Multi-Stage Adjustable Robust Location-Transportation Problems with Integer-Valued Demand

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

Geert-Jan van Houtum
Department of Industrial Engineering and Innovation Sciences, Eindhoven University of Technology, Eindhoven, The Netherlands
g.j.v.houtum@tue.nl

Abbas Khademi
School of Mathematics, Institute for Research in Fundamental Sciences (IPM), Tehran, Iran
abbaskhademi92@gmail.com

Abstract

A location-transportation 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 products within given delivery time targets. The company needs to first decide on the locations of local warehouses before the start of the time horizon. During the time horizon, the stocks at the local warehouses are repeatedly replenished, and the company has to decide how far the inventory levels are increased at those moments. Our problem is such that we can use time-independent base stock levels at all warehouses at those moments. Between any two replenishments, integer-valued demands are realized multiple times, and the company needs to satisfy them by shipments from the central and local warehouses to the customers.

In this paper, we follow an adjustable robust optimization approach for the design of the distribution network. We prove two main characteristics of our location-transportation problems, namely convexity and non-decreasingness of the optimal shipment cost function. Using these characteristics, we show for two commonly used uncertainty sets (hyper-box and budget uncertainty sets) that the optimal decisions on the location and the base stock levels of local warehouses can be made efficiently. For a general bounded uncertainty set, we propose a new method, called the 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 as well as an exact solution approach.

Keywords Adjustable robust optimization, Location-transportation problem, Multi-stage problems, Uncertainty modeling, Affine decision rule, Simplex-type method.

1 Introduction

An optimal design of a distribution network is extremely important but, at the same time, complicated. An optimal design 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 network, three layers of decisions are identified: strategic (long-term) planning decisions, which are the ones on the locations of the facilities, warehouses, or hubs; tactical (medium-term) planning decisions, which include the distribution of resources among the locations; and the operational (short-term) planning decisions, which concern how to allocate resources to satisfy the customers' demands [33].

In this paper, we focus on Location-Transportation (LT) problems [32], addressing the optimal decisions on the three layers, where especially in the operational layer, the decision maker takes actions on how to satisfy the integer-valued demand of customers, possibly from multiple locations. Such problems arise naturally in many real-life situations, including e-commerce [2], where a timely delivery service is key, and service logistics [49, 54], where the availability of spare parts plays an important role in the functionality of capital goods.

One of the challenges in making integrated decisions in LT problems is demand uncertainty. All decisions are made to satisfy customers' demands, and the uncertainty in demand increases the complexity of the problem. To deal with uncertainties in practical problems, there is an increase in the use of robust optimization methods, as such methods are independent of distribution functions and, hence, in many applications, computationally

© Ahmadreza Marandi & Geert-Jan van Houtum & Abbas Khademi;

Draft paper as of August 25, 2025

tractable (see, e.g., [6, 8, 60, 62]). In this paper, we contribute to the literature concerning robust optimization methodology to solve real-life LT problems with integer-valued uncertain parameters.

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 most computationally tractable method in robust optimization, formally introduced in [16], 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, using SRO for LT problems implies that not only the decisions on locations of warehouses (strategic) and distribution of resources among the warehouses (tactical) are made *earlier* than the realization of the customers' demand but also the decision on resource allocations to the customers (operational) are made *before* customers' demands are revealed. In [11] the authors 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 using SRO, and (iii) a robust problem with a *hyper-ellipsoid* uncertainty set 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*, in [10] the authors study a similar problem and propose an approximate algorithm to solve the robust problem formulated using the SRO method. Furthermore, in [9] the authors show that considering a larger uncertainty set does not necessarily result in a more robust solution.

The solutions obtained by the 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, in [15] the authors propose a new method, called Adjustable Robust Optimization (ARO), where a part of the decisions is *"here-and-now"*, while the rest of the decisions are made *after* the realization of the uncertain parameter, hence so-called *"wait-and-see"* decisions.

This framework not only captures a more realistic decision-making process but can also be extended to model multi-stage decision-making settings, offering greater flexibility and adaptability in addressing complex, uncertain environments. Although ARO improves solution quality by being less conservative, it is computationally more complex than SRO and is generally NP-hard [19]. Interestingly, under certain problem settings, ARO can be shown to be equivalent to SRO [24, 59]. Extensive studies have been conducted on the solvability of ARO problems, leveraging methods such as copositive programming [73], Fourier-Motzkin elimination [77], lifting uncertainty set [45], dual theory [34, 36, 52], and decision rules [17, 23, 71]. All these approaches are designed for problems with continuous variables as well as convex uncertainty sets.

For problems involving both continuous and discrete wait-and-see decisions, finite adaptability formulations can be applied by partitioning the convex uncertainty set [20, 63, 64]. Another commonly used approach is K-adaptability, where instead of complete flexibility in responding to uncertainty, only a pool of K response strategies is considered [50, 55, 69]. Furthermore, special decision rules have been developed by [21, 22]. For a detailed review of the literature, readers may refer to the survey papers [37, 44, 75] and the references therein, as well as Chapters 6-12 in [19] for a comprehensive overview of recent developments.

As ARO is capable of formulating multi-stage real-life situations, it has attracted significant attention, leading to extensive studies on the practical applicability of robust optimization problems using the ARO method, as evidenced by works such as [1, 7, 28, 35, 39, 48, 66, 78]. These studies highlight the versatility and effectiveness of ARO in addressing complex, uncertain decision-making environments. Given the applicability, there have been efforts in developing exact solution algorithms for such problems, including [63, 64]. However, such approximation and exact approaches suffer from the curse of dimensionality, showing the need for the development of tractable approaches.

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 demands 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 the day to satisfy the two-hour delivery service. Therefore, LT problems are formulated as *multi-stage* ARO problems. In [5] the authors show that even the two-stage LT problems (where the operational planning decisions are taken once) belong to the class of NP-hard problems, and provide a method to approximate them for budget uncertainty sets. Then, in [43] the authors develop an iterative cutting plane algorithm to approximate two-stage LT problems. In [67] the authors consider a similar model with a more complicated uncertainty set and derive the optimal resource allocation policy. In [4] the authors analyze multi-stage adjustable LT problems with real-valued demands, where 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. To approximate such problems, in [72] the authors restrict the "wait-and-see" variables to be affine in demand (instead of being a general function), and develop a Benders decomposition approach to solve the approximated problem.

The common assumption in the literature, which is not always practical for LT problems, is that the demands are non-negative *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, the assumption of having real-valued demands cannot be justified, especially when the demands have low values. On one hand, such ways of reformulating are not applicable to problems with *integer-valued demands*. We show, by means of an illustrative example, that relaxing this integrality restriction can have a significant negative impact and results in solutions with a large optimality gap. On the other hand, incorporating the integer-valued demands directly into the formulation results in a linear integer optimization problem, potentially with an exponential number of constraints and variables. To the best of our knowledge, there is no literature on developing tractable methods for multi-stage adjustable LT problems with integer-valued demands.

In this paper, we focus on such problems with integer-valued demands and close the existing gap in the literature by making a four-fold contribution. First, we show that for a multi-stage adjustable LT problem with integer-valued demands, the minimum cost of resource allocations is a convex function in demand (*convexity characteristic*). Using this characteristic, we show that: (i) using an enumeration of vertices of the uncertainty set, one can solve multi-stage adjustable LT problems to optimality; and (ii) if an explicit description of the convex hull of the uncertainty set is available, then a continuous relaxation of an LT problem provides the worst-case optimal value as well as a robust optimal strategic and tactical decisions. The importance of this contribution is on identifying uncertainty sets, for which an LT problem can be solved exactly.

Second, we show that for a class of multi-stage adjustable LT problems, the minimum cost of resource allocations is non-decreasing in demand. In other words, we mathematically prove that an increase in the demands of different customers for different products results in an equal or higher optimal cost of satisfying the demands (*non-decreasingness characteristic*). Based on this characteristic, we show that under some conditions, a multi-stage adjustable 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 adjustable 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 adjustable LT problem can be obtained by replacing the uncertainty set with its subset, which can lead to an improvement in computational complexity. For instance, using this result, we show that for a class of multi-stage adjustable LT problems with a specific budget uncertainty set, the problem is computationally tractable.

As one notices, the complexity of a multi-stage adjustable LT problem is highly dependent on the considered uncertainty set. As our third contribution in this paper, we design a new method to obtain a locally robust solution for a multi-stage adjustable LT problem with integer-valued demands and a general bounded uncertainty set. Our method uses the convexity characteristic and is based on a similar principle as the Simplex method; 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. We emphasize that the Simplex-type method can also be applied to a general linear ARO problem with right-hand-side uncertainty.

Finally, we conduct an extensive numerical study on a problem setting motivated by a real-life case to compare the performance of different approaches, including our implementation of the Finite Scenario approach, the ones proposed in this paper, and the one proposed in [64]. Our numerical experiments show that the approach proposed in [64] on small-sized instances is unable to find a good quality solution and only generates bounds that are extremely loose, while the Simplex-type method performs well in finding solutions with low optimality gaps. For this class of instances, the Finite Scenario approach generates much faster solutions with larger optimality gaps compared to the Simplex-type method. For medium-sized and large-sized instances, the Finite Scenario approach computes a fast solution with a larger gap, while our Simplex-type method provides a solution in a reasonable computation time with a lower gap. Hence, the computational experiments show that for the large-size problems, one can use our implementation of the Finite Scenario approach to generate a solution fast, or use the Simplex-type method that is more computationally expensive to get a better quality solution.

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. Section 4 demonstrates that relaxing integer demand uncertainty leads to suboptimal decisions with a significant optimality gap, which is most pronounced for low demand quantities with more customers, underscoring the practical relevance of the integrality assumption. In Section 5, we prove the theoretical results of this paper regarding the convexity and non-decreasingness characteristics and show how one can use them to find optimal solutions. In Section 6, we show how to approximate a multi-stage adjustable LT problem using the integer-valued affine decision rules and develop the Simplex-type method. Section 7 contains the numerical experiments of this paper and illustrates the effectiveness of our results in finding a good quality robust solution. We conclude the paper in Section 8.

Notation

In this paper, the sets of real numbers, integers, and positive integers are denoted by $\mathbb{R}$, $\mathbb{Z}$, and $\mathbb{N}$, respectively. We define $\mathbb{N}_0$ as $\mathbb{N} \cup \{0\}$. For any real number $x \in \mathbb{R}$, we define its positive part as $(x)^+ := \max\{x, 0\}$. For vectors $d^1, d^2 \in \mathbb{R}^n$, the notation $d^1 \lneqq d^2$ indicates that d^1 is component-wise less than or equal to d^2, with at least one strict inequality. The inner product of d^1 and d^2 is defined as $\langle d^1, d^2 \rangle := \sum_{i=1}^n d_i^1 d_i^2$. The convex hull of a set $\mathcal{X}$ is denoted by $\text{conv}(\mathcal{X})$, and the Cartesian product of sets is represented by $\prod$. Furthermore, we use $\lceil \cdot \rceil$ to denote the ceiling function, and $\lfloor \cdot \rfloor$ to denote the floor function.

2 Model Description

We consider a company serving many geographically dispersed customers who place integer-valued demands for multiple products. We divide the geographical area into smaller areas and consider the total demand from each small area as if it is demand from one customer. So, this customer is an "aggregate customer".

The customers are numbered from 1 to J, and $\mathcal{J}$ denotes the set of customers. The products are numbered from 1 to K, and $\mathcal{K}$ denotes the set of products. 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 $1, \cdots, I$ 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 lead-times 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 satisfy demands from customers directly. For the central warehouse, we use index 0. Each period consists of short time-slots numbered $1, \cdots, T$; let $\mathcal{T} := \{1, \cdots, T\}$. For each product 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 from demands in other time-slots and other periods, and belong to the bounded uncertainty set $\mathcal{D}_t \subseteq \mathbb{N}_0^{JK}$.

Customers place their orders during a time-slot. For a unit of product 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). At the end of a time-slot, the company decides for each demanded unit of product k by customer j which warehouse satisfies this demanded unit. Next, the part is delivered by a fast shipment. If the unit can be delivered from a sufficiently nearby local warehouse, the delivery time target will be met. Otherwise, the target is exceeded, and a penalty cost is paid per unit of tardiness. 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 in [53] and [58], that the shipment cost from a location i to customer j for product 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 product k is given by c_{ijk}^d and forms the first part of $\bar{c}_{ijk}$. The second part is formed by a unit penalty cost c_{jk}^p multiplied by the tardiness when the shipment from location i to customer j is not fast enough to meet the delivery time target $t_{jk}^{\max}$. This leads to the following formula for $\bar{c}_{ijk}$:

$$\bar{c}_{ijk} = c_{ijk}^d + c_{jk}^p \left(t_{ij} - t_{jk}^{\max}\right)^+, \qquad i \in \mathcal{I}_0, j \in \mathcal{J}, k \in \mathcal{K},$$

where $\mathcal{I}_0 := \mathcal{I} \cup \{0\}$.

Apart from shipment costs, we also have costs for opening local warehouses, inventory holding costs, and replenishment costs. Let F_i be the cost for opening a local warehouse at candidate location i. We assume that a *base stock*, also known as the *order up to level*, policy is followed by each local warehouse for each of the products and that a unit inventory holding cost h_{ik} is charged for on-hand stock of product 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. At the beginning of the time horizon, the replenishment quantity equals the base stock level of local warehouses. From the second period onwards, the total number of goods transported from the central warehouse to the local warehouses per period is equal to the total fulfilled demand in the previous period due to the base stock policy logic.

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.

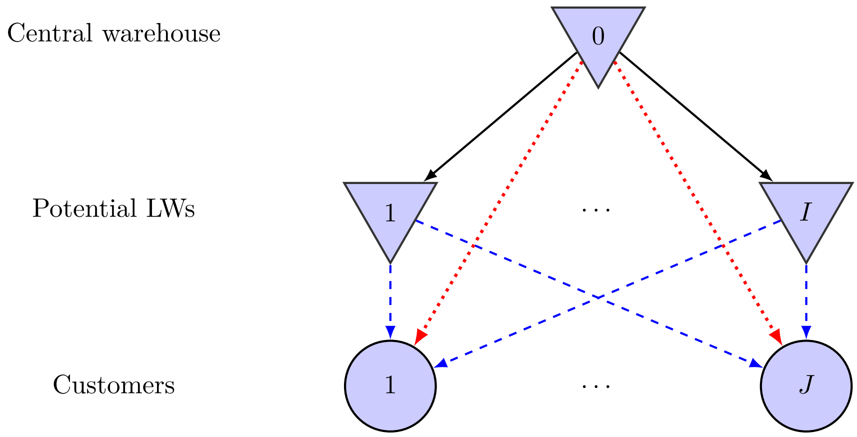


Figure 1 The schematic illustration of an LT problem with one central warehouse, I candidate locations for the local warehouses (LWs), and J customers. The solid (black), dashed (blue), and dotted (red) arrows are depicted to show the replenishment of the LWs, possible shipments from LWs to each customer,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

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 base stock 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 product 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 $\mathbb{N}_0^{JKTL}$ to $\mathbb{N}_0$.

3 Multi-Stage Adjustable Robust Optimization Formulation

In this section, we provide the formulation of the multi-stage adjustable robust LT problem with T time-slots and L periods. 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 base stock levels S_{ik}, $i \in \mathcal{I}$ and $k \in \mathcal{K}$, are made. Then, the holding cost is paid at the beginning of the first time-slot. During the first time-slot, d_{jk1} is realized. At the end of the first time-slot, z_{ijk1}, $i \in \mathcal{I}_0$, $j \in \mathcal{J}$, $k \in \mathcal{K}$, is chosen based on the realized demand. Then, the on-hand inventory is calculated and the holding cost is incurred. During the second time-slot, demand d_{jk2} is realized and at the end of this time-slot the shipment quantities z_{ijk2}, $i \in \mathcal{I}_0$, $j \in \mathcal{J}$, and $k \in \mathcal{K}$, are chosen. The sequence of incurred holding cost, demand realization, and decision making on shipment quantities continues until the end of time-slot T, after which the warehouses are replenished and the same sequence is observed.

Table 1 Nomenclature.

Notation	Description
Set	
$\mathcal{J}$	Set of J customers, indexed by j
$\mathcal{I}$	Set of I candidate locations for local warehouses, indexed by i, i'; central warehouse is denoted by 0
$\mathcal{K}$	Set of K different items, indexed by k
$\mathcal{T}$	Set of T time-slots between two replenishments, indexed by $t, \tilde{t}$
$\mathcal{D}_t$	Uncertainty set in time-slot t
Parameters	
L	Number of periods in the time horizon
t_{ij}	Delivery time of a shipment from the local warehouse i to customer j
r_i^{fix}	Fixed replenishment cost of local warehouse i
r_{ik}^{unit}	Unit transportation cost of replenishing local warehouse i for item k
$t_{jk}^{\max}$	Delivery time target of customer j for item k
c_{ijk}^d	Unit delivery cost of item k from local warehouse i to customer j
c_{jk}^p	Unit penalty cost of tardiness of item k for customer j
$\bar{c}_{ijk}$	Unit shipment cost from local warehouse i to customer j for item k at the end of time-slot t, calculated by $c_{ijk}^d + c_{jk}^p \left(t_{ij} - t_{jk}^{\max}\right)^+$
c_{ijk}	$\bar{c}_{ijk} + r_{ik}^{\text{unit}}$
F_i	Cost of opening a local warehouse at candidate location i
f_i	Aggregated fixed costs to open local the fixed cost of opening the local warehouse i scaled to one period, calculated by $f_i = \frac{F_i + L.r_i^{\text{fix}}}{L}$
h_{ik}	Unit inventory holding cost of item k in the local warehouse i
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
$S_{ik} \in \mathbb{N}_0$	Base stock level of item k in the local warehouse i
$z_{ijkt} \in \mathbb{N}_0$	Shipment quantity sent from local\central warehouse i of item k to customer j at the end of time-slot t of a time period

Note. The term “local warehouse i” refers to the potential local warehouse in the i-th candidate location.

In this paper, we aim to solve the following problem

$$\begin{aligned} \min_{y,S} \quad & \sum_{i\in\mathcal{I}} \left(F_i + L.r_i^{\text{fix}}\right) y_i + \sum_{\substack{i\in\mathcal{I}\\ k\in\mathcal{K}}} r_{ik}^{\text{unit}} S_{ik} + \sum_{\ell=1}^{L} H_{\mathcal{D}}^{\ell}(y,S), \\ \text{s.t.} \quad & S_{ik} \le M y_i, \qquad i\in\mathcal{I}, k\in\mathcal{K}, \\ & S_{ik} \in \mathbb{N}_0, \qquad i\in\mathcal{I}, k\in\mathcal{K}, \\ & y_i \in \{0,1\}, \qquad i\in\mathcal{I}, \end{aligned} \tag{1}$$

where the first term of the objective function is the aggregation of fixed opening cost as well as the fixed replenishment cost, the second term is cost of the first replenishment, and the last term is the aggregation of the worst-case cost incurred in each period, including replenishment, inventory holding, and delivery costs, with $H_{\mathcal{D}}^{\ell}(y,S)$ being the worst-case cost of period ℓ given the uncertainty set $\mathcal{D} = \prod_{t=1}^{T} \mathcal{D}_t$, and the values of y and S.

Because of the stationary periodic demand assumption, the worst-case cost during the first period, $H^1_{\mathcal{D}}(y,S)$, is the same as the one during the second period $H^2_{\mathcal{D}}(y,S)$, $\cdots$, and the same as the one during the L-th period, $H^L_{\mathcal{D}}(y,S)$. Therefore, (1) is equivalent to

$$
\begin{aligned}
\min_{y,S} \quad & \sum_{i\in\mathcal{I}} \left(F_i + L.r_i^{\text{fix}}\right) y_i + \sum_{\substack{i\in\mathcal{I}\\ k\in\mathcal{K}}} r_{ik}^{\text{unit}} S_{ik} + L.H^1_{\mathcal{D}}(y,S) \\
\text{s.t.} \quad & S_{ik} \le M y_i, && i\in\mathcal{I}, k\in\mathcal{K}, \\
& S_{ik} \in \mathbb{N}_0, && i\in\mathcal{I}, k\in\mathcal{K}, \\
& y_i \in \{0,1\}, && i\in\mathcal{I}.
\end{aligned} \tag{2}
$$

This shows that, without loss of generality, we can reduce the problem of minimizing the total cost during the whole time horizon to the problem of minimizing the scaled fixed costs of opening local warehouses plus one-time fixed replenishment cost, the scaled one-time replenishment shipment cost, and the worst-case holding and shipment costs during one period.

