, Salhi S (2007) Location-routing: Issues, models and methods. *European Journal of Operational Research* 177(2):649 – 672.

Ning C, You F (2018) Data-driven decision making under uncertainty integrating robust optimization with principal component analysis and kernel smoothing methods. *Computers & Chemical Engineering* 112:190 – 210.

O'Neil PE (1971) Hyperplane cuts of an n-cube. *Discrete Mathematics* 1(2):193 – 195.

Postek K, den Hertog D (2016) Multistage adjustable robust mixed-integer optimization via iterative splitting of the uncertainty set. *INFORMS Journal on Computing* 28(3):553–574.

Prodhon C, Prins C (2014) A survey of recent research on location-routing problems. *European Journal of Operational Research* 238(1):1–17.

Romeijnders W, Postek K (2018) Piecewise constant decision rules via branch-and-bound based scenario detection for integer adjustable robust optimization. *arXiv:1805.10079*.

Simchi-Levi D, Wang H, Wei Y (2018) Increasing supply chain robustness through process flexibility and inventory. *Production and Operations Management* 27(8):1476–1491.

Solyalı O, Cordeau JF, Laporte G (2015) The impact of modeling on robust inventory management under demand uncertainty. *Management Science* 62(4):1188–1201.

Subramanyam A, Gounaris CE, Wiesemann W (2019) K-adaptability in two-stage mixed-integer robust optimization. *Mathematical Programming Computation* 1–32.

Yang Y, Zhou R, Ran X (2012) Robust optimization with box set for reactive power optimization in wind power integrated system. *2012 IEEE Power and Energy Society General Meeting, San Diego, CA*, 1–6.

Yanıkoğlu İ, Gorissen BL, den Hertog D (2019) A survey of adjustable robust optimization. *European Journal of Operational Research* 277(3):799–813.

Zhao L, Zeng B (2012) An exact algorithm for two-stage robust optimization with mixed integer recourse problems. *Optimization Online*.

Zhen J, den Hertog D, Sim M (2018) Adjustable robust optimization via fourier–motzkin elimination. *Operations Research* 66(4):1086–1100.

Appendices

This paper contains three appendices. In Appendix A, we provide the formulation of a deterministic multi-stage LT problem. Appendix B contains two ways of proving one of the results of this paper. Lastly, Appendix C provides the reformulation of the multi-stage robust LT problem using the ARO method after the elimination of the equality restrictions.

A. Deterministic problem

In this section, we provide the mathematical formulation of the deterministic LT problem, where the demand vector $d \in \mathbb{N}_0^{J\times K\times T}$ is known exactly:

$$\begin{aligned}
\underline{Q}(d) = \sum_{\substack{j\in\mathcal{J}\\k\in\mathcal{K}\\t\in\mathcal{T}}} c_{0jk}d_{jkt} + \min_{y,S,z}\ & \sum_{i\in\mathcal{I}} f_i y_i + T\sum_{\substack{i\in\mathcal{I}\\k\in\mathcal{K}}} h_{ik}S_{ik} + \sum_{\substack{i\in\mathcal{I}\\j\in\mathcal{J}\\k\in\mathcal{K}\\t\in\mathcal{T}}} \left(c_{ijk} - (T-t)\,h_{ik} - c_{0jk}\right) z_{ijkt} \\
\text{s.t.}\ \ & S_{ik} \le M y_i, \qquad \forall i\in\mathcal{I}, k\in\mathcal{K}, \\
& \sum_{\substack{j\in\mathcal{J}\\t\in\mathcal{T}}} z_{ijkt} \le S_{ik}, \qquad \forall i\in\mathcal{I}, k\in\mathcal{K}, \\
& \sum_{i\in\mathcal{I}} z_{ijkt} \le d_{jkt}, \qquad \forall j\in\mathcal{J}, k\in\mathcal{K}, t\in\mathcal{T} \\
& S_{ik}, z_{ijkt} \in \mathbb{N}_0, \qquad \forall i\in\mathcal{I}, j\in\mathcal{J}, k\in\mathcal{K}, t\in\mathcal{T}, \\
& y_i\in\{0,1\}, \qquad \forall i\in\mathcal{I}.
\end{aligned} \tag{11}$$