Therefore, we remove the ℓ index from our notation and introduce $f_i = \frac{F_i + L.r_i^{\text{fix}}}{L}$ as the scaled fixed cost for location i. Furthermore, the scaled one-time replenishment shipment cost is negligible compared to the cost of one period. Therefore, we can limit ourselves to only one period and solve the following problem:

$$
\begin{aligned}
LT(\mathcal{D}) := \min_{y,S} \; & \sum_{i\in\mathcal{I}} f_i y_i + H^1_{\mathcal{D}}(y,S) \\
\text{s.t.} \; & S_{ik} \le M y_i, && i\in\mathcal{I}, k\in\mathcal{K}, \\
& S_{ik} \in \mathbb{N}_0, && i\in\mathcal{I}, k\in\mathcal{K}, \\
& y_i \in \{0,1\}, && i\in\mathcal{I}.
\end{aligned} \tag{3}
$$

The summary of the notation used in this paper is presented in Table 1.

Let us now focus only on a period, and let $d_{[t]} \in \mathbb{N}_0^{JKt}$ denote the vector of demands realized until the end of time-slot t in the period with $\mathcal{D}_{[t]} = \prod_{\tilde{t}=1}^{t} \mathcal{D}_{\tilde{t}} \subseteq \mathbb{N}_0^{JKt}$ being the uncertainty set with respect to $d_{[t]}$. Furthermore, we denote by $z_{[t]}$ the vector containing the shipment quantities from different warehouses (local and central) to customers for different products 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 $z_{[0]} := 0_{JK}$.

At the end of time-slot T (hence the end of the period), we need to make the optimal decision based on the realization of demands in the whole period, $d_{[T]}$, while knowing which warehouses are opened, y, what the base stock levels are in the opened warehouses, S, and how many units have been shipped in the previous time-slots, $z_{[T-1]}$. Hence, we need to solve the following optimization problem at the end of time-slot T:

$$
\begin{aligned}
Q_T(y,S,d_{[T]},z_{[T-1]}) := \min_{z_T} \; & \sum_{\substack{i\in\mathcal{I}_0\\ j\in\mathcal{J}\\ k\in\mathcal{K}}} c_{ijk} z_{ijkT} \\
\text{s.t.} \; & \sum_{j\in\mathcal{J}} z_{ijkT} \le S_{ik} - \sum_{j\in\mathcal{J}} \sum_{t=1}^{T-1} z_{ijkt}, && i\in\mathcal{I}, k\in\mathcal{K}, \\
& \sum_{i\in\mathcal{I}_0} z_{ijkT} = d_{jkT}, && j\in\mathcal{J}, k\in\mathcal{K}, \\
& z_{ijkT} \in \mathbb{N}_0, && i\in\mathcal{I}_0, j\in\mathcal{J}, k\in\mathcal{K},
\end{aligned} \tag{4}
$$

where the objective function is to minimize the shipment cost, the first set of constraints is to make sure that the total quantities shipped from a warehouse do not exceed its on-hand inventory at the beginning of time-slot T, and the second set of constraints ensures that the demands are satisfied by shipments from either central warehouse or local ones.

Inductively, if we know that $\max\{Q_{t+1}\left(y, S, d_{[t+1]}, z_{[t]}\right): d_{t+1} \in \mathcal{D}_{t+1}\}$ provides us with the worst cost that would incur from the end of time-slot $t+1$ until the end of the period, then during the time-slot t, we have:

$$\begin{aligned} Q_t\left(y, S, d_{[t]}, z_{[t-1]}\right) = \min_{z_t} \;& \sum_{\substack{i\in\mathcal{I}_0\\ j\in\mathcal{J}\\ k\in\mathcal{K}}} c_{ijk} z_{ijkt} + \sum_{\substack{i\in\mathcal{I}\\ k\in\mathcal{K}}} h_{ik}\left(S_{ik} - \sum_{j\in\mathcal{J}}\sum_{\tilde{t}=1}^{t-1} z_{ijk\tilde{t}}\right) \\ &+ \max_{d_{t+1}\in\mathcal{D}_{t+1}} Q_{t+1}\left(y, S, d_{[t+1]}, z_{[t]}\right) \\ \text{s.t.} \;& \sum_{j\in\mathcal{J}} z_{ijkt} \le S_{ik} - \sum_{j\in\mathcal{J}}\sum_{\tilde{t}=1}^{t-1} z_{ijk\tilde{t}}, \quad i\in\mathcal{I}, k\in\mathcal{K}, \\ & \sum_{i\in\mathcal{I}_0} z_{ijkt} = d_{jkt}, \quad j\in\mathcal{J}, k\in\mathcal{K}, \\ & z_{ijkt}\in\mathbb{N}_0, \quad i\in\mathcal{I}_0, j\in\mathcal{J}, k\in\mathcal{K}. \end{aligned} \tag{5}$$

Given the above formulations and (3), we can formulate a multi-stage adjustable robust LT problem as

$$\begin{aligned} LT(\mathcal{D}_{[T]}) := \min_{y,S} \;& \sum_{i\in\mathcal{I}} f_i y_i + \sum_{\substack{i\in\mathcal{I}\\ k\in\mathcal{K}}} h_{ik} S_{ik} + \max_{d_1\in\mathcal{D}_1} Q_1(y, S, d_{[1]}, 0_{JK}) \\ \text{s.t.} \;& S_{ik} \le M y_i, \quad i\in\mathcal{I}, k\in\mathcal{K}, \\ & S_{ik}\in\mathbb{N}_0, \quad i\in\mathcal{I}, k\in\mathcal{K}, \\ & y_i\in\{0,1\}, \quad i\in\mathcal{I}, \end{aligned} \tag{6}$$

where the objective function is the total worst-case cost, and the first constraint makes sure that the base stock levels of the closed local warehouses are zero, with M being a large enough number.

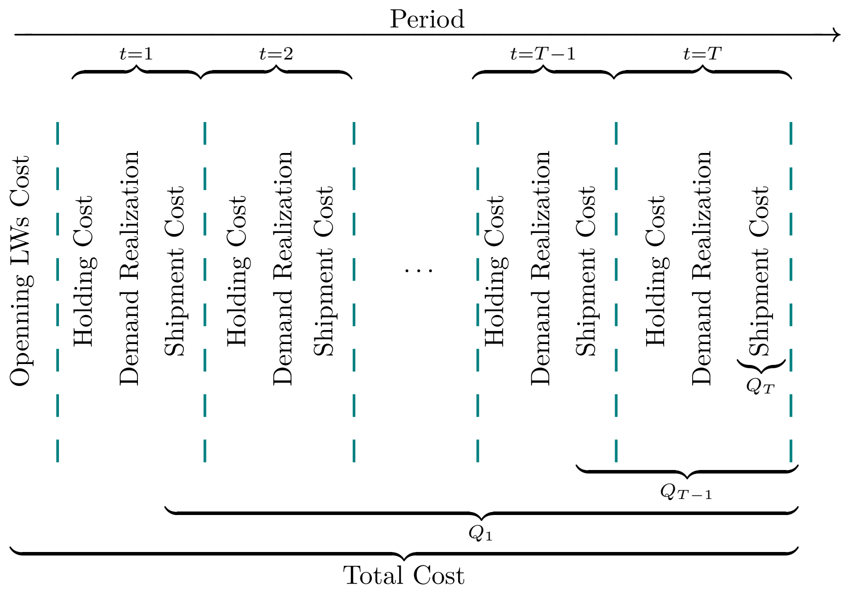


Figure 2 Illustration of the total cost that is split into the costs in each time-slot.

Figure 2 shows how the costs functions $Q_t(y, S, d_{[t]}, z_{[t-1]})$, $t = 1, \cdots, T$, 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.

Remark 1. *In (5), the decision at time-slot t, depends implicitly on both the realized demand up to time-slot t and the current inventory level. This dependency can be explicitly added to the problem by introducing a new variable I_{ikt} to denote the inventory level at the beginning of time-slot t with an extra inventory balance constraint*

$$I_{ikt} = I_{ikt-1} - \sum_{j\in\mathcal{J}} z_{ijkt}, \quad \forall i\in\mathcal{I}, t\in\mathcal{T}, k\in\mathcal{K},$$

with $I_{ik0} = S_{ik}$ for any $i \in \mathcal{I}$ and $k \in \mathcal{K}$. We emphasize that this formulation results in an equivalent problem, since we have

$$I_{ikt} = S_{ik} - \sum_{j \in \mathcal{J}} \sum_{\tilde{t}=1}^{t} z_{jik\tilde{t}} \quad \forall i \in \mathcal{I}, t \in \mathcal{T}, k \in \mathcal{K},$$

which is what we have considered in the objective function as well as the first constraint of (5).

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

4 The Importance of the Demand Integrality Assumption

An assumption we have made is that the demand is integer, whose importance may not be clear. Therefore, in this section, we provide an example of an LT problem for which relaxing the integrality restriction can result in a solution whose strategic and tactical decisions are far from optimal.

Let $J \in \mathbb{N}$ be larger than 2. Let us fix $\alpha \geq 0$ and let $0 < \epsilon < 1$ be such that $\epsilon J \in \mathbb{N}$. We consider an LT problem to satisfy the demands of J customers with one potential location to open a warehouse, indexed 1, within a time period with one time-slot, i.e., $T = 1$. We assume the cost of opening a warehouse is $f_1 = 4\alpha$, the unit holding cost is $h_{11} = \frac{\alpha}{10{,}000}$, the unit shipment cost from the potential location 1 to customer j is $c_{1j1} = 0.1\alpha$, and the unit shipment cost from the central warehouse to customer j is $c_{0j1} = \alpha$. We assume that the uncertainty set constructed from historical data is

$$\mathcal{D}_1 = \left\{ [d_{j11}]_{j \in \mathcal{J}} \in \mathbb{N}_0^J : \ n - 2 \leq d_{j11} \leq n - \epsilon, \quad \forall j \in \mathcal{J} \right\},$$

where $n \in \mathbb{N}$, and $n \geq 2$. In this example, we solve the LT problem with and without the integrality restriction in $\mathcal{D}_1$ to better understand the effect of such restrictions on the optimal solutions. Lifting the integrality restriction means the uncertainty set is

$$\overline{\mathcal{D}}_1 = \left\{ [d_{j11}]_{j \in \mathcal{J}} \in \mathbb{R}^J : \ n - 2 \leq d_{j11} \leq n - \epsilon, \quad \forall j \in \mathcal{J} \right\},$$

and the shipment quantities, z_{1j11} and z_{0j11}, $j \in \mathcal{J}$, can be non-negative real-values. Based on the results we will show in Section 5, we know that the worst-case scenario occurs at $d = (n - \epsilon)e$, where $e \in \mathbb{R}^J$ is the vector of all ones. Therefore,

- if $J < \frac{4}{0.8999(n-\epsilon)}$ then the solution is to open no warehouse with the cost of $J(n - \epsilon)\alpha$,
- if $J \geq \frac{4}{0.8999(n-\epsilon)}$, then the solution is to open the warehouse and store $J(n - \epsilon)$ number of units, with the cost of $4\alpha + 0.1001J(n - \epsilon)\alpha$.

Now, let us consider the problem with the integrality restriction. Given $\mathcal{D}_1$, it is clear that the worst-case scenario for the problem is when the demand of each customer is $n - 1$. Therefore,

- if $J < \frac{4}{0.8999(n-1)}$ then the solution is to open no warehouse with the cost of $J(n - 1)\alpha$,
- if $J \geq \frac{4}{0.8999(n-1)}$, then the solution is to open the warehouse and store $J(n - 1)$ number of units, with the cost of $4\alpha + 0.1001J(n - 1)\alpha$.

So, for some values of n there exists $J \in \mathbb{N}$, for which the strategic solution (whether to open a warehouse or not) provided by the relaxed problem is different than the one from the integer problem. For example, for $n = 2$, $J = 4$, and $\epsilon = 0.5$, the relaxed problem gives the solution of opening the warehouse and stocking 6 units with the cost of 4.6006α while the optimal solution is to open no warehouse with the cost of 4α.

To further illustrate the impact of the integrality assumption, we analyze scenarios varying the number of customers (J) and the base demand parameter n, with $\epsilon = 0.5$. Table 2 compares the optimal cost from the relaxed model with the optimal cost when the integrality assumption is considered. As the table shows and is expected, a large gap exists between the solutions of the models when n is small. This arises because the relaxed model overestimates worst-case demand ($n - \epsilon$ instead of $n - 1$), leading to excessive inventory and higher costs that is more significant when n is relatively small.

5 Exact Methods to Solve the ARO Formulation

The main focus of this paper is on how problem (6) can be solved. In this section, we provide theoretical results with which we prove tractability of (6) for some classes of uncertainty sets. In other words, we show, for some

Table 2 Impact of Demand Integrality Relaxation on Solution Quality

n	J	**Relaxed Model**	**Integer Model**	**Gap (%)**
	4	4.6006α	4α	15.02
2	10	5.5015α	5.001α	10.01
	100	19.015α	14.01α	35.72
	4	7.8038α	7.6036α	2.63
10	10	13.5095α	13.009α	3.85
	100	99.095α	94.09α	5.32
	4	43.8398α	43.6396α	0.46
100	10	103.5995α	103.099α	0.49
	100	999.995α	994.99α	0.51

classes of uncertainty sets, that an optimal solution of the ARO formulation (6) can be obtained by limiting the uncertainty set to a polynomial number of scenarios. In the rest of this section, we first show the main theoretical results of the paper and then provide methods to solve (6) exactly using the theoretical results.

5.1 Convexity and Non-Decreasingness of $Q_t\left(y, S, d_{[t]}, z_{[t-1]}\right)$ in d_t

We first show for $t = 1, \cdots, T$, that the function $Q_t\left(y, S, d_{[t]}, z_{[t-1]}\right)$ is convex in d_t, where the convexity of function $g(\cdot)$ over $\mathbb{N}_0^n$ is defined as follows: for any $d^1, d^2 \in \mathbb{N}_0^n$ and $\lambda \in (0,1)$, if $\lambda d^1 + (1-\lambda)d^2 \in \mathbb{N}_0^n$, then

$$g(\lambda d^1 + (1-\lambda)d^2) \leq \lambda g(d^1) + (1-\lambda)g(d^2).$$

Theorem 1. *Let $y \in \{0,1\}^I$ and $S \in \mathbb{N}_0^{IK}$ be given. Also, let $t = 1, \cdots, T$, be fixed and $z_{[t-1]}$ be a given vector of shipments for up to and including time-slot $t-1$. Then, $Q_t\left(y, S, d_{[t]}, z_{[t-1]}\right)$ is convex in $d_t \in \mathbb{N}_0^{JK}$.*

Proof. We prove this theorem using backward induction. As the induction hypothesis, we show that the theorem holds for $\bar{t} = T$. Then, using inductive steps, we show that if the theorem holds for $\bar{t} = t+1, \cdots, T$, it also holds for $\bar{t} = t$.

Induction hypothesis: Let us denote by $\widetilde{Q}_T(y, S, d_{[T]}, z_{[T-1]})$ the continuous relaxation of $Q_T(y, S, d_{[T]}, z_{[T-1]})$, where the integrality restriction on the shipments quantities is replaced by $z_{ijkT} \geq 0$. As discussed in [12, Section 10.2], any basic feasible solution of $\widetilde{Q}$ corresponds to a totally unimodular submatrix of the coefficient matrix. Therefore, the vertices of the feasible region of $\widetilde{Q}_T(y, S, d_{[T]}, z_{[T-1]})$ are integral if $d_T \in \mathbb{N}_0^{JK}$. Hence, if $d_T \in \mathbb{N}_0^{JK}$, then there exists an optimal solution of $\widetilde{Q}_T(y, S, d_{[T]}, z_{[T-1]})$ that is optimal for $Q_T(y, S, d_{[T]}, z_{[T-1]})$, and we have

$$\widetilde{Q}_T(y, S, d_{[T]}, z_{[T-1]}) = Q_T(y, S, d_{[T]}, z_{[T-1]}).$$

Let $d^1, d^2 \in \mathbb{N}_0^{JK}$, $\lambda \in (0,1)$ such that $\lambda d^1 + (1-\lambda)d^2 \in \mathbb{N}_0^{JK}$. Then, by Lemma EC.2 of [3], we have

$$\widetilde{Q}_T(y, S, \lambda d^1 + (1-\lambda)d^2, z_{[T-1]}) \leq \lambda \widetilde{Q}_T(y, S, d^1_{[T]}, z_{[T-1]}) + (1-\lambda)\widetilde{Q}_T(y, S, d^2_{[T]}, z_{[T-1]}).$$

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

$$Q_T(y, S, \lambda d^1 + (1-\lambda)d^2, z_{[T-1]}) \leq \lambda Q_T(y, S, d^1_{[T]}, z_{[T-1]}) + (1-\lambda)Q_T(y, S, d^2_{[T]}, z_{[T-1]}),$$

which shows that $Q_T(y, S, d_{[T]}, z_{[T-1]})$ is convex.

Inductive steps: Let us assume that the theorem holds for $\bar{t} = t+1, \cdots, T$. Therefore, we have

$$\begin{aligned}
Q_t\left(y, S, d_{[t]}, z_{[t-1]}\right) = \min_{z_t, \tau} \; & \sum_{\substack{i\in\mathcal{I}_0\\ j\in\mathcal{J}\\ k\in\mathcal{K}}} c_{ijk} z_{ijkt} + \sum_{\substack{i\in\mathcal{I}\\ k\in\mathcal{K}}} h_{ik}\left(S_{ik} - \sum_{j\in\mathcal{J}}\sum_{\tilde{t}=1}^{t} z_{ijk\tilde{t}}\right) + \tau \\
\text{s.t.} \; & Q_{t+1}\left(y, S, d_{[t+1]}, z_{[t]}\right) \leq \tau, && \forall d_{t+1} \in \Omega_{t+1}, \\
& \sum_{j\in\mathcal{J}} z_{ijkt} \leq S_{ik} - \sum_{j\in\mathcal{J}}\sum_{\tilde{t}=1}^{t-1} z_{ijk\tilde{t}}, && i \in \mathcal{I}, k \in \mathcal{K}, \\
& \sum_{i\in\mathcal{I}_0} z_{ijkt} = d_{jkt}, && j \in \mathcal{J}, k \in \mathcal{K}, \\
& z_{ijkt} \in \mathbb{N}_0, && i \in \mathcal{I}_0, j \in \mathcal{J}, k \in \mathcal{K},
\end{aligned} \tag{7}$$

where Ω_{t+1} is the set of corner points (vertices) of $\mathcal{D}_{t+1}$. Thus, we can rewrite Q_t in (7) as

$$\begin{aligned}
Q_t\left(y,S,d_{[t]},z_{[t-1]}\right) = \min_{z_t,\tau} \; & \sum_{\substack{i\in\mathcal{I}_0\\ j\in\mathcal{J}\\ k\in\mathcal{K}}} c_{ijk} z_{ijkt} + \sum_{\substack{i\in\mathcal{I}\\ k\in\mathcal{K}}} h_{ik}\left(S_{ik} - \sum_{j\in\mathcal{J}}\sum_{\tilde{t}=1}^{t} z_{ijk\tilde{t}}\right) + \tau \\
\text{s.t.} \; & \sum_{\hat{t}=t+1}^{T} g_{\hat{t}}^{\ell} \leq \tau, && \ell = 1,\cdots,L, \\
& \sum_{j\in\mathcal{J}} z_{ijk\hat{t}}^{\ell} \leq S_{ik} - \sum_{j\in\mathcal{J}}\sum_{\tilde{t}=1}^{\hat{t}-1} z_{ijk\tilde{t}}^{\ell}, && \begin{cases} i\in\mathcal{I}, \ell=1,\cdots,L, \\ k\in\mathcal{K}, \hat{t}=t+1,\cdots,T, \end{cases} \\
& \sum_{i\in\mathcal{I}_0} z_{ijk\hat{t}}^{\ell} = d_{jk\hat{t}}^{\ell}, && \begin{cases} j\in\mathcal{J}, \ell=1,\cdots,L, \\ k\in\mathcal{K}, \hat{t}=t+1,\cdots,T, \end{cases} \\
& \sum_{j\in\mathcal{J}} z_{ijkt} \leq S_{ik} - \sum_{j\in\mathcal{J}}\sum_{\tilde{t}=1}^{t-1} z_{ijk\tilde{t}}, && i\in\mathcal{I}, k\in\mathcal{K}, \\
& \sum_{i\in\mathcal{I}_0} z_{ijkt} = d_{jkt}, && j\in\mathcal{J}, k\in\mathcal{K}, \\
& z_{ijkt} \in \mathbb{N}_0, && i\in\mathcal{I}_0, j\in\mathcal{J}, k\in\mathcal{K}, \\
& z_{ijkt}^{\ell} \in \mathbb{N}_0, && \begin{cases} i\in\mathcal{I}_0, j\in\mathcal{J}, k\in\mathcal{K}, \\ \ell=1,\cdots,L, \\ \hat{t}=t+1,\cdots,T, \end{cases}
\end{aligned} \tag{8}$$

where

$$g_{\hat{t}}^{\ell} := \left(\sum_{\substack{i\in\mathcal{I}_0\\ j\in\mathcal{J}\\ k\in\mathcal{K}}} c_{ijk} z_{ijk\hat{t}}^{\ell} + \sum_{\substack{i\in\mathcal{I}\\ k\in\mathcal{K}}} h_{ik}\left(S_{ik} - \sum_{j\in\mathcal{J}}\sum_{\tilde{t}=1}^{\hat{t}} z_{ijk\tilde{t}}^{\ell}\right)\right),$$