THEOREM 3. *Consider the deterministic problem* (11)*. If for* $\bar{j}\in\mathcal{J}$ *and* $i_1\in\mathcal{I}$ *there exists* $i_2\in\mathcal{I}$ *such that*

$$\max\left\{h_{i_2\bar{k}} - h_{i_1\bar{k}}, T\left(h_{i_2\bar{k}} - h_{i_1\bar{k}}\right)\right\} < c_{i_1\bar{j}\bar{k}} - c_{i_2\bar{j}\bar{k}},$$

and in the optimal solution $y_{i_2}=1$ *then* $z_{\bar{j}i_1\bar{k}t}=0$ *for any* $t\in\mathcal{T}$*.*

Proof. Let us denote an optimal solution of (11) by (y^*, S^*, z^*). By contradiction, let us assume there exists a time-slot $\bar{t}\in\mathcal{T}$ such that $z^*_{\bar{j}i_1\bar{k}\bar{t}} > 0$. We will construct a new solution that has a better objective value. To this end, we set

$$S_{ik}^{new} := \begin{cases} S^*_{i_1\bar{k}} - z^*_{\bar{j}i_1\bar{k}\bar{t}} & \text{if } i=i_1, k=\bar{k}, \\ S^*_{i_2\bar{k}} + z^*_{\bar{j}i_1\bar{k}\bar{t}} & \text{if } i=i_2, k=\bar{k}, \\ S^*_{ik} & \text{otherwise,} \end{cases} \qquad z_{ijkt}^{new} = \begin{cases} 0 & \text{if } i=i_1, k=\bar{k}, t=\bar{t}, j=\bar{j}, \\ z^*_{\bar{j}i_2\bar{k}\bar{t}} + z^*_{\bar{j}i_1\bar{k}\bar{t}} & \text{if } i=i_2, k=\bar{k}, t=\bar{t}, j=\bar{j}, \\ z^*_{ijkt} & \text{otherwise,} \end{cases}$$

and $y^{new} = y^*$. It is easy to check that $(y^{new}, S^{new}, z^{new})$ is a feasible solution for (11). Let us denote by Obj^* and Obj^{new} the optimal value and the objective value of the constructed solution, respectively. We have

$$Obj^* - Obj^{new} = z^*_{\bar{j}i_1\bar{k}\bar{t}} \left(\bar{t}\left(h_{i_1\bar{k}} - h_{i_2\bar{k}}\right) + c_{i_1\bar{j}\bar{k}} - c_{i_2\bar{j}\bar{k}}\right). \tag{12}$$

Also, we have

$$\bar{t}\left(h_{i_2\bar{k}} - h_{i_1\bar{k}}\right) \leq \max\left\{h_{i_2\bar{k}} - h_{i_1\bar{k}}, T\left(h_{i_2\bar{k}} - h_{i_1\bar{k}}\right)\right\} < c_{i_1\bar{j}\bar{k}} - c_{i_2\bar{j}\bar{k}}, \tag{13}$$

where the right inequality is due to the assumption. Therefore, (12) and (13) imply that $Obj^* > Obj^{new}$, which contradicts optimality of (y^*, S^*, z^*). □

COROLLARY 3. *Let us fix $i_1, i_2 \in \mathcal{I}$. For the deterministic problem (11), let for any $j \in \mathcal{J}$, $k \in \mathcal{K}$,*

$$\max\left\{h_{i_2k} - h_{i_1k}, T\left(h_{i_2k} - h_{i_1k}\right)\right\} < c_{i_1jk} - c_{i_2jk},$$

and $f_{i_2} < f_{i_1}$. Then, in the optimal solution $y_{i_1} = 0$.

Proof. Let us assume by contradiction that in the optimal solution $y_{i_1} = 1$. If $y_{i_2} = 1$ then Theorem 3 implies that there exists a solution where $y_{i_1} = 0$ with a better objective function.

Now, let us assume that $y_{i_2} = 0$ in the optimal solution. Let us denote the optimal solution by (y^*, S^*, z^*). Then, consider a new solution (y^1, S^1, z^1), where $S^1 = S^*$, $z^1 = z^*$, and $y_i^1 = y_i^*$ for $i \in \mathcal{I} \setminus \{i_1, i_2\}$, $y_{i_1}^1 = 1$, and $y_{i_2}^1 = 1$. Them, clearly the objective value of the constructed solution, denoted by Obj^1, equals to $Obj^* + f_{i_2}$, where Obj^* denotes the optimal value. Now, Theorem 3 implies that there exists a solution (y^2, S^2, z^2) that has a better objective value Obj^2 than Obj^1 and where $z^2_{ji_1kt} = 0$ for any $j \in \mathcal{J}$, $k \in \mathcal{K}$, and $t \in \mathcal{K}$. So, we can easily construct a solution (y^3, S^3, z^3) by setting $z^3 = z^2$, $S^3 = S^2$, $y_i^3 = y_i^2$ for $i \in \mathcal{I} \setminus \{i_1\}$, and $y_{i_1}^3 = 0$, with the objective value of $Obj^2 - f_{i_1}$. Therefore, we have