L is the number of points in $\prod_{\hat{t}=t+1}^{T}\Omega_{\hat{t}}$, whose points are denoted by $\left[d_{[jk\hat{t}]}^{\ell}\right]_{\substack{j\in\mathcal{J}\\ k\in\mathcal{K}\\ \hat{t}=t+1,\cdots,T}}$. Also, $z_{ijk\hat{t}}^{\ell}$ denotes the shipment at time-slot $\hat{t}$ from location i to customer j for product k when demand at time-slots $t+1$ until T is the corner point indexed by ℓ. Now, let us consider the L points in $\left\{d_{[t]}\right\} \times \prod_{\hat{t}=t+1}^{T}\Omega_{\hat{t}}$ and denote them by $\left[\bar{d}_{[jk\hat{t}]}^{\ell}\right]_{\substack{j\in\mathcal{J}\\ k\in\mathcal{K}\\ \hat{t}=t,\cdots,T}}$. In other words, for each ℓ, we put $d_{[t]}$ as the first block of $\left[\bar{d}_{[jk\hat{t}]}^{\ell}\right]_{\substack{j\in\mathcal{J}\\ k\in\mathcal{K}\\ \hat{t}=t,\cdots,T}}$. Then, one can see that (8) can be rewritten as

$$\begin{aligned}
Q_t\left(y,S,d_{[t]},z_{[t-1]}\right) = \min_{z_t,\tau} \; & \sum_{\substack{i\in\mathcal{I}_0\\ j\in\mathcal{J}\\ k\in\mathcal{K}}} c_{ijk} z_{ijkt} + \sum_{\substack{i\in\mathcal{I}\\ k\in\mathcal{K}}} h_{ik}\left(S_{ik} - \sum_{j\in\mathcal{J}}\sum_{\tilde{t}=1}^{t} z_{ijk\tilde{t}}\right) + \tau \\
\text{s.t.} \; & \sum_{\hat{t}=t+1}^{T} g_{\hat{t}}^{\ell} \leq \tau, && \ell = 1,\cdots,L, \\
& \sum_{j\in\mathcal{J}}\sum_{\tilde{t}=1}^{\hat{t}} z_{ijk\tilde{t}}^{\ell} \leq S_{ik}, && \begin{cases} i\in\mathcal{I}, \ell=1,\cdots,L, \\ k\in\mathcal{K}, \hat{t}=t,\cdots,T, \end{cases} \\
& \sum_{i\in\mathcal{I}_0} z_{ijk\hat{t}}^{\ell} = \bar{d}_{jk\hat{t}}^{\ell}, && \begin{cases} j\in\mathcal{J}, \ell=1,\cdots,L, \\ k\in\mathcal{K}, \hat{t}=t,\cdots,T, \end{cases} \\
& z_{ijkt}^{\ell} \in \mathbb{N}_0, && \begin{cases} j\in\mathcal{J}, \ell=1,\cdots,L, \\ k\in\mathcal{K}, \hat{t}=t,\cdots,T, \end{cases}
\end{aligned} \tag{9}$$

where $z_{ijkt}^{\ell} = z_{ijkt}$. Let us consider the continuous relaxation of (9). If we show that the relaxation has an integer-valued optimal solution, then similar to the proof of the induction hypothesis, we can prove that the inductive steps are valid.

Let us consider the coefficient matrix of the relaxation of (9) after rewriting it in the standard form:

$$\begin{matrix} \tau & z & s^1 & s^2 \end{matrix}$$
$$\begin{bmatrix} -\mathbb{1} & A & I & 0 \\ 0 & B & 0 & I \\ 0 & C & 0 & 0 \end{bmatrix},$$

where z denotes the vector containing all $z^{\ell}_{ijk\hat{t}}$, $i \in \mathcal{I}_0$, $j \in \mathcal{J}$, $k \in \mathcal{K}$, $\ell = 1, \cdots, L$, and $\hat{t} = t, \cdots, T$, s^1 and s^2 denote the vector of slack variables with respect to the first and second set of constraints, respectively. Also, $\mathbb{1} \in \mathbb{R}^L$ is the vector of ones corresponding to the coefficient of τ, I is the identity matrix with the suitable size corresponding to the slack variables, and A, B, and C are the coefficient of $z^{\ell}_{ijk\hat{t}}$ at the first, second, and third set of constraints, respectively.

Let $(\tau^*, z^*, s^{1*}, s^{2*})$ be an optimal corner point of the relaxed problem. If

$$\left[\bar{d}^{\ell}_{[jk\hat{t}]}\right]_{\substack{j\in\mathcal{J}\\ k\in\mathcal{K}\\ \hat{t}=t,\cdots,T}} = 0, \quad \forall \ell = 1, \cdots, L,$$

then clearly $z = 0$. Otherwise, we know that $\tau^* > 0$ and hence it is a basic variable.

Let us now consider two cases:

Case I: There is only one demand scenario in $\prod_{\hat{t}=1}^{T} \Omega_{\hat{t}}$ that is the worst-case. This implies that among the constraints in the first set of constraints in (9) only one active constraint exists. Thus, $L-1$ components of s^1 are non-zeros, hence are basic variables. Let us assume that ℓ is the index where $s_{1_\ell} = 0$. Then, the submatrix containing the columns related to the basic variables is

$$D = \begin{pmatrix} -\mathbb{1} & \bar{A} & I \setminus \{e_\ell\} & 0 \\ 0 & \bar{B} & 0 & \bar{I} \\ 0 & \bar{C} & 0 & 0 \end{pmatrix},$$

where $[\bar{A}^\top \ \bar{B}^\top \ \bar{C}^\top]^\top$ is the collection of columns related to components of z that are basic, $I \setminus \{e_\ell\}$ is the identity matrix excluding the ℓ-th column, and $\bar{I}$ is the collection of columns of identity matrix related to the basic component of s^2. Due to having columns of identity matrix, determinant of D is either 1 or -1 times the determinant of

$$\bar{D} = \begin{pmatrix} -1 & \bar{a} & 0 \\ 0 & \bar{B} & \bar{I} \\ 0 & \bar{C} & 0 \end{pmatrix},$$

where $\bar{a}$ is the ℓ-th row of $\bar{A}$. Moreover, the determinant of $\bar{D}$ is the same as -1 times the determinant of $\begin{pmatrix} \bar{B} & \bar{I} \\ \bar{C} & 0 \end{pmatrix}$, which is a totally unimodular matrix because of the structure of (9) (as discussed in [12, Section 10.2]). Therefore, D is a unimodular matrix.

Case II: There exist multiple worst-case demand scenarios in $\prod_{\hat{t}=1}^{T} \Omega_{\hat{t}}$. Let us denote the number of worst-case scenarios by $\bar{L}$. Then, $L - \bar{L}$ of components of s^1 are non-zero and the other $\bar{L}$ are zero. So, the non-zero components of s^1 are basic. Moreover, in this case, we have $\bar{L} - 1$ redundant constraints related to the worst-case scenarios. Therefore, there exists a collection of basic variables containing all components of s^1 except one. So, the columns related to basic variables can be represented by D, which is a unimodular matrix.

Since, b is an integer-valued vector, and we have shown that the matrix containing the columns related to the basic variables is unimodular, there exists an optimal solution of the relaxation of (9) that has integer-valued z. This concludes the theorem. ◀

Based on Theorem 1, 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, the here-and-now optimal decisions for problem (6) can be obtained by solving a deterministic problem with a polynomial (in I and J) number of constraints.

Another important use of Theorem 1 is in reformulating (6) as a multi-stage adjustable LT problem where the demands as well as the shipment quantities are continuous.

Proposition 1. *A point (y^*, S^*) is an optimal solution of (6) 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 quantities is relaxed and the uncertainty set is replaced with its convex hull.*

Proof. Let us denote the optimal value of the continuous relaxation by $\widetilde{LT}(\mathcal{D}_{[T]})$. 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 $\widetilde{Q}_t$ 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 $\widetilde{Q}_t(y, S, d_{[t]}, z_{t-1})$ is integer-valued. Hence, we have

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

where $\equiv$ means the problems have the same optimal value and optimal "here-and-now" solutions. ◀

Proposition 1 asserts that continuous relaxation of (6) can be used to obtain the optimal value as well as the "here-and-now" part of optimal solutions. Therefore, if an explicit description of the convex hull of the uncertainty set is available, then one can use exact methods for ARO problems with continuous uncertain parameters and "wait-and-see" variables to solve the continuous relaxation of (6); see e.g., [17, 25, 77].

Another important property of $Q_t\left(y, S, d_{[t]}, z_{[t-1]}\right)$, next to its convexity in d_t, is that it is non-decreasing in d_t for a class of problems, which is shown 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 $d^1, d^2 \in \mathcal{D}_1$ be such that $d^1 \lneq d^2$, then*

$$Q_T(y, S, d^1_{[T]}, 0_{JK}) \leq Q_T(y, S, d^2_{[T]}, 0_{JK}).$$

Proof. Clearly, due to the ample stock of the central warehouse, (4) is always feasible. Let us denote by $z^*_{ijk1}(d)$ an optimal shipment of (4), given the demand vector d, location variable y_i, and stock level S_{ik}, $i \in \mathcal{I}$, $k \in \mathcal{K}$. For simplicity, let us denote by $Q(y, S, d^1)$ the optimal value of (4). 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

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

Since $d^1 \lneq d^2$, $\overline{\mathcal{JK}}$ is not empty. Furthermore, let $\{i_0, i_1, \cdots, i_I\} := \mathcal{I}_0$, and for any $j \in \mathcal{J}$, $k \in \mathcal{K}$, set

$$\mathcal{I}(j,k) := \left\{i \in \mathcal{I}_0 : \ z^*_{ijk1}(d^2) > 0\right\}.$$

For any $(j,k) \in \overline{\mathcal{JK}}$ and $\ell = 0, \cdots, I$, 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:

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

Clearly, for any $i \in \mathcal{I}$, $j \in \mathcal{J}$, and $k \in \mathcal{K}$, we have $z^{\text{new}}_{ijk1}(d^1) \leq z^*_{ijk1}(d^2)$. Hence, $z^{\text{new}}_{ijk1}(d^1) \in \mathbb{N}_0$ and the first set of constraints in (4) is 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

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

Now, we show that the last set of constraints in (4) holds as well. By construction, we have for any $(j,k) \notin \overline{\mathcal{JK}}$

$$\sum_{i \in \mathcal{I}_0} z^{\text{new}}_{ijk1}(d^1) = \sum_{i \in \mathcal{I}_0} z^*_{ijk1}(d^2) = d^2_{jk1} = d^1_{jk1},$$

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

$$\begin{aligned}\sum_{i\in\mathcal{I}_0} z^{\text{new}}_{ijk1}(d^1) &= \sum_{\ell=\tilde{\ell}_{jk}}^{I} \max\left\{z^*_{i_\ell jk1}(d^2) - a_{j,k,\ell}, 0\right\} \\ &= z^*_{i_{\tilde{\ell}_{jk}} jk1}(d^2) - a_{j,k,\tilde{\ell}_{jk}} + \sum_{\ell=\tilde{\ell}_{jk}+1}^{I} z^*_{i_\ell jk1}(d^2) \\ &= \sum_{\bar{\ell}=0}^{\tilde{\ell}_{jk}} z^*_{i_{\bar{\ell}} jk1}(d^2) - a_{j,k,0} + \sum_{\ell=\tilde{\ell}_{jk}+1}^{I} z^*_{i_\ell jk1}(d^2) \\ &= \sum_{\ell=0}^{I} z^*_{i_\ell jk1}(d^2) - a_{j,k,0} = d^2_{jk1} - a_{j,k,0} = d^1_{jk1},\end{aligned} \tag{10}$$

where the equality in (10) is because of the construction of $a_{j,k,\ell}$, $\ell = 1, \cdots, I$. Now, since for any $i \in \mathcal{I}$, $j \in \mathcal{J}$, and $k \in \mathcal{K}$, we have $z^{\text{new}}_{ijk1}(d^1) \le z^*_{ijk1}(d^2)$, and $\sum_{i,j,k} c_{ijk} z^*_{ijk1}(d^1) > \sum_{i,j,k} c_{ijk} z^*_{ijk1}(d^2)$, we have

$$\sum_{i,j,k} c_{ijk} z^*_{ijk1}(d^1) > \sum_{i,j,k} c_{ijk} z^{\text{new}}_{ijk1}(d^1).$$

So, $z^{\text{new}}_{ijk1}(d^1)$, $i \in \mathcal{I}_0$, $j \in \mathcal{J}$, $k \in \mathcal{K}$ is a feasible shipment policy that is cheaper than the optimal policy $z^*_{ijk1}(d^1)$, $i \in \mathcal{I}_0$, $j \in \mathcal{J}$, $k \in \mathcal{K}$, which leads to a contradiction. ◀

We emphasize that this result also holds for problems with continuous demand where the shipment policies are continuous (with a similar proof). This theorem asserts that $Q_T(y, S, d_{[T]}, z_{[T-1]})$ is a non-decreasing function in d when $T = 1$. In the next corollary, we extend the results to $T \ge 1$.

Corollary 1. *Let us fix $y_i \in \{0,1\}$ and $S_{ik} \in \mathbb{N}_0$, for all $i \in \mathcal{I}$, $k \in \mathcal{K}$. Let $t = 1, \cdots, T$, be fixed and for any $i \in \mathcal{I}$, $j \in \mathcal{J}$, and $k \in \mathcal{K}$, $c_{ijk} \ge h_{ik}$. Also, let the shipment quantities $z_{[t-1]}$ be given. For a given uncertainty set $\prod_{t=1}^{T} \mathcal{D}_{[t]} \subseteq \mathbb{N}_0^{JKT}$, if $d^1_{[t]}, d^2_{[t]} \in \mathcal{D}_{[t]}$ be such that $d^1_t \lneqq d^2_t$ and $d^1_{[t-1]} = d^2_{[t-1]}$, then*

$$Q_t(y, S, d^1_{[t]}, z_{[t-1]}(d^1_{[t]})) \le Q_t(y, S, d^2_{[t]}, z_{[t-1]}(d^2_{[t]})).$$

Proof. One can prove the corollary using similar lines of reasoning as in the proof of Theorem 2 and the fact that

$$\begin{aligned} Q_t\left(y, S, d_{[t]}, z_{[t-1]}\right) = \min_{z_t} & \sum_{\substack{i\in\mathcal{I}_0\\ j\in\mathcal{J}\\ k\in\mathcal{K}}} (c_{ijk} - h_{ik}) z_{ijkt} + \sum_{\substack{i\in\mathcal{I}\\ k\in\mathcal{K}}} h_{ik}\left(S_{ik} - \sum_{j\in\mathcal{J}}\sum_{\tilde{t}=1}^{t-1} z_{ijk\tilde{t}}\right) \\ & + \max_{d_{t+1}\in\mathcal{D}_{[t+1]}} Q_{t+1}\left(y, S, d_{[t+1]}, z_{[t]}\right) \\ \text{s.t.} \quad & \sum_{j\in\mathcal{J}} z_{ijkt} \le S_{ik} - \sum_{j\in\mathcal{J}}\sum_{\tilde{t}=1}^{t-1} z_{ijk\tilde{t}}, \quad i \in \mathcal{I}, k \in \mathcal{K}, \\ & \sum_{i\in\mathcal{I}_0} z_{ijkt} = d_{jkt}, \quad j \in \mathcal{J}, k \in \mathcal{K}, \\ & z_{ijkt} \in \mathbb{N}_0, \quad i \in \mathcal{I}_0, j \in \mathcal{J}, k \in \mathcal{K}, \end{aligned} \tag{11}$$

where $h_{0k} = 0$, for $k \in \mathcal{K}$. ◀

5.2 Practicality of Convexity and Non-Decreasingness of $Q_t(y, S, d_{[t]}, z_{[t-1]})$

In this section, we show how the results in Section 5.1 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 5.1 imply tractability of LT problems (6) with two typically used uncertainty sets. More specifically, we use the results in Section 5.1 to construct another uncertainty set, known as a dominating uncertainty set [14], for which the LT problem can be efficiently solved.

5.2.1 LT Problems with Hyper-Box Uncertainty Sets

A hyper-box uncertainty set is defined as $\mathcal{D}_t = \mathbb{N}_0^{JK} \cap \prod_{\substack{j\in\mathcal{J}\\k\in\mathcal{K}}} [\underline{d}_{jkt}, \overline{d}_{jkt}]$, where $\overline{d}_{jkt} \in \mathbb{N}$, and have been studied extensively in the literature of robust optimization problems in general [29, 41, 74, 75, 77] and LT problems in specific [4, 68]. In the next corollary, we show how our results imply that (6) with a hyper-box uncertainty set is tractable.

Corollary 2. *Let us assume that $\mathcal{D}_t = \mathbb{N}_0^{JK} \cap \prod_{\substack{j\in\mathcal{J}\\k\in\mathcal{K}}} [\underline{d}_{jkt}, \overline{d}_{jkt}]$, where $\overline{d}_{jkt} \in \mathbb{N}$, and $c_{ijk} \geq h_{ik}$, $j \in \mathcal{J}$, $k \in \mathcal{K}$, $t \in \mathcal{T}$. Then, problem (6) is equivalent to the following deterministic problem*

$$\begin{aligned}
\underline{Q}(\overline{d}) = \sum_{\substack{j\in\mathcal{J}\\k\in\mathcal{K}\\t\in\mathcal{T}}} c_{0jk}\overline{d}_{jkt} + \min_{y,S,z}\ & \sum_{i\in\mathcal{I}} f_i y_i + T\sum_{\substack{i\in\mathcal{I}\\k\in\mathcal{K}}} h_{ik}S_{ik} + \sum_{\substack{i\in\mathcal{I}\\j\in\mathcal{J}\\k\in\mathcal{K}\\t\in\mathcal{T}}} (c_{ijk} - (T-t)\,h_{ik} - c_{0jk})\, z_{ijkt} \\
\text{s.t.}\ & S_{ik} \leq My_i, && i\in\mathcal{I}, k\in\mathcal{K},\\
& \sum_{\substack{j\in\mathcal{J}\\t\in\mathcal{T}}} z_{ijkt} \leq S_{ik}, && i\in\mathcal{I}, k\in\mathcal{K},\\
& \sum_{i\in\mathcal{I}} z_{ijkt} \leq \overline{d}_{jkt}, && j\in\mathcal{J}, k\in\mathcal{K}, t\in\mathcal{T},\\
& S_{ik}, z_{ijkt} \in \mathbb{N}_0, && i\in\mathcal{I}, j\in\mathcal{J}, k\in\mathcal{K}, t\in\mathcal{T},\\
& y_i \in \{0,1\}, && i\in\mathcal{I},
\end{aligned} \tag{12}$$

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 (6) if and only if it is optimal for $\underline{Q}(\overline{d})$.

Proof. Corollary 2 can be proved in two different ways using different properties of the problem.
Proof 1. Using Corollary 1 and backward induction starting from $t = T$, one can see that

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

since $c_{ijk} \geq h_{ik}$, for any $i \in \mathcal{I}$, $j \in \mathcal{J}$, and $k \in \mathcal{K}$.
Proof 2. We prove the corollary for $T = 1$, and the extension can be done by backward induction, 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

$$\begin{aligned}
Q(y,S,d) = \min_z\ & \sum_{\substack{i\in\mathcal{I}_0\\j\in\mathcal{J}\\k\in\mathcal{K}}} c_{ijk}z_{ijk1} \\
\text{s.t.}\ & \sum_{j\in\mathcal{J}} z_{ijk1} \leq S_{ik}, && i\in\mathcal{I}, k\in\mathcal{K},\\
& \sum_{i\in\mathcal{I}_0} z_{ijk1} \geq d_{jk1}, && j\in\mathcal{J}, k\in\mathcal{K},\\
& z_{ijk1}, z_{j0k1} \geq 0, && i\in\mathcal{I}, j\in\mathcal{J}, k\in\mathcal{K},
\end{aligned} \tag{13}$$

where the uncertainty set is $\text{conv}(\mathcal{D})$. 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. (13) has a constraint-wise uncertainty, the results in [59] imply that (6) is equivalent to its static robust optimization formulation, i.e.,

$$\begin{aligned}
\min_{y,S,z}\ & \sum_{i\in\mathcal{I}} f_i y_i + \sum_{\substack{i\in\mathcal{I}\\k\in\mathcal{K}}} h_{ik}S_{ik} + \sum_{\substack{i\in\mathcal{I}_0\\j\in\mathcal{J}\\k\in\mathcal{K}}} c_{ijk}z_{ijk1} \\
\text{s.t.}\ & \sum_{j\in\mathcal{J}} z_{ijk1} \leq S_{ik}, && i\in\mathcal{I}, k\in\mathcal{K},\\
& \sum_{i\in\mathcal{I}_0} z_{ijk1} \geq d_{jk1}, && \forall d\in\mathcal{D}, j\in\mathcal{J}, k\in\mathcal{K},\\
& S_{ik} \leq My_i, && i\in\mathcal{I}, k\in\mathcal{K},\\
& y_i \in \{0,1\}, && i\in\mathcal{I},\\
& z_{ijk1}, z_{j0k1} \geq 0, && i\in\mathcal{I}, j\in\mathcal{J}, k\in\mathcal{K},\\
& S_{ik} \in \mathbb{N}_0, && i\in\mathcal{I}, k\in\mathcal{K},
\end{aligned} \tag{14}$$

which is equivalent to $\underline{Q}(\overline{d})$. ◀

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 problems using existing solvers.

5.2.2 LT Problems with Budget Uncertainty Sets

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

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

and where $\Gamma_{kt}, \underline{d}_{jkt}, \overline{d}_{jkt} \in \mathbb{R}$, for all $j \in \mathcal{J}$, $k \in \mathcal{K}$, and $t \in \mathcal{T}$. Budget uncertainty sets are introduced in [26] to decrease the conservativeness of the robust solutions obtained by using hyper-box uncertainty set (since it cuts the corner of a hyper-box), and extensively used in literature [5, 13, 43, 42].

Using Theorems 1 and 2, we see in the following corollary that we can shrink the budget uncertainty set.

Corollary 3. *Let us assume that $c_{ijk} \geq h_{ik}$, $j \in \mathcal{J}$, $k \in \mathcal{K}$, $t \in \mathcal{T}$, and $\mathcal{D}_t$, $t \in \mathcal{T}$, is a budget uncertainty set. Then,*

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

where $\bar{\mathcal{D}}_t = \prod_{k\in\mathcal{K}} \bar{\mathcal{D}}_{kt}$, $t \in \mathcal{T}$, and

$$\bar{\mathcal{D}}_{kt} = \left\{ d_{jkt} \in \mathbb{N}_0 : \ \lceil \underline{d}_{jkt} \rceil \leq d_{jkt} \leq \lfloor \bar{d}_{jkt} \rfloor, \ \forall j \in \mathcal{J}, \ \sum_{j\in\mathcal{J}} d_{jkt} = \lfloor \Gamma_{kt} \rfloor \right\}. \tag{15}$$

Proof. We show that the convex hull of $\mathcal{D}_{kt}$ is the same as the one of $\widetilde{\mathcal{D}}_{kt}$, for any $k \in \mathcal{K}$ and $t \in \mathcal{T}$, where

$$\widetilde{\mathcal{D}}_{kt} = \left\{ d_{jkt} \in \mathbb{N}_0 : \ \lceil \underline{d}_{jkt} \rceil \leq d_{jkt} \leq \lfloor \bar{d}_{jkt} \rfloor, \ \forall j \in \mathcal{J}, \ \sum_{j\in\mathcal{J}} d_{jkt} \leq \lfloor \Gamma_{kt} \rfloor \right\}.$$

Then, the result follows analogously to Proof 1 of Corollary 2.

Clearly, $\text{conv}(\widetilde{\mathcal{D}}_{kt}) \subseteq \text{conv}(\mathcal{D}_{kt})$. Now, let $d \in \text{conv}(\mathcal{D}_{kt})$. Hence, there exist $d^1, \cdots, d^L \in \mathcal{D}_{kt}$ and $\lambda_1, \cdots, \lambda_L \geq 0$ such that

$$d = \sum_{\ell=1}^{L} \lambda_\ell d^\ell, \quad \sum_{\ell=1}^{L} \lambda_\ell = 1.$$

For any $\ell = 1, \cdots, L$, since $d^\ell \in \mathcal{D}_{kt}$, all its components are integer and hence

$$\lceil \underline{d}_{jkt} \rceil \leq d^\ell_{jkt} \leq \lfloor \bar{d}_{jkt} \rfloor, \ \forall j \in \mathcal{J}, \ \sum_{j\in\mathcal{J}} d^\ell_{jkt} \leq \lfloor \Gamma_{kt} \rfloor.$$

Hence, $\sum_{\ell=1}^{L} \lambda_\ell = 1$ implies that $d \in \text{conv}(\widetilde{\mathcal{D}}_{kt})$. ◀

Corollary 3 asserts that the optimal robust decisions on the location of the local warehouses as well as their base stock 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 $\prod_{t\in\mathcal{T}} \mathcal{D}_t$. So, based on Proposition 1, we can use the convex hull of $\prod_{t\in\mathcal{T}} \bar{\mathcal{D}}_t$, which read as $\prod_{\substack{t\in\mathcal{T}\\k\in\mathcal{K}}} \hat{\mathcal{D}}_t$, where

$$\hat{\mathcal{D}}_{kt} = \left\{ d_{jkt} \in \mathbb{R} : \ \lceil \underline{d}_{jkt} \rceil \leq d_{jkt} \leq \lfloor \bar{d}_{jkt} \rfloor, \ d_{jkt} \geq 0, \quad \forall j \in \mathcal{J}, \ \sum_{j\in\mathcal{J}} d_{jkt} = \lfloor \Gamma_{kt} \rfloor \right\}.$$

In general, it is proved in [61] that the number of vertices of $\hat{\mathcal{D}}_{kt}$ is at most in the order of $\mathcal{O}(J^J)$. For problems where $\prod_{t\in\mathcal{T}} \hat{\mathcal{D}}_t$ has a polynomial number of vertices, the optimal "here-and-now" decisions can be obtained by solving a deterministic problem with a polynomial number of constraints. As an example, for any $k \in \mathcal{K}$ and $t \in \mathcal{T}$, if

$$\sum_{j\in\mathcal{J}} \lfloor \bar{d}_{jkt} \rfloor - \min_{j\in\mathcal{J}} \lfloor \bar{d}_{jkt} \rfloor < \lfloor \Gamma_{kt} \rfloor < \sum_{j\in\mathcal{J}} \bar{d}_{jkt},$$

then $\mathcal{D}_{kt}$ has $2^J + J - 1$ vertices while $\hat{\mathcal{D}}_{kt}$ has only J vertices.

Furthermore, despite intractability of (6) for a general budget uncertainty set [5], one can use Corollary 3 and the efficient algorithm proposed in [57] to solve (6) to optimality.

Hitherto, we have shown how to solve (6) containing hyper-box or 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 (6) 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 in [59]). For the explicit formulation of the problem after the elimination, we refer the reader to Appendix B. We emphasize that all the theoretical results in this paper hold for the new formulation.

6 Approximation 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 [20, 63, 65, 70, 76]. 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 (6) after the elimination, finding the explicit formulation of the convex hull of the subsets may be inefficient. Therefore, such methods are not applicable to LT problems with integer demand, as partitioning the convex hull may result in sets whose vertices are not necessarily integer-valued.

The second class contains methods that restrict integer wait-and-see variables to be (piece-wise) affine functions; see e.g., [18, 56]. In the remainder of 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 have the convexity characteristic.

6.1 Integer-Valued 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 [15] and piece-wise affine decision rules [30, 71]. The idea behind Affine Decision Rules (ADR) is to restrict the wait-and-see variables to be affine in the uncertain parameters. This type of approximation is known to perform well on the continuous relaxation of (6), where the integrality restriction over the demand and the shipment policies are relaxed (see, e.g., [4]). However, such approximations are not applicable to multi-stage robust problems with integer wait-and-see variables. 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

$$z_{ijkt}(d_{[t]}) := u_{ijkt} + \left\langle V^{ijkt}, d_{[t]} \right\rangle,$$

where $u_{ijkt} \in \mathbb{R}$ and $V^{ijkt} \in \mathbb{R}^{JKt}$. Clearly, there is no guarantee that the value of $u_{ijkt} + \left\langle V^{ijkt}, d_{[t]} \right\rangle$ is integer even with an integer-valued demand vector $d_{[t]}$.

To resolve this issue, in [18] the author 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 $u_{ijkt} + \left\langle V^{ijkt}, d_{[t]} \right\rangle$, where $u_{ijkt} \in \mathbb{N}_0$ and $V^{ijkt} \in \mathbb{N}_0^{JKt}$. 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-Valued Affine Decision Rule (IADR).

To be able to use the conventional reformulation after applying the IADR, the uncertainty set needs to be convex. But considering the LT problem, the uncertainty set after using IADR can be replaced by its convex hull, as the worst-case scenarios are among the vertices. Therefore, one can approximate (6) in three steps: first, the equality constraints need to be eliminated, then the IADR 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., [46] for how we can solve the approximation).

Now that we know how we can use an IADR, the question we are interested in is whether we cover all affine policies. In the following proposition, we show that this is the only way of restricting an integer wait-and-see variable to be affine in an integer uncertain parameter.

Proposition 2. *For given $n, m \in \mathbb{N}$, a 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, $x = \sum_{i=1}^n x_i e_i$, 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

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

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

Based on the proof of Proposition 2, any affine policy to restrict an integer wait-and-see variable with a full-dimensional uncertainty set is an IADR. However, the full-dimensionality of the uncertainty set is a crucial assumption. To see that, let us consider the uncertainty set where $J = 3$, $K = 1$, $T = 1$, and

$$\mathcal{D} = \left\{ d \in \mathbb{Z}^3 : \sum_{i=1}^3 d_i = 3,\ 0 \le d_i \le 6 \right\}.$$

Then, for any $d \in \mathcal{D}$, we know $z(d) = Vd + 1 \in \mathbb{Z}$, where $V = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$, because for any $d \in \mathcal{D}$

$$Vd + 1 = \sum_{i=1}^3 \frac{d_i}{3} + 1 = \frac{1}{3}\sum_{i=1}^3 d_i + 1 = 2.$$

So, IADRs do not necessarily contain all affine policies if the uncertainty set is not full-dimensional. This shows another importance of Proposition 1. Based on this proposition, we know that we can use the continuous relaxation of the problem and the customary affine decision rule to be sure that all affine policies are considered while approximating (6).

One of the main drawbacks of applying (integer-valued) ADR is the number of new variables that need to be added in order to reformulate to a deterministic optimization problem. This number can be extremely large, making the (integer-valued) ADR impractical. In the next section, we provide a new iterative technique to obtain a locally robust solution for practical problems.

6.2 A Simplex-Type Method to Obtain a Lower Bound

In this paper, we assume that the uncertainty set is bounded. Hence, it has a finite number of vertices; hence its convex hull is a polytope (see, e.g., [38, Theorem 6]). So, we let the convex hull to be in the form

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

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, e.g., [12, Chapter 3]). As we have seen in the proof of Corollary 3, the worst-case scenario occurs in a vertex of $\mathcal{D}$. So, given a vertex, check the objective values of the deterministic LT problems with the demand vectors 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 select the starting vertex, we randomly generate a vector $c \in [-1, 1]^{JKT}$ and solve the linear optimization problem $\min\{c^\top d : d \in \mathcal{D}\}$. Let us denote by d^0 the optimal vertex of the linear optimization problem. Then, similar to the Simplex method, we denote by B and N the basic and non-basic submatrices of A corresponding to d^0. Then, increasing any non-basic variable d_k to

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

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

We emphasize that the Simplex-type method is applicable to any linear adjustable robust optimization problem with the right-hand-side uncertainty since, for such problems, the worst-case scenario is among the vertices of the uncertainty set [4].

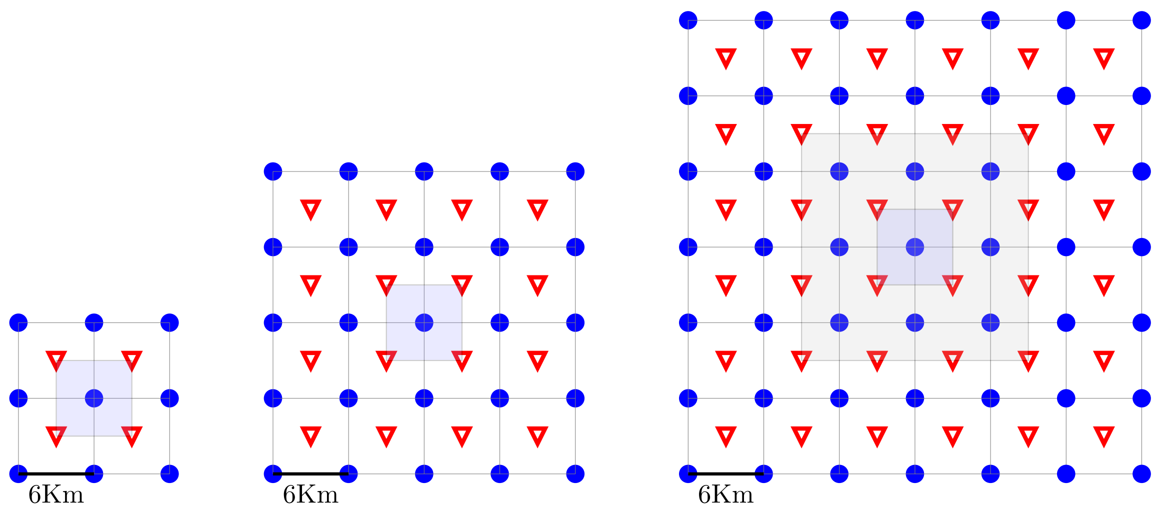
7 Numerical Experiment

In this section, we show the effectiveness of the solution methods, proposed in previous sections, in solving robust LT problems. We first discuss the testbed used in this paper to construct random instances (Section 7.1). Then, we show how the methods perform on instances with budget uncertainty sets (Section 7.2).

The numerical results of this work were carried out on the Dutch national e-infrastructure with the support of SURF Cooperative. We used a virtual machine with 8 processors, 2.30 GHz, and 20.00 GB RAM running Julia 1.11.3 [27]. We use JuMP [40] to pass Mixed Integer Linear Optimization (MILO) problems to Gurobi 12.0.1 [47]. To ensure a fair comparison, the time reported in this section includes the time needed to construct the model, pass it to the solver, and the time taken by the solver to solve the optimization problem. We use the Python package RSOME [31] to approximate the problem using affine decision rules. For all methods, a time limit of 3,600 seconds is applied per problem instance.

7.1 Testbed

To compare different solution methods, we randomly generate three classes of LT instances to optimally design the distribution network of an e-commerce company. The decisions are in finding the optimal locations of warehouses and their inventory of each product while considering the optimal shipment policies to the customers. We distinguish three classes of instances: (I) Class One instances generated for a 12Km × 12Km square city with $J = 9$ customers and one product valued €50. (II) Class Two instances generated for a 24Km × 24Km square city with $J = 25$ customers and two products valued €50 and €200. (III) Class Three generated for a 36Km × 36Km square city with $J = 49$ customers and two products valued €100 and €200. The candidate locations to open warehouses are the middle points of the 6Km × 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 Class One instances the coordinate of the customer in the south-west is $\begin{bmatrix} 0 \\ 0 \end{bmatrix}$, and the candidate warehouse location close to it has the coordinate $\begin{bmatrix} 3 \\ 3 \end{bmatrix}$.



(a) Class One **(b)** Class Two **(c)** Class Three

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 × 6Km square shows the center of the city.

The company offers a same-day delivery service for the time horizon $L = 365$ days. 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 ($t_{jk}^{\max} = 1h$, for all $j \in \mathcal{J}$

and $k \in \mathcal{K}$). In case of tardiness, a unit penalty of 20% of the product value per hour delay is paid to the customer. Thus, $c^p_{jk} = 0.2v_k$, 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 unit ($c_{0jk} =$ €50, for customer j per unit of product k). This cost covers both the transportation cost and the penalty cost for being too late.

One way of reducing the shipment cost is by opening warehouses in the city. In Class One, the cost of opening a warehouse (excluding replenishment costs) is €69,350. In Class Two, the cost of opening a warehouse is €69,350 for a location outside the city center and €142,350 for a location in the city center (highlighted by a light blue square in Figure 3b). In Class Three, the cost of opening a warehouse is €69,350 for a location outside the city center, €142,350 for a location in the middle of the city (located in boundary of a light gray square in Figure 3c), and €215,350 for a location in the city center (highlighted by a light blue square in Figure 3c). Moreover, the inventory holding cost per unit 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 units from the warehouses in the city to the customers, the company plans to use couriers at the cost of $9\frac{€}{\text{hour}}$ who travel at an average speed of $18\frac{\text{Km}}{\text{hour}}$. 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 $\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$, the Manhattan distance is used, which is defined as

$$\text{Dist}\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}\right) := |x_1 - y_1| + |x_2 - y_2|. \tag{16}$$

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 and a unit transit cost of €0.5. Therefore, the scaled total fixed cost per warehouse is determined by its location tier and applies consistently across all instance classes: for warehouses outside the city center (Classes One, Two, and Three),

$$f_i = \frac{69{,}350 + 3{,}650}{365} = 200 \quad \left(\frac{€}{\text{day}}\right),$$

for warehouses midside the center (Classes Two and Three),

$$f_i = \frac{142{,}350 + 3{,}650}{365} = 400 \quad \left(\frac{€}{\text{day}}\right),$$

and for warehouses inside the center of the large city (Class Three only),

$$f_i = \frac{215{,}350 + 3{,}650}{365} = 600 \quad \left(\frac{€}{\text{day}}\right).$$

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

$$c_{ijk} = \underbrace{\frac{2 \times 9 \times \text{Dist}_{ij}}{18}}_{\text{courier cost}} + \underbrace{0.2v_k\left(\frac{\text{Dist}_{ij}}{18} - 1\right)^+}_{\text{tardiness penalty cost}} + \underbrace{0.5}_{\text{replenishment cost}} \quad \left(\frac{€}{\text{unit}}\right), \tag{17}$$

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 3.

The company wants to use the demand data on the last 100 days to make the decisions. To construct the demand data, we first fix $\hat{d} \in \{1, 2, 3, 4\}$, where $\hat{d}$ denotes the maximum demand rate. Then, we randomly generate the demand rate $\tilde{d} \in [0, \hat{d}]^{JKT}$ from a uniform distribution. For each $j \in \mathcal{J}$, $k \in \mathcal{K}$, and $t \in \mathcal{T}$, we draw 100 scalar data points data_{Pjkt} (so, data_P is a vector data point) from the Poisson distribution with rate $\tilde{d}_{jkt}$. We also draw another 100 data points $\text{data}_{\beta Bin}$ 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 $[\frac{\text{data}_P + \text{data}_{\beta Bin}}{2}]$, where $[\cdot]$ is the rounding function. We have chosen this procedure to avoid having a demand vector with a specific distribution.

Table 3 Parameters Used in the Generated Instances.

Parameter (Units)	Class One	Class Two	Class Three
I	4	16	36
J	9	25	49
K	1	2	2
T	6	6	6
f_i ($\frac{€}{\text{day}}$)	200	200, 400	200, 400, 600
v_k ($\frac{€}{\text{item}}$)	50	50, 200	100, 200
h_{ik} ($\frac{€}{\text{item}}$)	$9 \times 10^{-5} v_k$	$9 \times 10^{-5} v_k$	$9 \times 10^{-5} v_k$
Dist_{ij} (Km)	Defined in (16)	Defined in (16)	Defined in (16)
c_{ijk} ($\frac{€}{\text{item}}$)	Defined in (17)	Defined in (17)	Defined in (17)
c_{0jk} ($\frac{€}{\text{item}}$)	50	50	50

7.2 LT Instances with Budget Uncertainty Set

Based on the dataset, we construct the uncertainty set using the empirical distribution of the demands of the customers. More specifically, we use the set

$$\text{conv}(\mathcal{D}_t) := \left\{ d_{kt} \in \mathbb{R}^J : \sum_{\substack{j \in \mathcal{J} \\ k \in \mathcal{K}}} d_{jkt} \leq \bar{d}_t^{\alpha},\ 0 \leq d_{jkt} \leq \bar{d}_{jkt}^{\alpha},\ \forall j \in \mathcal{J}, k \in \mathcal{K} \right\}, \tag{18}$$

as the convex hull of the budget uncertainty set, where $\bar{d}_{jkt}^{\alpha}, \bar{d}_t^{\alpha} \in \mathbb{N}$ 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 $j \in \mathcal{J}$, $k \in \mathcal{K}$, $t \in \mathcal{T}$. To see the impact of α in the final solution, we vary its value in $\{0.60, 0.65, 0.70, \cdots, 0.95\}$.

As stated in Corollary 3, one can solve the LT instances by enumerating the vertices of the uncertainty set. Table 4 shows the number of vertices of the set constructed in (18) for the Class One instances.

Table 4 Number of Vertices of $\prod_{t \in \mathcal{T}} \text{conv}(\mathcal{D}_t)$ for Class One Instances with Different Values of α and $\hat{d}$.