$$Obj^3 := Obj^2 - f_{i_1} < Obj^1 - f_{i_1} = Obj^* + f_{i_2} - f_{i_1} < Obj^*,$$

where the right inequality is due to the assumption $f_{i_2} < f_{i_1}$. This contradicts the optimality of (y^*, S^*, z^*). □

B. Proof of Corollary 1

Corollary 1 can be proved in two different ways using different properties of the problem.

Proof 1. Let us first assume that $T = 1$. Using Theorem 2, we have $\overline{d} \in \operatorname{argmax}_{d \in \mathcal{D}} Q(y, S, d)$. Therefore, (3) is equivalent to $\underline{Q}(\overline{d})$.

For a general $T \geq 1$, one can easily use the same argument as for $T = 1$, starting from $t = T$, and find that

$$[\overline{d}_{jkt}]_{\substack{j \in \mathcal{J} \\ k \in \mathcal{K}}} \in \operatorname{argmax}_{d_t \in \mathcal{D}_t} Q_t\left(y, S, d_{[t]}, z_{t-1}(d_{[t-1]})\right),$$

since $c_{ijk} \geq h_{ik}$, for any $i \in \mathcal{I}$, $j \in \mathcal{J}$, and $k \in \mathcal{K}$. □

The first proof uses the nondecreasingness property of $Q(y, S, d)$. In the next proof, we show how to use the convexity of $Q(y, S, d)$ to prove Corollary (3).

Proof 2. We prove the corollary for $T = 1$, and the extension can be done analogous to *Proof 1*. Let $T = 1$. By Theorem 1, and since $c_{ijk} \geq 0$, $i \in \mathcal{I}$, $j \in \mathcal{J}$, $k \in \mathcal{K}$, we have

$$\begin{aligned} Q(y, S, d) = \min_{z(d)} \quad & \sum_{\substack{i \in \mathcal{I} \cup \{0\}, \\ j \in \mathcal{J}, \\ k \in \mathcal{K}}} c_{ijk} z_{ijk1}(d) \\ \text{s.t.} \quad & \sum_{j \in \mathcal{J}} z_{ijk1}(d) \leq S_{ik}, && \forall i \in \mathcal{I}, k \in \mathcal{K}, \\ & \sum_{i \in \mathcal{I} \cup \{0\}} z_{ijk1}(d) \geq d_{jk1}, && \forall j \in \mathcal{J}, k \in \mathcal{K}, \\ & z_{ijk1}(d), z_{j0k1}(d) \geq 0, && \forall i \in \mathcal{I}, j \in \mathcal{J}, k \in \mathcal{K}, \end{aligned} \tag{14}$$

where the uncertainty set is $conv(\mathcal{D})$. As the uncertainty set is a hyper-box and for any $j \in \mathcal{J}$, $k \in \mathcal{K}$, d_{jk1} appears only in one constraint, i.e. (14) has a constraint-wise uncertainty, results by

Marandi and den Hertog (2018) imply that (2) is equivalent to its static Robust Optimization formulation, i.e.,

$$
\begin{aligned}
\min_{y,\ S,\ z} \quad & \sum_{i\in\mathcal{I}} f_i y_i + \sum_{\substack{i\in\mathcal{I}\\ k\in\mathcal{K}}} h_{ik}S_{ik} + \sum_{\substack{i\in\mathcal{I}\cup\{0\},\\ j\in\mathcal{J},\\ k\in\mathcal{K}}} c_{ijk}z_{ijk1} \\
\text{s.t.} \quad & \sum_{j\in\mathcal{J}} z_{ijk1} \le S_{ik}, && \forall i\in\mathcal{I}, k\in\mathcal{K},\\
& \sum_{i\in\mathcal{I}\cup\{0\}} z_{ijk1} \ge d_{jk1}, && \forall d\in\mathcal{D},\ \forall j\in\mathcal{J}, k\in\mathcal{K},\\
& S_{ik} \le My_i, && \forall i\in\mathcal{I}, k\in\mathcal{K},\\
& y_i\in\{0,1\}, && \forall i\in\mathcal{I},\\
& z_{ijk1}, z_{j0k1} \ge 0, && \forall i\in\mathcal{I}, j\in\mathcal{J}, k\in\mathcal{K},\\
& S_{ik}\in\mathbb{N}_0, && \forall i\in\mathcal{I}, k\in\mathcal{K},
\end{aligned}
$$

which is equivalent to $\underline{Q}(\bar{d})$. $\square$

C. Elimination of the equality constraints

In this section, we provide the explicit formulation of the LT problem where the equality constraints as well as the variables corresponding to the shipments from the central warehouse to customers are eliminated.