	$\hat{d} = 1$		$\hat{d} = 2$		$\hat{d} = 3$		$\hat{d} = 4$	
α	0.60	0.90	0.60	0.90	0.60	0.90	0.60	0.90
Number of Vertices	1×10^9	2×10^{14}	7×10^9	2×10^{14}	9×10^{10}	3×10^{14}	4×10^9	1×10^{14}

As one can see in Table 4, the enumeration of the vertices of the budget uncertainty set is not computationally efficient to get the exact optimal solution. Therefore, we compare the performance of the following methods:

- The lower bound derived from the backward-forward procedure proposed in [64], which breaks down the multi-stage problem into multiple simpler two-stage problems. This approach begins with a finite set of pre-defined feasible decisions for each stage. During the backward phase, it identifies scenarios, while in the forward phase, it refines these decisions by incorporating new ones. In the implementation, we initialize the method using the solution from the deterministic LT problem as the initial feasible decisions, execute the backward-forward procedure, and report the lower bound (even though the method also provides an upper bound, it turns out that its upper bound is very loose in our setting). We refer to this method as BFP.
- We apply the finite scenario approach, where we randomly choose a vertex scenario from an uncertainty set, and then find scenarios that would generate the highest emergency shipment cost. We then restrict the uncertainty set to these scenarios to find a lower bound. We refer to the finite scenario approach as FS in what follows.

- We apply the Simplex-type method, defined in Section 6, where the initial scenario is generated randomly. Then, after each iteration, we run the FS approach given those obtained scenarios to construct a lower bound, and terminate the algorithm if there is no improvement in the obtained lower bound compared to the previous iteration. To increase the chance of obtaining the global optimum, we run the method for 3 randomly selected initial vertices. Hereafter, we refer to this method by SM.
- We apply the SM approach starting from the scenario in the uncertainty set that has the highest emergency shipment cost. We refer to this method as SMFS, as it uses the same scenarios as in FS to start the procedure.
- We apply a two-phase procedure, where in the first phase, we apply the FS approach. Then, for the obtained scenarios, we find the one that has the worst objective function and use it as the initial scenario for the SM algorithm. We refer to this FS-then-SM procedure as FSSM.

We emphasize that the main difference between SMFS and FSSM is that in SMFS, we do not run FS and just "guess" the worst-case scenario, while in FSSM, we choose the worst-case scenario of the FS approach as the initial scenario for SM.

Since all these methods are generating a lower bound, to construct an upper bound, we use the IADR. Applying IADR to the LT problem for these instances yields results that are nearly identical to those obtained by applying ADR to the continuous relaxations of the LT problem. Therefore, we only present the results obtained by applying ADR to the continuous relaxations of the LT problem.

Moreover, as Corollary 3 asserted, the uncertainty set can be reduced to the scenarios in the cut (15). In the numerical experiments, we investigate 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.

7.2.1 Results for Class One Instances

For this class, we generate five random instances for various values of $\hat{d}$ and α, with details provided in Appendix C.1. To assess the methods, we analyze the solution gap between the upper bound obtained using the ADR and the derived lower bounds. Specifically, we evaluate the performance of these lower bounds by computing the optimality gap, defined as:

$$\text{OGap}(\%) = \left(\frac{\text{UB} - \text{LB}}{|\text{LB}| + 10^{-4}} \right) \times 100,$$

where UB represents the upper bound using ADR, and LB denotes the lower bound for a given method. The addition of 10^{-4} to the denominator ensures that a division by zero does not occur.

In some cases, we observe a negative optimality gap of approximately -0.2%, which is caused by the numerical errors and indicates that the optimal solution has been found. Therefore, we replace negative gaps with zero.

Table 5 presents time and optimality gap comparisons among various methods for each group of small-sized instances for only $\hat{d} = 2$ for the brevity of exposition (see Appendix C.1 for more details). For each group, the table lists the mean optimality gap with standard deviations shown between brackets. Boldface numbers highlight the best optimality gap in each group.

Let us start with the upper bounds. In comparing ADR and ADR_{cut}, the results show that both methods yield nearly identical upper bounds across all values of $\hat{d}$ and α. However, ADR consistently achieves these results in significantly less time – often under 0.06 seconds – while ADR_{cut} incurs notably higher runtimes (ranging from 0.11 to over 1 second) without any meaningful improvement in solution quality. This indicates that ADR_{cut} does not enhance the bound-tightening process but instead introduces computational overhead. Therefore, for Class One instances, ADR is the more efficient and effective choice, particularly as $\hat{d}$ increases and the structure of the uncertainty set becomes more favorable for fast, accurate bound computation.

Regarding the time of obtaining the lower bounds, clearly, FS is the fastest approach, and BFP is the slowest. BFP consistently reaches the time limit of 3,600 seconds, indicating its computational intensity, and yields the largest optimality gaps, making it less practical for Class One instances. The BFP's primary computational challenge stems from the backward steps. Due to the structure of the LT problem at each time-slot, the algorithm must approximate a two-stage subproblem by exhaustively evaluating all combinations of pre-determined candidate solutions from prior time-slots. This leads to a combinatorial explosion of subproblems, resulting in prohibitive computational costs, slow convergence to a good solution, and frequent hitting of time limits. Furthermore, the Simplex-type method requires more time than the FS method, as it solves numerous deterministic LP problems (which, of course, can be parallelized).

Table 5 Mean Optimality Gaps (%) and Computation Time (s) for Class One with $\hat{d} = 2$.

	SM		SM_{cut}		FA	
α	OGap	Time	OGap	Time	OGap	Time
0.60	0.79 [0.85]	34.66 [14.94]	0.55 [0.52]	5.16 [3.40]	2.51 [1.07]	0.09 [0.01]
0.65	**1.17** [0.85]	24.04 [3.92]	1.43 [1.43]	4.71 [1.81]	3.49 [1.66]	0.08 [0.01]
0.70	**1.81** [1.65]	25.01 [5.19]	3.15 [1.29]	2.94 [0.64]	3.36 [1.29]	0.08 [0.01]
0.75	1.84 [0.64]	26.34 [10.55]	2.45 [1.86]	4.79 [1.84]	3.05 [0.83]	0.07 [0.01]
0.80	2.28 [1.19]	25.72 [6.77]	**2.26** [1.28]	8.16 [4.09]	4.82 [1.31]	0.07 [0.01]
0.85	**0.99** [0.59]	34.32 [9.62]	3.60 [0.92]	6.84 [3.30]	4.35 [1.72]	0.08 [0.01]
0.90	2.28 [1.38]	22.07 [6.55]	4.19 [1.65]	5.81 [3.30]	6.08 [1.88]	0.07 [0.00]
0.95	**0.83** [0.88]	23.05 [7.80]	2.50 [1.80]	6.26 [2.68]	5.21 [1.22]	0.07 [0.01]
	FA_{cut}		FASM		$FASM_{cut}$	
α	OGap	Time	OGap	Time	OGap	Time
0.60	2.11 [1.25]	0.07 [0.01]	0.61 [0.81]	4.01 [2.82]	1.10 [1.32]	5.21 [2.36]
0.65	3.24 [0.64]	0.08 [0.00]	1.73 [0.98]	3.73 [2.63]	2.29 [0.97]	3.13 [1.73]
0.70	4.50 [1.52]	0.08 [0.00]	2.12 [1.37]	2.64 [1.67]	2.37 [1.35]	3.77 [3.95]
0.75	3.93 [1.53]	0.07 [0.01]	2.34 [0.97]	4.59 [2.78]	2.50 [1.22]	3.21 [1.19]
0.80	4.89 [1.71]	0.08 [0.01]	3.27 [1.30]	5.57 [3.98]	2.30 [1.39]	4.57 [1.89]
0.85	5.38 [0.99]	0.13 [0.11]	3.71 [0.59]	3.92 [3.66]	2.74 [1.31]	6.07 [5.95]
0.90	5.18 [1.69]	0.07 [0.01]	2.33 [1.83]	7.43 [3.84]	4.05 [1.45]	3.40 [2.36]
0.95	5.21 [2.00]	0.07 [0.01]	3.18 [1.79]	4.57 [5.25]	2.83 [1.20]	3.73 [1.80]
	SMFA		$SMFA_{cut}$		BFP	
α	OGap	Time	OGap	Time	OGap	Time
0.60	**0.41** [0.76]	7.17 [2.89]	1.26 [1.04]	4.54 [2.96]	5.67 [0.46]	3,600*
0.65	1.56 [0.53]	4.76 [1.67]	1.96 [0.57]	5.56 [1.55]	5.44 [1.05]	3,600*
0.70	2.29 [0.91]	4.11 [2.21]	3.08 [0.85]	4.05 [2.38]	5.52 [1.03]	3,600*
0.75	2.98 [0.71]	3.49 [1.92]	**1.48** [0.84]	6.82 [2.66]	5.86 [1.47]	3,600*
0.80	3.03 [1.28]	6.64 [2.42]	3.05 [1.53]	6.24 [2.43]	6.78 [0.73]	3,600*
0.85	3.10 [1.49]	7.57 [3.22]	2.00 [1.69]	7.44 [4.54]	7.12 [0.96]	3,600*
0.90	2.56 [1.64]	10.29 [5.46]	**1.62** [1.10]	6.28 [4.88]	7.53 [1.30]	3,600*
0.95	3.11 [1.75]	6.68 [5.86]	2.01 [1.58]	4.86 [1.68]	6.98 [1.34]	3,600*

Note. The asterisk '*' indicates that the method reached the time limit.

Regarding the quality of the solution, the Appendix C.1 shows that SM generally outperforms FS and BFP. For $\hat{d} = 1$ and $\hat{d} = 2$, SM achieves mean optimality gaps ranging from 0.59% to 2.61% and 0.79% to 2.28%, respectively, while FS and BFP exhibit larger gaps (up to 6.44% for FS and 7.93% for BFP). For $\hat{d} = 3$ and $\hat{d} = 4$, SM's performance improves significantly, with gaps as low as 0% for $\hat{d} = 4$ across all α values, reflecting the reduced number of vertices in the uncertainty set as $\hat{d}$ increases, which allows SM to better identify worst-case scenarios. The cut variants (e.g., SM_{cut}, FS_{cut}) often reduce computation time (e.g., SM_{cut} takes 1.15–5.08 seconds compared to SM's 17.35–39.88 seconds), but they do not consistently improve the optimality gap, sometimes resulting in larger gaps (e.g., SM_{cut} at $\alpha = 0.95$, $\hat{d} = 1$ has a 4.95% gap vs. SM's 2.61%). This is mainly due to the fact that if the starting scenario is 'close' to the worst-case scenarios, then we can reach them faster, however, if the initial scenario is far from the worst ones, the uncertainty set with cut can get stuck in a local scenario, while the full dimensional one has more flexibility in skipping such scenarios.

The hybrid methods FSSM and SMFS show competitive performance, often achieving gaps close to SM (e.g., SMFS at $\hat{d} = 2$, $\alpha = 0.60$ has a 0.41% gap), but they require slightly more computation time than FS due to their hybrid nature. Therefore, among all the approaches, the results on these instances reveal a clear trade-off: the various SM approaches yield better solutions than FS, but at the expense of longer runtimes.

7.2.2 Results for Class Two Instances

In this class, we generate five random instances for different values of $\hat{d}$ and α. We only present and discuss the results of instances with $\hat{d} = 2$. For the other values of $\hat{d}$, similar results are obtained. For this class of instances, IADR (and ADR for the continuous relaxation) is not applicable as it results in an "*out-of-memory*" error. Moreover, given the poor performance of BFP, we do not use it for the rest of the analysis. Therefore, the Simplex-type method, finite scenario approach, and their variant are the only approaches that can be used to find a solution to the LT instances.

Since no reliable upper bounds are available (due to the intractability of ADR/IADR), we evaluate solution quality by comparing lower bounds across methods, as used in [51], to compare solution methods together. We compute the solution gap as:

$$\text{SGap}(\%) = \left(\frac{\text{LB}^{(\text{best})} - \text{LB}}{|\text{LB}| + 10^{-4}} \right) \times 100,$$

where $\text{LB}^{(\text{best})}$ denotes the highest lower bound obtained among all methods, and LB represents the lower bound for a specific method.

The solution gap (SGap) and computation time are reported as the mean and standard deviation (in brackets) for each instance group across the runs, as shown in Table 6 (see Appendix C.2 for details).

Table 6 Mean Solution Gaps (%) and Computation Time (s) for Class Two with $\hat{d} = 2$.

	SM		SM_cut		FA		FA_cut	
α	SGap	Time	SGap	Time	SGap	Time	SGap	Time
0.60	9.81 [0.97]	3,600*	**0.02** [0.02]	1923.09 [1006.53]	0.66 [0.09]	1.73 [0.42]	0.63 [0.34]	2.24 [0.48]
0.65	6.94 [0.55]	3,600*	0.20 [0.22]	1728.50 [976.47]	0.77 [0.09]	1.51 [0.15]	0.40 [0.08]	1.59 [0.18]
0.70	4.07 [0.94]	3,600*	0.03 [0.05]	2049.95 [750.32]	1.00 [0.04]	1.29 [0.15]	0.71 [0.36]	1.60 [0.16]
0.75	3.42 [1.06]	3,600*	**0.01** [0.01]	2927.43 [910.40]	1.19 [0.14]	1.37 [0.18]	1.28 [0.30]	1.70 [0.27]
0.80	2.51 [0.71]	3,600*	0.10 [0.13]	1621.19 [947.46]	1.08 [0.15]	1.38 [0.25]	**0.07** [0.04]	1.71 [0.21]
0.85	1.28 [0.93]	3,600*	0.21 [0.35]	1612.77 [576.20]	1.20 [0.14]	1.34 [0.18]	0.52 [0.52]	1.92 [0.47]
0.90	0.55 [0.22]	3,600*	0.12 [0.14]	2169.09 [856.72]	1.13 [0.21]	1.41 [0.20]	0.44 [0.64]	1.56 [0.10]
0.95	**0.03** [0.02]	3508.07 [155.35]	0.09 [0.08]	622.05 [193.70]	0.97 [0.16]	1.50 [0.15]	1.04 [0.24]	1.32 [0.17]

	FASM		FASM_cut		SMFA		SMFA_cut	
α	SGap	Time	SGap	Time	SGap	Time	SGap	Time
0.60	0.16 [0.26]	597.72 [231.17]	0.08 [0.11]	1529.29 [1691.22]	0.37 [0.30]	1301.99 [913.26]	0.03 [0.03]	1357.13 [866.56]
0.65	**0.01** [0.02]	2322.37 [1411.28]	0.04 [0.03]	752.88 [462.35]	0.13 [0.22]	2353.63 [891.45]	0.17 [0.20]	952.35 [442.52]
0.70	0.13 [0.13]	998.62 [336.24]	**0.02** [0.02]	1578.19 [1432.44]	0.48 [0.37]	1871.77 [789.13]	0.09 [0.13]	1978.21 [787.35]
0.75	0.41 [0.22]	892.70 [419.61]	0.05 [0.05]	1551.29 [1491.70]	0.61 [0.40]	1204.33 [259.71]	0.19 [0.26]	2227.21 [839.40]
0.80	0.31 [0.18]	616.47 [213.36]	0.26 [0.57]	526.44 [290.17]	0.26 [0.27]	1569.40 [380.44]	0.35 [0.34]	1490.91 [987.13]
0.85	**0.12** [0.16]	1137.04 [1234.85]	0.18 [0.26]	1713.35 [1159.62]	0.44 [0.28]	1848.58 [961.65]	0.13 [0.12]	1025.46 [340.95]
0.90	**0.03** [0.03]	758.61 [323.13]	0.48 [0.43]	714.01 [415.20]	0.54 [0.45]	1448.64 [915.09]	0.06 [0.08]	1367.19 [1123.99]
0.95	0.17 [0.33]	807.05 [358.12]	0.07 [0.10]	599.70 [175.34]	0.26 [0.23]	1296.42 [593.27]	0.05 [0.05]	816.61 [369.11]

As shown in Table 6, in this class of instances, SM_cut generally outperforms SM across most α values, achieving tighter lower bounds with mean solution gaps ranging from 0.01% ($\alpha = 0.75$) to 0.21% ($\alpha = 0.85$), compared to SM's 0.03% ($\alpha = 0.95$) to 9.81% ($\alpha = 0.60$). The exception is at $\alpha = 0.95$, where SM yields a smaller SGap of 0.03% versus SM_cut's 0.09%. By reducing the vertex count in its uncertainty set, SM_cut enhances solution quality and accelerates convergence. Additionally, SM_cut is far more efficient, with computation times ranging from 622.05 seconds ($\alpha = 0.95$) to 2,927.43 seconds ($\alpha = 0.75$), while SM frequently reaches the 3,600-second limit without converging.

Furthermore, the result reveals that the hybrid methods SMFS and FSSM, along with their cut variants SMFS_cut and FSSM_cut, consistently achieve tighter lower bounds than FS in all groups. The same holds true for FS_cut except for a group with $\alpha = 0.80$. Despite this, both FS and FS_cut excel in computational speed, consistently solving instances in under 2.24 seconds for all groups. This rapid convergence stems from FS's strategy of initially selecting a random demand scenario from the uncertainty set, next to the selection of 'heuristic' scenarios.

Considering the cut uncertainty set in FS_{cut} aims to restrict the uncertainty set to a subset of scenarios, potentially focusing the search on more relevant regions. However, this modification does not consistently improve performance. For instance, at $\alpha = 0.65$, FS_{cut} achieves a mean SGap of 0.40%, outperforming FS's 0.77%, but at $\alpha = 0.75$, FS_{cut}'s SGap of 1.28% is higher than FS's 1.19%. As a result, FS_{cut} does not provide a reliable advantage over FS, making both methods less competitive in terms of solution quality compared to SM-based or hybrid approaches, though their speed remains a significant benefit for time-sensitive applications.

On the contrary, considering the cut uncertainty set boosts the quality of the solution obtained by SM. The main reason is that the vertices needed to be enumerated by SM get fewer with better qualities when we restrict the uncertainty set to the cut. This means the time needed to find the solution also improves in SM, considering the cut uncertainty set. Generally speaking, FS can report a solution faster than our other proposed methods, but with lower quality.

7.2.3 Results for Class Three Instances

To evaluate the performance of our solution methods on Class three instances, we consider the following approaches: SM_{cut},FS, FS_{cut}, FSSM, $FSSM_{cut}$, SMFS, and $SMFS_{cut}$, as the others showed poor performance compared to these methods for other classes.

Table 7 Mean Solution Gaps (%) and Computation Time (s) for Class Three with $\hat{d} = 2$.

	SM_{cut}		FA		FA_{cut}	
α	SGap	Time	SGap	Time	SGap	Time
0.60	0.24 [0.15]	3,600*	1.03 [0.20]	11.18 [0.50]	0.35 [0.19]	13.20 [1.75]
0.65	0.32 [0.06]	3,600*	1.10 [0.06]	11.51 [1.60]	0.52 [0.28]	12.10 [2.35]
0.70	0.30 [0.15]	3,600*	1.09 [0.10]	11.17 [2.25]	0.53 [0.31]	11.68 [3.08]
0.75	0.32 [0.10]	3,600*	0.93 [0.08]	9.84 [1.04]	0.76 [0.19]	11.27 [1.66]
0.80	0.22 [0.11]	3,600*	0.80 [0.09]	11.37 [2.28]	0.61 [0.34]	9.77 [2.07]
0.85	0.19 [0.08]	3,600*	0.71 [0.09]	9.46 [1.23]	0.74 [0.09]	9.06 [0.86]
0.90	0.24 [0.15]	3,600*	0.62 [0.19]	10.27 [2.16]	0.58 [0.15]	8.64 [0.97]
0.95	0.22 [0.17]	3,600*	0.54 [0.12]	7.61 [0.63]	0.46 [0.10]	8.04 [1.11]

	FASM		$FASM_{cut}$		SMFA		$SMFA_{cut}$	
α	SGap	Time	SGap	Time	SGap	Time	SGap	Time
0.60	**0.00** [0.00]	2544.75 [1113.02]	0.38 [0.08]	3190.88 [825.34]	0.83 [0.27]	3530.31 [632.31]	0.26 [0.16]	3,600*
0.65	**0.00** [0.00]	3338.75 [671.68]	0.39 [0.21]	3304.45 [1527.47]	1.46 [0.17]	3,600*	0.63 [0.08]	3,600*
0.70	**0.01** [0.01]	3088.88 [1041.08]	0.42 [0.20]	2983.40 [1426.72]	1.39 [0.41]	3,600*	0.60 [0.10]	3,600*
0.75	**0.00** [0.00]	3540.74 [1253.58]	0.56 [0.37]	3537.13 [1422.05]	1.71 [0.21]	3,600*	0.58 [0.14]	3,600*
0.80	**0.00** [0.00]	3,600*	0.72 [0.34]	3102.24 [1435.17]	1.64 [0.15]	3,600*	0.52 [0.24]	3,600*
0.85	**0.00** [0.00]	2713.08 [1217.19]	0.66 [0.12]	3542.75 [820.69]	1.55 [0.18]	3,600*	0.48 [0.07]	3,600*
0.90	**0.00** [0.00]	1649.63 [934.23]	0.67 [0.16]	2809.43 [1720.84]	1.38 [0.23]	3,600*	0.19 [0.13]	3,600*
0.95	**0.03** [0.06]	2261.23 [1230.80]	0.82 [0.20]	2640.14 [1279.64]	1.38 [0.15]	3,600*	0.25 [0.18]	3,600*

Table 7 provides the statistics on the solution gap as well as the computation time of all (see Appendix C.3). As one can see, FSSM achieves the most effective performance in terms of solution quality. It consistently delivers the tightest lower bounds, with solution gaps (SGaps) approaching zero across all α levels. This is attributed to the worst-case scenario selection of FS and SM's efficient vertex enumeration. However, this high accuracy comes with increased computation time, with runtimes ranging from approximately 1,649 to 3,600 seconds, which is the timelimt.