$$
\begin{aligned}
LT(\mathcal{D}) := \min_{y,S} \quad & \sum_{i\in\mathcal{I}} f_i y_i + \sum_{\substack{i\in\mathcal{I}\\ k\in\mathcal{K}}} h_{ik}S_{ik} + \max_{d_{[1]}\in\mathcal{D}_1} Q_1(y,S,d_{[1]},0_{JK}) \\
\text{s.t.} \quad & S_{ik} \le My_i, && \forall i\in\mathcal{I}, k\in\mathcal{K},\\
& S_{ik}\in\mathbb{N}_0, && \forall i\in\mathcal{I}, k\in\mathcal{K},\\
& y_i\in\{0,1\}, && \forall i\in\mathcal{I},
\end{aligned}
\tag{3}
$$

where $\mathcal{D} = \prod_{t\in\mathcal{T}} \mathcal{D}_t$, and for $t = 1,...,T-1$,

$$Q_t\left(y, S_0, d_{[t]}, z_{[t-1]}(d_{[t-1]})\right) =$$

$$\begin{aligned}
\min_{z_t(d_{[t]})} \quad & \sum_{\substack{i \in \mathcal{I} \\ j \in \mathcal{J} \\ k \in \mathcal{K}}} c_{ijk} z_{ijkt}(d_{[t]}) + \sum_{\substack{i \in \mathcal{I} \\ k \in \mathcal{K}}} h_{ik} S_{ik} - \sum_{\substack{i \in \mathcal{I} \\ k \in \mathcal{K} \\ j \in \mathcal{J}}} h_{ik} \left(\sum_{\tilde{t}=1}^{t-1} z_{jik\tilde{t}}(d_{[\tilde{t}]}) \right), \\
& + \sum_{\substack{j \in \mathcal{J}, \\ k \in \mathcal{K}}} c_{0jk} \left(d_{jkt} - \sum_{i \in \mathcal{I}} z_{ijkt}(d_{[t]}) \right) + \max_{d_{t+1} \in \mathcal{D}_{t+1}} Q_{t+1}\left(y, S, d_{[t+1]}, z_{[t]}(d_{[t]})\right) \qquad (15) \\
\text{s.t.} \quad & \sum_{i \in \mathcal{I}} z_{ijkt}(d_{[t]}) \le d_{jkt}, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}, \\
& z_{ijkt}(d_{[t]}) \in \mathbb{N}_0, \qquad \forall i \in \mathcal{I}, j \in \mathcal{J}, k \in \mathcal{K},
\end{aligned}$$

and

$$\begin{aligned}
& Q_T(y, S, d_{[T]}, z_{[T-1]}(d_{[T-1]})) = \\
& \min_{z_T(d_{[T]})} \quad \sum_{\substack{i \in \mathcal{I}, \\ j \in \mathcal{J}, \\ k \in \mathcal{K}}} c_{ijk} z_{ijkt}(d_{[T]}) + \sum_{\substack{j \in \mathcal{J}, \\ k \in \mathcal{K}}} c_{0jk} \left(d_{jkT} - \sum_{i \in \mathcal{I}} z_{ijkt}(d_{[T]}) \right) \\
& \text{s.t.} \quad \sum_{j \in \mathcal{J}} z_{ijkt}(d_{[T]}) \le S_{ik} - \sum_{j \in \mathcal{J}} \sum_{t=1}^{T-1} z_{jikt}(d_{[t]}), \qquad \forall i \in \mathcal{I}, k \in \mathcal{K}, \qquad (16) \\
& \qquad \sum_{i \in \mathcal{I}} z_{ijkt}(d_{[T]}) \le d_{jkT}, \qquad \forall j \in \mathcal{J}, k \in \mathcal{K}, \\
& \qquad z_{ijkT}(d_{[T]}) \in \mathbb{N}_0, \qquad \forall i \in \mathcal{I}, j \in \mathcal{J}, k \in \mathcal{K}.
\end{aligned}$$