SM_{cut} ranks second in performance, achieving small SGaps between 0.19% and 0.32%. This method consistently reaches the time limit of 3,600 seconds. $FSSM_{cut}$ and $SMFS_{cut}$ show competitive performance, with SGaps ranging from 0.38% to 0.82% and 0.19% to 0.63%, respectively. While both frequently hit the 3,600-second time limit, $FSSM_{cut}$ occasionally completes earlier (in the range of 2,640 to 3,542 seconds).

In contrast, FS and FS_{cut} produce higher SGaps but require significantly less computation time. These methods prioritize speed by solving a deterministic approximation of the problem. Additionally, the cut-based uncertainty set in FS_{cut} does not guarantee consistent improvement over FS; for instance, at $\alpha = 0.60$, FS_{cut} outperformsFS, whereas at $\alpha = 0.85$, it underperforms.

SMFS performs the weakest among all tested methods in this class of instances, exhibiting the highest SGaps, ranging from 0.83% to 1.71%, and consistently reaching the 3,600-second time limit. In contrast, $SMFS_{cut}$ demonstrates acceptable performance and provides a tighter lower bound than FS.

To sum up, FSSM demonstrates negligible SGaps across all α levels. SM_{cut}, $FSSM_{cut}$, and $SMFS_{cut}$ also maintain relatively low gaps, although these increase slightly for lower values of α. On the other hand, FS and FS_{cut} show higher variability in solution gap while having the smallest computation time. SMFS consistently exhibits the largest gaps.

8 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.

We have shown that the optimal shipment cost from locations of facilities to customers, $Q_t(y, S, d_{[t]}, z_{[t-1]})$, is a convex function in demand d_t. Using this characteristic, we have proved that the LT problem can be solved by vertex enumeration techniques. Moreover, we have shown that for a class of LT problems this characteristic implies tractability.

Then, we have shown that under some conditions, $Q_t(y, S, d_{[t]}, z_{[t-1]})$ is a non-decreasing function in demand d_t, which leads to tractability of another 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 methods with the ones obtained by the existing methods in the literature. The experiments have been set up to design distribution networks of a hypothetical e-commerce company in three different cities. The experiments have shown that our methods have provided solutions of a better quality, but slower than the finite scenario approach. Since our method is parallelizable, the combination of the finite scenario approach and the Simplex-type method provides us with a new state-of-the-art method to approximate large instances.

References

1. Adam F. Abdin, Aakil Caunhye, Enrico Zio, and Michel-Alexandre Cardin. Optimizing generation expansion planning with operational uncertainty: A multistage adaptive robust approach. *Applied Energy*, 306:118032, 2022.
2. Amazon. Amazon prime air. `https://www.aboutamazon.com/news/transportation/amazon-drone-deliver-package`, 2020.
3. Amir Ardestani-Jaafari and Erick Delage. Robust optimization of sums of piecewise linear functions with application to inventory problems. *Operations Research*, 64(2):474–494, 2016.
4. Amir Ardestani-Jaafari and Erick Delage. The value of flexibility in robust location–transportation problems. *Transportation Science*, 52(1):189–209, 2018.
5. Alper Atamtürk and Muhong Zhang. Two-stage robust network flow and design under demand uncertainty. *Operations Research*, 55(4):662–673, 2007.
6. Öykü Naz Attila, Agostinho Agra, Kerem Akartunalı, and Ashwin Arulselvan. Robust formulations for economic lot-sizing problem with remanufacturing. *European Journal of Operational Research*, 288(2):496–510, 2020.
7. Hannah Bakker, Fabian Dunke, and Stefan Nickel. A structuring review on multi-stage optimization under uncertainty: Aligning concepts from theory and practice. *Omega*, 96:102080, 2020.
8. Burcu Balcik and İhsan Yanıkoğlu. A robust optimization approach for humanitarian needs assessment planning under travel time uncertainty. *European Journal of Operational Research*, 282(1):40–57, 2020.
9. Roberto Baldacci, Marco Caserta, Emiliano Traversi, and Roberto Wolfler Calvo. Robustness of solutions to the capacitated facility location problem with uncertain demand. *Optimization Letters*, 16(9):2711–2727, 2022.
10. M Gisela Bardossy and S Raghavan. Robust optimization for the connected facility location problem. *Electronic Notes in Discrete Mathematics*, 44:149–154, 2013.
11. Opher Baron, Joseph Milner, and Hussein Naseraldin. Facility location: A robust optimization approach. *Production and Operations Management*, 20(5):772–785, 2011.

12 Mokhtar S Bazaraa, John J Jarvis, and Hanif D Sherali. *Linear programming and network flows*. John Wiley & Sons, 2011.

13 Aharon Ben-Tal, Byung Do Chung, Supreet Reddy Mandala, and Tao Yao. Robust optimization for emergency logistics planning: Risk mitigation in humanitarian relief supply chains. *Transportation Research Part B: Methodological*, 45(8):1177–1189, 2011.

14 Aharon Ben-Tal, Omar El Housni, and Vineet Goyal. A tractable approach for designing piecewise affine policies in two-stage adjustable robust optimization. *Mathematical Programming*, 182:57–102, 2020.

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

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

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

18 Dimitris Bertsimas and Constantine Caramanis. Adaptability via sampling. In *46th IEEE Conference on Decision and Control*, pages 4717–4722, 2007.

19 Dimitris Bertsimas and Dick den Hertog. *Robust and adaptive optimization*. Dynamic Ideas LLC, Belmont, MA, 2022.

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

21 Dimitris Bertsimas and Angelos Georghiou. Design of near optimal decision rules in multistage adaptive mixed-integer optimization. *Operations Research*, 63(3):610–627, 2015.

22 Dimitris Bertsimas and Angelos Georghiou. Binary decision rules for multistage adaptive mixed-integer optimization. *Mathematical Programming*, 167:395–433, 2018.

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

24 Dimitris Bertsimas, Vineet Goyal, and Brian Y Lu. A tight characterization of the performance of static solutions in two-stage adjustable robust linear optimization. *Mathematical Programming*, 150(2):281–319, 2015.

25 Dimitris Bertsimas, Dan A. Iancu, and Pablo A. Parrilo. Optimality of affine policies in multistage robust optimization. *Mathematics of Operations Research*, 35(2):363–394, 2010.

26 Dimitris Bertsimas and Melvyn Sim. The price of robustness. *Operations Research*, 52(1):35–53, 2004.

27 Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B Shah. Julia: A fresh approach to numerical computing. *SIAM Review*, 59(1):65–98, 2017.

28 Ali Borumand, Ahmadreza Marandi, Ali S Nookabadi, and Zümbül Atan. An oracle-based algorithm for robust planning of production routing problems in closed-loop supply chains of beverage glass bottles. *Omega*, 122:102939, 2024.

29 Viktoryia Buhayenko and Dick den Hertog. 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, 2017.

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

31 Zhi Chen and Peng Xiong. Rsome in python: An open-source package for robust stochastic optimization made easy. *INFORMS Journal on Computing*, 35(4):717–724, 2023.

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

33 Teodor Gabriel Crainic and Gilbert Laporte. Planning models for freight transportation. *European Journal of Operational Research*, 97(3):409–438, 1997.

34 Maryam Daryalal, Ayse N Arslan, and Merve Bodur. Two-stage and lagrangian dual decision rules for multistage adaptive robust optimization. *Optimization Online*, 2024.

35 Danique de Moor, Joris Wagenaar, Robert Poos, Dick den Hertog, and Hein Fleuren. A robust approach to food aid supply chains. *European Journal of Operational Research*, 318(1):269–285, 2024.

36 Frans JCT De Ruiter, Jianzhe Zhen, and Dick Den Hertog. Dual approach for two-stage robust nonlinear optimization. *Operations Research*, 71(5):1794–1799, 2023.

37 Erick Delage and Dan A Iancu. Robust multistage decision making. *INFORMS Tutorials in Operations Research*, pages 20–46, 2015.

38 Santanu S Dey and Diego A Morán R. Some properties of convex hulls of integer points contained in general convex sets. *Mathematical Programming*, 141(1):507–526, 2013.

39 Amin Ahmadi Digehsara, Menglei Ji, Amir Ardestani-Jaafari, and Hoda Bidkhori. Equity-driven facility location: A two-stage robust optimization approach. *Computers & Operations Research*, 176:106920, 2024.

40 Iain Dunning, Joey Huchette, and Miles Lubin. JuMP: A modeling language for mathematical optimization. *SIAM Review*, 59(2):295–320, 2017.

41 Hadi El-Amine, Ebru K Bish, and Douglas R Bish. Robust postdonation blood screening under prevalence rate uncertainty. *Operations Research*, 66(1):1–17, 2018.
42 Omar El Housni and Vineet Goyal. On the optimality of affine policies for budgeted uncertainty sets. *Mathematics of Operations Research*, 46(2):674–711, 2021.
43 Virginie Gabrel, Mathieu Lacroix, Cécile Murat, and Nabila Remli. Robust location transportation problems under uncertain demands. *Discrete Applied Mathematics*, 164:100–111, 2014.
44 Virginie Gabrel, Cécile Murat, and Aurélie Thiele. Recent advances in robust optimization: An overview. *European Journal of Operational Research*, 235(3):471–483, 2014.
45 Angelos Georghiou, Angelos Tsoukalas, and Wolfram Wiesemann. A primal–dual lifting scheme for two-stage robust optimization. *Operations Research*, 68(2):572–590, 2020.
46 Bram L Gorissen, İhsan Yanıkoğlu, and Dick Den Hertog. A practical guide to robust optimization. *Omega*, 53:124–137, 2015.
47 Gurobi Optimization. LLC: Gurobi optimizer reference manual, 2025.
48 Aldis Jakubovskis. Strategic facility location, capacity acquisition, and technology choice decisions under demand uncertainty: Robust vs. non-robust optimization approaches. *European Journal of Operational Research*, 260(3):1095–1104, 2017.
49 Hongcheng Jiang. Central warehouse planning within the service parts network of ASML. Master's thesis, Eindhoven University of Technology, the Netherlands, 2012.
50 Esther Julien, Krzysztof Postek, and Ş İlker Birbil. Machine learning for K-adaptability in two-stage robust optimization. *INFORMS Journal on Computing*, 2024.
51 Abbas Khademi and Ahmadreza Marandi. Quadratic optimization through the lens of adjustable robust optimization. *INFORMS Journal on Computing*, 2025.
52 Abbas Khademi, Ahmadreza Marandi, and Majid Soleimani-Damaneh. A new dual-based cutting plane algorithm for nonlinear adjustable robust optimization. *Journal of Global Optimization*, 89(3):559–595, 2024.
53 John G Klincewicz. Solving a freight transport problem using facility location techniques. *Operations Research*, 38(1):99–109, 1990.
54 Koninklijke Philips N.V. Philips comprehensive on-site service agreement, 2019.
55 Jannis Kurtz. Bounding the optimal number of policies for robust K-adaptability. *Optimization Online*, 2024.
56 Nikolaos H Lappas and Chrysanthos E Gounaris. Multi-stage adjustable robust optimization for process scheduling under uncertainty. *AIChE Journal*, 62(5):1646–1667, 2016.
57 Carlos Lara, Juan J Flores, and Felix Calderon. On the hyperbox–hyperplane intersection problem. *INFOCOMP Journal of Computer Science*, 8(4):21–27, 2009.
58 Sabine Limbourg and Bart Jourquin. Optimal rail-road container terminal locations on the European network. *Transportation Research Part E: Logistics and Transportation Review*, 45(4):551–563, 2009.
59 Ahmadreza Marandi and Dick Den Hertog. When are static and adjustable robust optimization problems with constraint-wise uncertainty equivalent? *Mathematical Programming*, 170:555–568, 2018.
60 Stefano Moret, Frédéric Babonneau, Michel Bierlaire, and François Maréchal. Decision support for strategic energy planning: A robust optimization framework. *European Journal of Operational Research*, 280(2):539–554, 2020.
61 Patrick E O'neil. Hyperplane cuts of an n-cube. *Discrete Mathematics*, 1(2):193–195, 1971.
62 Steven F Petit, Sebastiaan Breedveld, Jan Unkelbach, Dick den Hertog, and Marleen Balvert. Robust dose-painting-by-numbers vs. nonselective dose escalation for non-small cell lung cancer patients. *Medical Physics*, 48(6):3096–3108, 2021.
63 Krzysztof Postek and Dick den Hertog. Multistage adjustable robust mixed-integer optimization via iterative splitting of the uncertainty set. *INFORMS Journal on Computing*, 28(3):553–574, 2016.
64 Krzysztof Postek, Ward Romeijnders, and Wolfram Wiesemann. Multi-stage robust mixed-integer programming. *Operations Research*, 2025.
65 Ward Romeijnders and Krzysztof Postek. Piecewise constant decision rules via branch-and-bound based scenario detection for integer adjustable robust optimization. *INFORMS Journal on Computing*, 33(1):390–400, 2021.
66 Arkajyoti Roy, Shaunak S Dabadghao, and Ahmadreza Marandi. Value of intermediate imaging in adaptive robust radiotherapy planning to manage radioresistance. *Annals of Operations Research*, 339(3):1307–1328, 2024.
67 David Simchi-Levi, He Wang, and Yehua Wei. Increasing supply chain robustness through process flexibility and inventory. *Production and Operations Management*, 27(8):1476–1491, 2018.
68 O. Solyalı, J. F. Cordeau, and G. Laporte. The impact of modeling on robust inventory management under demand uncertainty. *Management Science*, 62(4):1188–1201, 2015.
69 Anirudh Subramanyam, Chrysanthos E Gounaris, and Wolfram Wiesemann. K-adaptability in two-stage mixed-integer robust optimization. *Mathematical Programming Computation*, 12:193–224, 2020.
70 Anirudh Subramanyam, Chrysanthos E Gounaris, and Wolfram Wiesemann. K-adaptability in two-stage mixed-integer robust optimization. *Mathematical Programming Computation*, 12:193–224, 2020.
71 Simon Thomä, Grit Walther, and Maximilian Schiffer. Designing tractable piecewise affine policies for multi-stage

adjustable robust optimization. *Mathematical Programming*, 208:661–716, 2024.
72 Xin Wang, Yong-Hong Kuo, Houcai Shen, and Lianmin Zhang. Target-oriented robust location-transportation problem with service-level measure. *Transportation Research Part B: Methodological*, 153:1–20, 2021.
73 Guanglin Xu and Samuel Burer. A copositive approach for two-stage adjustable robust optimization with uncertain right-hand sides. *Computational Optimization and Applications*, 70:33–59, 2018.
74 Yuwei Yang, Renjun Zhou, and Xiaohong Ran. Robust optimization with box set for reactive power optimization in wind power integrated system. In *2012 IEEE Power and Energy Society General Meeting, San Diego, CA*, pages 1–6, 2012.
75 İhsan Yanıkoğlu, Bram L Gorissen, and Dick den Hertog. A survey of adjustable robust optimization. *European Journal of Operational Research*, 277(3):799–813, 2019.
76 Long Zhao and Bo Zeng. An exact algorithm for two-stage robust optimization with mixed integer recourse problems. *Optimization Online*, 2012.
77 Jianzhe Zhen, Dick den Hertog, and Melvyn Sim. Adjustable robust optimization via Fourier–Motzkin elimination. *Operations Research*, 66(4):1086–1100, 2018.
78 Tengkuo Zhu, Stephen D. Boyles, and Avinash Unnikrishnan. Two-stage robust facility location problem with drones. *Transportation Research Part C: Emerging Technologies*, 137:103563, 2022.

Appendices

This paper contains three appendices. In Appendix A, we provide the formulation of a deterministic multi-stage LT problem. Appendix B provides the reformulation of the multi-stage adjustable robust LT problem (6) after the elimination of the equality restrictions. Finally, in Appendix C we provide the details of the numerical results.

A Deterministic Problem

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

$$\begin{aligned} \underline{Q}(d) = \sum_{\substack{j\in\mathcal{J}\\k\in\mathcal{K}\\t\in\mathcal{T}}} c_{0jk}d_{jkt} + \min_{y,S,z} \ & \sum_{i\in\mathcal{I}} f_i y_i + T\sum_{\substack{i\in\mathcal{I}\\k\in\mathcal{K}}} h_{ik}S_{ik} + \sum_{\substack{i\in\mathcal{I}\\j\in\mathcal{J}\\k\in\mathcal{K}\\t\in\mathcal{T}}} \left(c_{ijk} - (T-t)\,h_{ik} - c_{0jk}\right) z_{ijkt} \\ \text{s.t.} \quad & S_{ik} \le My_i, \qquad i\in\mathcal{I}, k\in\mathcal{K}, \\ & \sum_{\substack{j\in\mathcal{J}\\t\in\mathcal{T}}} z_{ijkt} \le S_{ik}, \qquad i\in\mathcal{I}, k\in\mathcal{K}, \\ & \sum_{i\in\mathcal{I}} z_{ijkt} \le d_{jkt}, \qquad j\in\mathcal{J}, k\in\mathcal{K}, t\in\mathcal{T}, \\ & S_{ik}, z_{ijkt} \in \mathbb{N}_0, \qquad i\in\mathcal{I}, j\in\mathcal{J}, k\in\mathcal{K}, t\in\mathcal{T}, \\ & y_i\in\{0,1\}, \qquad i\in\mathcal{I}. \end{aligned} \tag{19}$$

Theorem 3. *Consider the deterministic problem (19). 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, in an optimal solution, $z_{\bar{j}i_1\bar{k}t} = 0$ for any $t\in\mathcal{T}$.

Proof. Let us denote an optimal solution of (19) 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

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

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

and $y^{\text{new}} = y^*$. It is easy to check that $(y^{\text{new}}, S^{\text{new}}, z^{\text{new}})$ is a feasible solution for (19). Let us denote by Obj^* and Obj^{new} the optimal value and the objective value of the constructed solution, respectively. We have

$$Obj^* - Obj^{\text{new}} \quad = \quad z^*_{\bar{j}i_1\bar{k}\bar{t}}\left(\bar{t}\left(h_{i_1\bar{k}} - h_{i_2\bar{k}}\right) + c_{i_1\bar{j}\bar{k}} - c_{i_2\bar{j}\bar{k}}\right). \tag{20}$$

Also, we have

$$\bar{t}\left(h_{i_2\bar{k}} - h_{i_1\bar{k}}\right) \le \max\left\{h_{i_2\bar{k}} - h_{i_1\bar{k}}, T\left(h_{i_2\bar{k}} - h_{i_1\bar{k}}\right)\right\} < c_{i_1\bar{j}\bar{k}} - c_{i_2\bar{j}\bar{k}}, \tag{21}$$

where the right inequality is due to the assumption. Therefore, (20) and (21) imply that $Obj^* > Obj^{\text{new}}$, which contradicts optimality of (y^*, S^*, z^*). ◀

Corollary 4. *Let us fix $i_1, i_2\in\mathcal{I}$. For the deterministic problem (19), 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 all optimal solutions $y^*_{i_1} = 0$.

Proof. Let us assume by contradiction that in an 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 $y^1_i = y^*_i$ for $i \in \mathcal{I} \setminus \{i_1, i_2\}$, $y^1_{i_1} = 1$, and $y^1_{i_2} = 1$. 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 $z^3 = z^2$, $S^3 = S^2$, $y^3_i = y^2_i$ for $i \in \mathcal{I} \setminus \{i_1\}$, and $y^3_{i_1} = 0$, 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 Elimination of the Equality Constraints

In this section, we provide the explicit formulation of the multi-stage LT problem where the equality constraints as well as the variables corresponding to the shipments from the central warehouse to customers in (6) are eliminated.

$$\begin{aligned} LT(\mathcal{D}) := \min_{y,S}\ & \sum_{i\in\mathcal{I}} f_i y_i + \sum_{\substack{i\in\mathcal{I}\\k\in\mathcal{K}}} h_{ik} S_{ik} + \max_{d_{[1]}\in\mathcal{D}_1} Q_1(y, S, d_{[1]}, 0_{JK}) \\ \text{s.t. }\ & S_{ik} \le M y_i, && i\in\mathcal{I}, k\in\mathcal{K}, \\ & S_{ik} \in \mathbb{N}_0, && i\in\mathcal{I}, k\in\mathcal{K}, \\ & y_i \in \{0,1\}, && i\in\mathcal{I}, \end{aligned} \tag{6}$$

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

$$\begin{aligned} Q_t\left(y, S, d_{[t]}, z_{[t-1]}\right) = \min_{z_t}\ & \sum_{\substack{i\in\mathcal{I}\\j\in\mathcal{J}\\k\in\mathcal{K}}} c_{ijk} z_{ijkt} + \sum_{\substack{i\in\mathcal{I}\\k\in\mathcal{K}}} h_{ik} S_{ik} - \sum_{\substack{i\in\mathcal{I}\\k\in\mathcal{K}\\j\in\mathcal{J}}} h_{ik} \left(\sum_{\tilde{t}=1}^{t-1} z_{ijk\tilde{t}}\right) \\ & + \sum_{\substack{j\in\mathcal{J}\\k\in\mathcal{K}}} c_{0jk} \left(d_{jkt} - \sum_{i\in\mathcal{I}} z_{ijkt}\right) + \max_{d_{t+1}\in\mathcal{D}_{t+1}} Q_{t+1}\left(y, S, d_{[t+1]}, z_{[t]}\right) \\ \text{s.t. }\ & \sum_{j\in\mathcal{J}} z_{ijkt} \le S_{ik} - \sum_{j\in\mathcal{J}} \sum_{\tilde{t}=1}^{t-1} z_{ijk\tilde{t}}, && i\in\mathcal{I}, k\in\mathcal{K}, \\ & \sum_{i\in\mathcal{I}} z_{ijkt} \le d_{jkt}, && j\in\mathcal{J}, k\in\mathcal{K}, \\ & z_{ijkt} \in \mathbb{N}_0, && i\in\mathcal{I}, j\in\mathcal{J}, k\in\mathcal{K}, \end{aligned} \tag{22}$$

and

$$\begin{aligned} Q_T(y, S, d_{[T]}, z_{[T-1]}) = \min_{z_T}\ & \sum_{\substack{i\in\mathcal{I}\\j\in\mathcal{J}\\k\in\mathcal{K}}} c_{ijk} z_{ijkT} + \sum_{\substack{j\in\mathcal{J}\\k\in\mathcal{K}}} c_{0jk} \left(d_{jkT} - \sum_{i\in\mathcal{I}} z_{ijkT}\right) \\ \text{s.t. }\ & \sum_{j\in\mathcal{J}} z_{ijkT} \le S_{ik} - \sum_{j\in\mathcal{J}} \sum_{t=1}^{T-1} z_{ijkt}(d_{[t]}), && i\in\mathcal{I}, k\in\mathcal{K}, \\ & \sum_{i\in\mathcal{I}} z_{ijkT} \le d_{jkT}, && j\in\mathcal{J}, k\in\mathcal{K}, \\ & z_{ijkT} \in \mathbb{N}_0, && i\in\mathcal{I}, j\in\mathcal{J}, k\in\mathcal{K}. \end{aligned} \tag{23}$$

C Detailed Results of Numerical Experiments

C.1 Class One

Table 8 Results for Class One with $\hat{d} = 1$.

Instance	α	SM		SM$_{cut}$		FA		FA$_{cut}$		FASM		FASM$_{cut}$		SMFA		SMFA$_{cut}$		BFP		ADR		ADR$_{cut}$	
		LB	Time	LB	Time	LB	Time	LB	Time	LB	Time	LB	Time	LB	Time	LB	Time	LB	Time	UB	Time	UB	Time
1	0.60	1085.3230	24.47	1073.2960	7.25	1059.2960	0.09	1053.2960	0.08	1079.2960	11.78	1067.3230	1.97	1073.2960	8.46	1073.2960	5.71	1037.2960	3,600*	1089.5050	0.04	1089.5050	0.13
2	0.60	1054.2780	14.98	1078.2780	1.69	1042.2780	0.08	1046.2780	0.08	1042.2780	0.76	1048.2780	1.21	1048.2780	2.75	1042.2780	1.49	1022.2780	3,600*	1076.5500	0.04	1076.5500	0.11
3	0.60	1104.8500	36.94	1080.8500	2.53	1074.8500	0.10	1080.8500	0.07	1092.8500	2.30	1080.8500	1.75	1086.8500	4.43	1092.8500	4.17	1062.8500	3,600*	1102.9830	0.06	1102.9830	0.15
4	0.60	1086.3455	17.30	1086.3455	2.47	1068.3725	0.07	1074.3455	0.07	1086.3455	3.38	1086.3455	4.58	1086.3455	3.59	1080.3455	4.42	1056.3455	3,600*	1096.4960	0.05	1096.4960	0.16
5	0.60	1085.3050	34.64	1067.3050	1.44	1067.3050	0.07	1055.3050	0.07	1061.3050	4.50	1073.3050	1.46	1067.3050	4.97	1085.3050	3.07	1031.3050	3,600*	1089.5280	0.04	1089.5280	0.14
1	0.65	1110.8500	32.98	1104.8500	4.51	1078.8500	0.09	1086.8500	0.08	1086.8500	2.98	1086.8500	1.78	1098.8500	3.89	1098.8500	2.83	1060.8500	3,600*	1121.0050	0.04	1121.0050	0.18
2	0.65	1117.3725	24.76	1093.3725	1.55	1081.3725	0.08	1093.3725	0.07	1093.3725	1.93	1093.3725	1.70	1081.3725	1.68	1093.3725	4.17	1079.3725	3,600*	1115.4870	0.04	1115.4870	0.13
3	0.65	1130.4085	38.52	1112.4085	5.39	1106.4085	0.08	1100.4085	0.07	1106.4085	2.51	1112.4085	5.89	1124.4085	5.00	1118.4085	3.09	1082.4085	3,600*	1128.4740	0.05	1128.4740	0.13
4	0.65	1136.9085	42.26	1130.9085	2.22	1094.9085	0.08	1100.9085	0.07	1130.9085	3.78	1130.9085	2.48	1118.9085	2.91	1136.9085	5.15	1094.9085	3,600*	1134.9420	0.04	1134.9420	0.15
5	0.65	1092.8545	26.26	1104.8545	2.83	1074.8545	0.07	1080.8545	0.08	1092.8545	3.22	1098.8545	1.90	1080.8545	2.17	1092.8545	2.40	1072.8545	3,600*	1115.0100	0.04	1115.0100	0.17
1	0.70	1149.9445	28.62	1131.9445	2.65	1113.9445	0.07	1125.9445	0.08	1161.9445	10.39	1143.9445	4.05	1143.9445	3.25	1149.9445	5.63	1113.9445	3,600*	1159.9560	0.04	1159.9560	0.21
2	0.70	1124.3995	21.13	1124.3995	1.54	1118.3995	0.09	1112.3995	0.07	1118.3995	2.75	1130.3995	1.04	1118.3995	2.81	1112.3995	2.81	1094.3995	3,600*	1140.4870	0.05	1140.4870	0.28
3	0.70	1149.9535	48.00	1149.9805	2.33	1137.9535	0.08	1125.9805	0.08	1143.9535	10.39	1137.9535	3.74	1137.9535	5.43	1137.9535	2.27	1113.9535	3,600*	1147.9420	0.05	1147.9420	0.16
4	0.70	1156.4535	33.35	1156.4535	6.35	1144.4535	0.08	1156.4535	0.07	1144.4535	2.36	1144.4535	1.87	1144.4535	2.30	1150.4535	2.25	1138.4535	3,600*	1166.4240	0.05	1166.4240	0.22
5	0.70	1118.9220	30.24	1118.9220	2.49	1106.9220	0.09	1130.9220	0.07	1118.9220	1.21	1124.9220	2.95	1124.9220	4.25	1124.9220	2.12	1094.9220	3,600*	1140.9780	0.05	1140.9780	0.32
1	0.75	1150.4535	10.66	1150.4805	2.29	1156.4535	0.07	1144.4805	0.07	1138.4805	0.68	1132.4535	1.22	1150.4535	2.26	1150.4535	2.63	1120.4535	3,600*	1166.4560	0.04	1166.4550	0.37
2	0.75	1162.4490	20.93	1156.4490	1.95	1132.4490	0.07	1156.4490	0.08	1132.4490	1.18	1156.4490	1.05	1138.4490	1.30	1168.4490	3.51	1120.4490	3,600*	1166.4420	0.04	1166.4420	0.39
3	0.75	1151.5075	30.90	1199.5345	9.22	1157.5075	0.07	1175.5075	0.07	1175.5075	5.79	1181.5075	1.13	1157.5075	1.80	1163.5075	2.40	1157.5075	3,600*	1197.4740	0.04	1197.4730	0.41
4	0.75	1189.0435	22.12	1189.0435	1.67	1171.0435	0.08	1183.0435	0.08	1201.0435	3.55	1189.0435	3.82	1189.0435	4.41	1189.0435	3.15	1153.0435	3,600*	1198.9110	0.04	1198.9100	0.49
5	0.75	1168.4895	24.42	1138.4625	2.19	1132.4625	0.08	1126.4625	0.08	1144.4625	3.18	1162.4625	4.00	1144.4625	2.74	1150.4625	2.50	1120.4625	3,600*	1166.4650	0.04	1166.4640	0.46
1	0.80	1195.0300	21.67	1195.0300	3.17	1165.0300	0.09	1201.0300	0.07	1189.0300	1.17	1183.0570	1.41	1183.0300	3.61	1183.0300	1.94	1165.0300	3,600*	1210.9380	0.05	1210.9380	0.46
2	0.80	1189.0390	29.65	1177.0390	2.15	1159.0390	0.08	1183.0390	0.07	1189.0390	5.33	1195.0390	2.21	1201.0390	4.72	1195.0390	3.33	1153.0390	3,600*	1216.9330	0.05	1216.9330	0.47
3	0.80	1201.5525	36.52	1201.5525	4.51	1171.5525	0.08	1183.5525	0.08	1183.5525	1.26	1189.5525	1.15	1213.5525	3.67	1189.5525	1.61	1171.5525	3,600*	1229.4420	0.05	1229.4420	0.46
4	0.80	1208.0750	17.08	1238.0750	9.74	1220.0750	0.07	1202.0750	0.07	1214.0750	1.55	1208.0750	1.07	1220.0750	3.98	1202.0750	1.96	1184.0750	3,600*	1235.9330	0.05	1235.9330	0.51
5	0.80	1176.0165	13.13	1170.0165	5.84	1158.0165	0.07	1170.0165	0.07	1182.0165	3.07	1170.0165	2.85	1170.0165	2.17	1170.0165	1.86	1146.0165	3,600*	1197.9780	0.05	1197.9780	0.44
1	0.85	1233.6020	27.06	1233.6020	0.89	1221.6020	0.07	1221.6020	0.07	1227.6290	2.44	1233.6020	5.43	1227.6020	5.06	1251.6020	8.12	1197.6020	3,600*	1261.4200	0.04	1261.4200	0.42
2	0.85	1246.6830	21.01	1252.6560	4.89	1222.6560	0.07	1210.6560	0.08	1240.6560	1.67	1276.6560	6.05	1246.6560	4.89	1258.6560	2.16	1222.6560	3,600*	1286.4330	0.04	1286.4330	0.47
3	0.85	1282.6515	23.26	1246.6515	2.45	1234.6515	0.07	1246.6515	0.08	1240.6515	1.68	1246.6515	0.69	1252.6515	3.09	1258.6515	5.16	1234.6515	3,600*	1286.4290	0.05	1286.4290	0.44
4	0.85	1240.1380	16.09	1240.1380	5.87	1228.1380	0.07	1222.1380	0.07	1228.1380	1.62	1222.1380	1.14	1228.1380	1.50	1246.1380	5.69	1216.1380	3,600*	1267.9470	0.05	1267.9460	0.43
5	0.85	1226.5930	24.31	1214.5930	6.04	1208.5930	0.07	1208.5930	0.07	1226.5930	4.77	1220.5930	3.35	1208.5930	2.95	1202.5930	4.05	1178.5930	3,600*	1248.4600	0.05	1248.4600	0.48
1	0.90	1284.1875	19.81	1284.1875	2.86	1266.1875	0.08	1230.1875	0.07	1278.2145	5.79	1278.1875	1.00	1290.1875	5.14	1296.1875	8.09	1240.1875	3,600*	1323.9560	0.05	1323.9550	0.43
2	0.90	1334.2550	31.98	1292.2550	4.33	1292.2550	0.08	1268.2550	0.07	1316.2550	3.50	1298.2550	4.49	1298.2550	3.26	1316.2550	5.37	1256.2550	3,600*	1337.9020	0.04	1337.9020	0.51
3	0.90	1291.2100	24.29	1279.2100	2.06	1267.2370	0.10	1255.2100	0.07	1273.2100	1.14	1273.2100	1.01	1285.2100	3.78	1297.2100	5.40	1249.2100	3,600*	1336.9200	0.05	1336.9200	0.50
4	0.90	1297.7775	17.10	1345.7505	9.03	1261.7775	0.07	1249.7505	0.08	1297.7505	3.09	1297.7505	0.97	1303.7505	4.57	1279.7505	2.75	1261.7505	3,600*	1343.4470	0.04	1343.4470	0.48
5	0.90	1308.1920	34.72	1272.1920	5.38	1272.1920	0.09	1254.1920	0.07	1266.1920	4.63	1272.1920	3.61	1278.1920	4.98	1278.1920	4.57	1254.1920	3,600*	1317.9290	0.04	1317.9280	0.51
1	0.95	1341.8135	15.50	1329.8135	1.58	1311.8135	0.07	1293.8135	0.07	1323.8135	1.22	1341.8135	3.88	1347.8135	6.72	1317.8135	2.08	1269.8135	3,600*	1387.4380	0.04	1387.4380	0.55
2	0.95	1403.8540	18.71	1337.8540	3.77	1349.8540	0.07	1361.8540	0.08	1355.8540	4.88	1349.8540	2.68	1349.8540	5.35	1349.8540	2.62	1307.8540	3,600*	1413.3880	0.04	1413.3880	0.46
3	0.95	1386.9035	14.12	1362.9035	1.68	1368.9035	0.08	1374.9035	0.07	1404.9035	11.97	1386.9035	1.15	1368.9035	2.38	1398.9035	6.05	1344.9035	3,600*	1438.3970	0.05	1438.3970	0.61
4	0.95	1430.9575	22.06	1376.9575	2.77	1400.9575	0.07	1406.9575	0.07	1394.9845	3.94	1406.9575	1.02	1418.9575	3.86	1388.9575	2.32	1358.9575	3,600*	1464.3930	0.06	1464.3930	0.38
5	0.95	1366.8450	21.99	1366.8450	10.17	1354.8450	0.10	1330.8720	0.09	1354.8450	1.45	1354.8450	5.39	1354.8450	7.58	1384.8450	11.06	1306.8450	3,600*	1406.4470	0.05	1406.4460	0.39

Table 9 Results for Class One with $\hat{d} = 2$.

Instance	α	SM		SM$_{cut}$		FA		FA$_{cut}$		FASM		FASM$_{cut}$		SMFA		SMFA$_{cut}$		BFP		ADR		ADR$_{cut}$	
		LB	Time	LB	Time	LB	Time	LB	Time	LB	Time	LB	Time	LB	Time	LB	Time	LB	Time	UB	Time	UB	Time
1	0.60	1804.8350	22.78	1816.8080	8.89	1774.8080	0.08	1792.8080	0.08	1816.8080	3.53	1816.8080	8.88	1816.8080	11.42	1816.8080	3.43	1720.8080	3,600*	1812.9970	0.04	1812.9970	0.63
2	0.60	1729.1730	15.92	1747.1730	3.40	1753.1730	0.08	1711.1730	0.07	1765.1730	7.10	1723.1730	3.72	1765.1730	6.84	1723.1730	2.76	1681.1730	3,600*	1767.5510	0.04	1767.5510	0.37
3	0.60	1810.2990	38.16	1804.2990	2.04	1756.2990	0.10	1756.2990	0.07	1822.3260	6.77	1774.2990	2.84	1822.2990	8.10	1792.2990	5.17	1714.2990	3,600*	1818.5240	0.04	1818.5240	0.55
4	0.60	1854.3395	45.00	1836.3395	2.65	1794.3395	0.10	1818.3395	0.07	1818.3395	1.46	1842.3395	4.62	1818.3395	3.63	1842.3395	9.39	1752.3395	3,600*	1850.4920	0.04	1850.4920	0.58
5	0.60	1773.2675	51.45	1791.2405	8.80	1737.2405	0.10	1773.2405	0.08	1767.2405	1.21	1791.2405	6.01	1785.2405	5.84	1755.2405	1.96	1683.2405	3,600*	1787.5100	0.04	1787.5100	0.65
1	0.65	1854.8575	27.76	1872.8575	5.96	1854.8575	0.08	1818.8575	0.08	1830.8575	1.91	1806.8575	1.14	1848.8575	5.14	1848.8575	6.17	1764.8575	3,600*	1875.0240	0.05	1875.0230	0.63
2	0.65	1773.7540	26.16	1797.7540	2.38	1749.7540	0.07	1773.7540	0.07	1821.7540	7.14	1791.7540	5.79	1785.7540	3.36	1785.7540	7.91	1743.7540	3,600*	1818.0370	0.09	1818.0370	0.36
3	0.65	1848.8845	23.95	1824.8845	4.35	1788.8845	0.08	1800.8845	0.08	1830.8845	5.90	1830.8845	2.38	1848.8845	5.99	1824.8845	4.07	1764.8845	3,600*	1869.0330	0.04	1869.0330	0.63
4	0.65	1899.9385	17.53	1857.9655	7.00	1869.9385	0.10	1845.9385	0.08	1881.9385	2.59	1869.9385	2.76	1875.9385	2.72	1869.9385	4.36	1839.9385	3,600*	1919.9560	0.04	1919.9560	0.69
5	0.65	1822.2810	24.82	1822.2810	3.84	1732.2810	0.08	1774.2810	0.08	1786.2810	1.13	1798.2810	3.56	1804.2810	6.61	1798.2810	5.29	1714.2810	3,600*	1824.5730	0.04	1824.5730	0.50
1	0.70	1916.9250	18.37	1892.9250	2.49	1814.9250	0.08	1844.9250	0.08	1850.9250	0.98	1892.9250	4.73	1880.9250	5.37	1844.9250	2.17	1814.9250	3,600*	1913.0460	0.04	1913.0460	0.52
2	0.70	1866.8350	24.03	1800.8350	3.44	1824.8620	0.07	1806.8350	0.08	1812.8350	0.86	1812.8350	0.74	1812.8350	1.42	1812.8350	2.94	1806.8350	3,600*	1881.0240	0.04	1881.0240	0.50
3	0.70	1852.4970	25.63	1876.4970	2.99	1888.4970	0.08	1804.4970	0.07	1912.4970	4.66	1912.4970	10.25	1888.4970	2.06	1864.4970	2.16	1822.4970	3,600*	1932.5240	0.04	1932.5240	0.79
4	0.70	1940.5735	32.85	1910.5735	3.66	1940.5735	0.08	1922.5735	0.08	1970.6005	3.55	1940.5735	2.05	1940.5735	5.34	1940.5735	5.38	1886.5735	3,600*	1984.3840	0.04	1984.3840	0.62
5	0.70	1862.3980	24.18	1832.4250	2.13	1826.3980	0.07	1814.3980	0.08	1862.3980	3.16	1826.3980	1.06	1868.3980	6.35	1856.3980	7.60	1772.3980	3,600*	1894.5010	0.04	1894.5010	0.57
1	0.75	1913.4970	13.65	1877.4970	2.71	1889.4970	0.07	1889.4970	0.07	1931.4970	6.58	1883.4970	2.03	1889.4970	2.72	1919.4970	10.38	1859.4970	3,600*	1963.5460	0.04	1963.5460	0.58
2	0.75	1918.4745	40.46	1918.4745	7.25	1894.4745	0.08	1846.4745	0.08	1888.4745	2.57	1900.4745	4.28	1894.4745	5.72	1912.4745	4.06	1840.4745	3,600*	1938.5370	0.03	1938.5370	0.55
3	0.75	1933.6005	18.89	1945.6005	6.04	1909.6005	0.07	1873.6005	0.07	1921.6005	7.30	1933.6005	2.94	1933.6005	5.36	1957.6005	7.41	1825.6005	3,600*	1977.5150	0.05	1977.5140	0.53
4	0.75	2004.1815	31.74	2028.1815	4.32	1974.1815	0.07	2010.1815	0.08	1974.1815	0.81	2022.1815	4.62	1974.1815	1.50	2034.1815	7.95	1956.1815	3,600*	2041.8620	0.04	2041.8620	0.93
5	0.75	1912.4700	26.94	1858.4700	3.63	1900.4700	0.08	1870.4700	0.07	1918.4700	5.68	1882.4700	2.20	1882.4700	2.15	1894.4700	4.30	1834.4700	3,600*	1938.5370	0.04	1938.5370	0.59
1	0.80	1982.0915	31.01	2006.1185	12.64	1928.0915	0.08	1880.0915	0.09	1982.0915	11.75	1946.0915	3.41	1976.0915	9.62	1946.0915	6.80	1886.0915	3,600*	2020.0550	0.05	2020.0550	0.53
2	0.80	1945.0420	34.42	1957.0420	4.11	1909.0420	0.08	1927.0420	0.08	1903.0420	3.96	1963.0420	5.17	1927.0420	4.53	1939.0420	6.25	1867.0420	3,600*	2001.0420	0.05	2001.0420	0.60
3	0.80	1972.7085	17.94	1984.6815	7.61	1942.6815	0.07	1972.6815	0.07	1972.6815	4.17	2014.6815	5.79	1966.6815	4.65	1972.6815	3.93	1942.6815	3,600*	2052.5150	0.04	2052.5150	0.60
4	0.80	2079.7985	22.56	2031.7985	4.35	2049.7985	0.07	2037.8255	0.07	2049.7985	1.19	2097.7985	6.58	2079.7985	8.86	2097.7985	9.98	1977.7985	3,600*	2105.3620	0.04	2105.3620	0.62
5	0.80	1982.0735	22.69	1982.0735	12.11	1892.0735	0.07	1898.0735	0.08	1958.0735	6.78	1940.0735	1.91	1940.0735	5.55	1934.0735	4.24	1868.0735	3,600*	2008.0280	0.04	2008.0280	0.48
1	0.85	2109.2715	31.87	2035.2445	12.07	1981.2445	0.09	2023.2445	0.07	2041.2445	6.42	2023.2445	1.03	2085.2445	12.29	2089.2445	13.40	1981.2445	3,600*	2105.0100	0.05	2105.0100	0.51
2	0.85	2045.1950	38.32	1967.1680	5.87	2009.1680	0.08	1955.1680	0.10	1985.1680	1.44	1991.1950	3.01	2021.1680	9.47	2027.1680	9.14	1919.1680	3,600*	2065.0150	0.05	2065.0150	0.45
3	0.85	2090.3030	26.84	2060.3300	7.81	2006.3030	0.08	1982.3030	0.07	2024.3030	1.18	2096.3030	16.16	2024.3030	4.56	2036.3030	1.63	1988.3030	3,600*	2116.0100	0.04	2116.0100	0.66
4	0.85	2167.9380	49.01	2131.9380	3.59	2143.9380	0.09	2077.9380	0.32	2125.9380	9.16	2125.9380	3.86	2109.9380	5.58	2119.9380	4.44	2053.9380	3,600*	2193.3750	0.04	2193.3750	0.65
5	0.85	2045.6815	25.56	1997.6815	4.87	1979.6815	0.07	1979.6815	0.08	2003.6815	1.41	2039.6815	6.30	1999.6815	5.93	2081.6815	8.60	1913.6815	3,600*	2077.5060	0.03	2077.5060	0.59
1	0.90	2080.8525	20.30	2044.8525	2.39	2014.8525	0.08	2068.8525	0.07	2074.8525	5.12	2062.8795	3.79	2136.8525	14.63	2166.8525	14.38	2014.8525	3,600*	2162.5240	0.05	2162.5240	0.58
2	0.90	2073.8075	21.60	2031.8075	5.02	2007.8075	0.07	2049.8345	0.07	2151.8075	12.42	2115.8075	6.35	2061.8075	3.35	2097.8075	6.08	2007.8075	3,600*	2147.4650	0.04	2147.4650	0.64
3	0.90	2142.9380	17.10	2142.9380	9.78	2114.9380	0.07	2064.9380	0.07	2100.9380	2.35	2082.9380	4.81	2172.9380	16.76	2142.9380	2.47	2010.9380	3,600*	2182.5190	0.03	2182.5190	0.70
4	0.90	2234.0865	33.34	2166.0865	3.15	2148.0865	0.07	2190.0865	0.08	2202.0865	8.45	2154.0865	1.15	2178.0865	7.17	2234.0865	6.07	2136.0865	3,600*	2259.4070	0.03	2259.4070	0.70
5	0.90	2122.8120	18.03	2072.8120	8.72	1988.8120	0.07	1988.8120	0.08	2122.8120	8.81	2054.8120	0.89	2074.8120	9.56	2084.8120	2.38	1964.8120	3,600*	2142.5870	0.03	2142.5870	0.69
1	0.95	2272.5955	32.49	2214.5955	8.81	2178.6225	0.07	2144.5955	0.06	2178.5955	3.12	2214.5955	2.60	2178.5955	2.80	2232.5955	3.60	2156.5955	3,600*	2267.9920	0.03	2267.9920	0.66
2	0.95	2226.0730	19.19	2184.0730	4.66	2174.0730	0.07	2100.0730	0.07	2266.0730	13.85	2184.1000	3.49	2266.0730	17.01	2238.0730	7.47	2076.0730	3,600*	2261.4560	0.04	2261.4560	0.57
3	0.95	2259.1270	18.00	2167.1540	3.50	2131.1270	0.07	2169.1270	0.07	2185.1270	1.20	2193.1270	2.90	2191.1270	5.60	2193.1270	3.18	2125.1270	3,600*	2274.5640	0.03	2274.5640	0.84
4	0.95	2357.3655	30.31	2331.3115	4.88	2215.3115	0.08	2227.3115	0.09	2269.3115	2.91	2331.3115	2.76	2271.3115	3.89	2349.3115	5.11	2205.3115	3,600*	2352.4250	0.03	2352.4250	0.65
5	0.95	2207.0460	15.25	2233.0460	9.43	2141.0460	0.08	2201.0460	0.07	2161.0460	1.79	2171.0460	6.89	2161.0460	4.10	2171.0460	4.92	2099.0460	3,600*	2248.5460	0.03	2248.5460	0.70

Table 10 Results for Class One with $\hat{d} = 3$.

Instance	α	SM		SM$_{cut}$		FA		FA$_{cut}$		FASM		FASM$_{cut}$		SMFA		SMFA$_{cut}$		BFP		ADR		ADR$_{cut}$	
		LB	Time	LB	Time	LB	Time	LB	Time	LB	Time	LB	Time	LB	Time	LB	Time	LB	Time	UB	Time	UB	Time
1	0.60	2428.9510	62.66	2390.9510	0.92	2366.9510	0.08	2380.9510	0.07	2402.9510	3.04	2452.9510	9.86	2390.9510	3.44	2390.9510	1.37	2336.95	3,600*	2447.5240	0.03	2447.5240	0.69
2	0.60	2351.8475	35.01	2325.8205	6.55	2313.8205	0.08	2295.8205	0.07	2301.8205	3.62	2321.8205	5.21	2289.8205	1.60	2351.8205	7.88	2289.82	3,600*	2370.6100	0.04	2370.6100	0.72
3	0.60	2419.4510	31.89	2419.4780	10.67	2341.4510	0.09	2347.4510	0.08	2353.4510	3.25	2419.4510	10.94	2415.4510	7.58	2395.4510	7.23	2329.45	3,600*	2428.0600	0.03	2428.0600	0.62
4	0.60	2518.2075	35.99	2462.1805	2.38	2488.1535	0.07	2474.1535	0.07	2488.1535	4.33	2506.1535	3.25	2518.1535	9.30	2518.1535	2.48	2372.15	3,600*	2512.4480	0.03	2512.4480	0.69
5	0.60	2420.9375	33.84	2316.9105	4.49	2322.9105	0.07	2348.9105	0.09	2406.9105	5.85	2384.9105	2.55	2388.9105	6.88	2384.9105	5.08	2322.91	3,600*	2415.5690	0.03	2415.5690	0.77
1	0.65	2505.6580	30.91	2451.6040	3.02	2449.6040	0.07	2443.6040	0.08	2473.6040	1.18	2421.6040	0.79	2451.6040	4.57	2421.6040	0.66	2389.60	3,600*	2508.0330	0.03	2508.0330	0.44
2	0.65	2434.4735	27.44	2440.4735	12.81	2372.4735	0.08	2390.4735	0.10	2422.4735	3.36	2408.4735	5.35	2402.4735	8.49	2428.4735	8.82	2324.47	3,600*	2443.1280	0.04	2443.1280	0.54
3	0.65	2416.5500	36.15	2440.5500	8.70	2368.5770	0.08	2392.5500	0.07	2422.5500	2.06	2466.5500	4.61	2382.5500	1.97	2424.5500	2.05	2368.55	3,600*	2469.0960	0.03	2469.0960	0.67
4	0.65	2550.7615	20.67	2544.7615	3.70	2524.7345	0.09	2524.7345	0.08	2552.7345	4.99	2524.7345	2.52	2460.7345	1.05	2552.7345	7.66	2460.73	3,600*	2546.9970	0.03	2546.9970	0.70
5	0.65	2468.0050	30.85	2398.0050	2.72	2386.0050	0.08	2392.0050	0.09	2468.0050	3.06	2416.0050	3.23	2412.0050	2.38	2406.0050	2.83	2376.01	3,600*	2462.5830	0.03	2462.5830	0.63
1	0.70	2545.2525	28.39	2539.2255	8.93	2489.2255	0.07	2501.2525	0.09	2519.2525	3.47	2551.2255	10.98	2509.2255	3.64	2545.2255	4.54	2471.23	3,600*	2553.5240	0.03	2553.5240	0.46
2	0.70	2425.0545	20.62	2479.0545	5.70	2387.0545	0.07	2351.0545	0.07	2423.0545	4.25	2453.0545	7.10	2423.0545	5.51	2411.0545	4.28	2387.05	3,600*	2475.6550	0.04	2475.6550	0.61
3	0.70	2539.7300	32.70	2537.7300	5.24	2457.7300	0.08	2451.7300	0.07	2487.7300	3.25	2481.7300	0.93	2453.7300	1.93	2493.7300	8.75	2453.73	3,600*	2534.1190	0.04	2534.1190	0.53
4	0.70	2591.8650	28.13	2589.8380	5.07	2575.8380	0.07	2533.8380	0.07	2511.8380	0.62	2553.8380	2.30	2505.8380	1.10	2547.8380	1.43	2505.84	3,600*	2586.0190	0.02	2586.0200	0.50
5	0.70	2499.0815	35.14	2499.0815	3.84	2419.0815	0.08	2415.1085	0.07	2455.1085	2.94	2445.1085	3.23	2499.0815	8.69	2451.0815	4.29	2389.08	3,600*	2501.5920	0.03	2501.5920	0.59
1	0.75	2615.8920	19.40	2617.8650	7.71	2589.8650	0.08	2547.8650	0.07	2617.8650	1.99	2615.8920	6.06	2617.8650	9.41	2613.8650	6.56	2543.87	3,600*	2612.0150	0.03	2612.0150	0.63
2	0.75	2536.1985	43.90	2538.2255	6.62	2452.2255	0.08	2490.2255	0.08	2542.1985	4.39	2464.1985	1.27	2546.1985	14.08	2546.1985	3.28	2452.20	3,600*	2540.5780	0.04	2540.5780	0.48
3	0.75	2602.8740	29.64	2604.8740	11.49	2604.8740	0.09	2504.8740	0.07	2590.8740	1.18	2564.8740	5.53	2530.8740	1.38	2604.8740	10.98	2530.87	3,600*	2599.0640	0.03	2599.0640	0.67
4	0.75	2661.5045	19.58	2663.5045	5.55	2637.5045	0.08	2569.5045	0.07	2653.5045	1.70	2663.5045	5.20	2637.5045	3.51	2661.5045	6.81	2587.50	3,600*	2657.4650	0.03	2657.4650	0.69
5	0.75	2572.2705	28.61	2498.2435	7.00	2490.2435	0.07	2516.2435	0.07	2550.2435	6.71	2514.2435	1.87	2492.2435	1.96	2538.2435	5.13	2472.24	3,600*	2566.5640	0.03	2566.5640	0.69
1	0.80	2662.4820	25.75	2676.4820	8.91	2618.4820	0.07	2636.4820	0.07	2636.4820	3.58	2648.4820	1.75	2608.4820	2.83	2632.4820	4.58	2606.48	3,600*	2670.4750	0.01	2670.4750	0.75
2	0.80	2589.9235	28.98	2617.8965	8.66	2549.8965	0.07	2511.8965	0.08	2617.8965	5.30	2617.8965	2.04	2549.8965	2.55	2617.8965	2.98	2513.90	3,600*	2612.0780	0.03	2612.0780	0.60
3	0.80	2657.0360	35.88	2656.9820	3.24	2598.9820	0.07	2630.9820	0.06	2604.9820	2.86	2656.9820	5.10	2588.9820	2.06	2656.9820	4.00	2588.98	3,600*	2651.0240	0.04	2651.0240	0.67
4	0.80	2702.6395	13.55	2670.6395	5.76	2656.6125	0.07	2616.6125	0.07	2674.6125	1.56	2686.6125	2.22	2634.6125	1.85	2702.6125	7.44	2634.61	3,600*	2696.5060	0.03	2696.5060	0.84
5	0.80	2595.3560	26.27	2611.3560	5.35	2547.3560	0.08	2577.3560	0.08	2547.3560	2.00	2577.3560	2.78	2577.3560	5.14	2585.3560	9.40	2501.36	3,600*	2605.5870	0.03	2605.5870	0.69
1	0.85	2709.1440	25.72	2735.1170	5.63	2677.1170	0.07	2631.1440	0.07	2735.1170	5.70	2703.1170	0.96	2709.1170	6.66	2735.1170	2.64	2637.12	3,600*	2728.9070	0.03	2728.9070	0.76
2	0.85	2688.0810	33.09	2696.0810	10.73	2634.0810	0.07	2616.0810	0.08	2694.0810	8.10	2696.0810	4.90	2634.0810	1.75	2696.0810	5.62	2634.08	3,600*	2690.0100	0.01	2690.0110	0.77
3	0.85	2728.7295	24.94	2728.6755	2.06	2702.6755	0.08	2702.6755	0.07	2712.6755	2.21	2708.6755	2.65	2666.6755	1.20	2728.6755	4.32	2666.68	3,600*	2722.4970	0.03	2722.4970	0.72
4	0.85	2780.8150	13.24	2780.7880	2.93	2718.7880	0.08	2700.7880	0.07	2780.7880	4.68	2780.7880	0.83	2718.7880	1.88	2760.7880	6.18	2718.79	3,600*	2774.4390	0.03	2774.4390	0.92
5	0.85	2709.0945	14.02	2709.0945	1.99	2647.0945	0.07	2695.0945	0.06	2709.0945	5.18	2709.0945	12.72	2689.0945	5.15	2671.0945	8.71	2621.09	3,600*	2702.9830	0.03	2702.9830	0.78
1	0.90	2800.2970	22.50	2780.2700	5.40	2784.2700	0.07	2750.2700	0.07	2780.2700	2.97	2780.2970	2.34	2768.2700	4.14	2800.2700	2.33	2702.27	3,600*	2793.9020	0.03	2793.9020	0.88
2	0.90	2760.7700	20.36	2766.7700	3.52	2728.7970	0.07	2742.7700	0.08	2778.7700	5.98	2760.7700	1.04	2780.7700	7.33	2780.7700	3.78	2724.77	3,600*	2774.4290	0.03	2774.4300	0.82
3	0.90	2793.8510	31.11	2767.8240	2.55	2785.8240	0.06	2725.8240	0.07	2761.8240	2.36	2793.8240	6.33	2779.8240	5.00	2793.8240	3.30	2707.82	3,600*	2787.5060	0.03	2787.5060	0.91
4	0.90	2878.5220	27.02	2878.5220	0.78	2856.5220	0.09	2844.5220	0.07	2874.5490	1.04	2878.5220	4.05	2828.5220	2.22	2878.5220	2.36	2828.52	3,600*	2871.9160	0.03	2871.9160	1.02
5	0.90	2780.3420	27.79	2800.3150	0.78	2732.3150	0.07	2762.3150	0.07	2800.3150	2.70	2792.3150	1.17	2792.3150	3.40	2800.3150	6.29	2732.32	3,600*	2793.9290	0.03	2793.9290	0.75
1	0.95	2956.7740	28.31	2956.6930	3.69	2912.6930	0.08	2956.6930	0.08	2956.6930	1.36	2956.6930	0.81	2912.6930	2.53	2956.6930	7.19	2876.69	3,600*	2949.8530	0.02	2949.8530	0.77
2	0.95	2872.0085	31.28	2871.9815	2.25	2833.9815	0.07	2871.9815	0.06	2871.9815	3.47	2871.9815	3.27	2833.9815	3.08	2871.9815	3.95	2785.98	3,600*	2865.4210	0.04	2865.4200	0.94
3	0.95	2917.6300	11.82	2879.6300	3.52	2917.6300	0.06	2855.6300	0.07	2917.6300	1.47	2885.6300	1.52	2891.6300	2.10	2917.6300	2.69	2843.63	3,600*	2911.0060	0.02	2911.0060	1.05
4	0.95	2969.7515	25.93	2913.7245	3.38	2919.7245	0.08	2925.7245	0.08	2969.7245	2.03	2969.7245	2.57	2969.7245	4.83	2969.7245	1.38	2889.72	3,600*	2962.9070	0.03	2962.9070	0.95
5	0.95	2909.1075	30.64	2911.1075	2.16	2911.1075	0.06	2911.1075	0.07	2855.1075	1.10	2891.1075	5.45	2911.1075	5.27	2911.1075	3.47	2771.11	3,600*	2904.4570	0.03	2904.4570	0.77

Table 11 Results for Class One with $\hat{d} = 4$.

Instance	α	SM		SM$_{cut}$		FA		FA$_{cut}$		FASM		FASM$_{cut}$		SMFA		SMFA$_{cut}$		BFP		ADR		ADR$_{cut}$	
		LB	Time	LB	Time	LB	Time	LB	Time	LB	Time	LB	Time	LB	Time	LB	Time	LB	Time	UB	Time	UB	Time
1	0.60	3165.2465	12.82	3165.2195	3.20	3165.2195	0.24	3165.2195	0.06	3165.2195	2.71	3165.2195	3.88	3165.2195	4.60	3165.2195	2.30	3103.2195	3600*	3157.3670	0.03	3157.3670	0.84
2	0.60	3008.8415	28.41	3008.7875	2.31	2982.7875	0.07	3006.7875	0.07	3008.7875	2.05	3008.7875	1.80	3008.7875	4.06	2994.7875	4.59	2952.7875	3600*	3001.4970	0.03	3001.4980	0.97
3	0.60	3093.5440	28.87	3093.5440	2.43	3093.5440	0.06	3093.5440	0.07	3093.5440	1.11	3093.5440	2.32	3093.5440	3.93	3093.5440	1.16	3043.5440	3600*	3085.9260	0.03	3085.9260	1.02
4	0.60	3178.2420	18.23	3178.2420	3.95	3178.2420	0.06	3178.2420	0.07	3178.2420	0.88	3178.2420	1.61	3178.2420	2.37	3178.2420	1.60	3178.2420	3600*	3170.3130	0.04	3170.3130	0.89
5	0.60	3100.0530	26.86	3099.9990	0.80	3099.9990	0.07	3099.9990	0.07	3099.9990	2.99	3099.9990	0.96	3099.9990	3.81	3099.9990	3.54	3031.9990	3600*	3092.3580	0.03	3092.3580	0.80
1	0.65	3236.9310	16.60	3236.9040	0.82	3236.9040	0.06	3222.9040	0.07	3236.9040	2.57	3236.9040	1.52	3236.9040	3.76	3236.9040	1.23	3138.9040	3600*	3228.9170	0.03	3228.9170	0.81
2	0.65	3067.4630	11.80	3067.4360	3.17	3067.4360	0.07	3067.4360	0.07	3067.4360	1.56	3067.4360	1.31	3067.4360	3.76	3067.4360	1.74	3023.4360	3600*	3060.0920	0.04	3060.0920	0.75
3	0.65	3145.6835	27.21	3145.6565	0.83	3145.6565	0.06	3145.6565	0.06	3145.6565	2.02	3145.6565	2.23	3145.6565	2.56	3145.6565	2.14	3101.6565	3600*	3137.9710	0.04	3137.9710	0.89
4	0.65	3223.8680	14.40	3223.8410	0.79	3223.8410	0.06	3223.8410	0.07	3223.8410	2.13	3223.8410	1.95	3223.8410	1.50	3223.8410	1.53	3179.8410	3600*	3215.8760	0.04	3215.8760	0.66
5	0.65	3145.6655	16.76	3145.6115	2.97	3145.6115	0.06	3145.6115	0.08	3145.6115	0.86	3145.6115	1.05	3145.6115	2.91	3145.6115	4.68	3089.6115	3600*	3137.9390	0.03	3137.9390	0.78
1	0.70	3282.5570	26.20	3282.5030	1.44	3282.5030	0.06	3282.5030	0.07	3282.5030	2.74	3282.5030	1.94	3282.5030	4.17	3282.5030	2.35	3190.5030	3600*	3274.4300	0.03	3274.4300	0.72
2	0.70	3126.1160	13.79	3126.0620	2.13	3094.0620	0.06	3126.0620	0.07	3100.0620	1.98	3126.0620	1.63	3126.0620	5.42	3126.0620	5.36	3088.0620	3600*	3118.5650	0.03	3118.5650	0.73
3	0.70	3210.8725	22.59	3210.8185	4.42	3210.8185	0.07	3208.8455	0.07	3210.8185	1.17	3210.8185	3.22	3210.8185	4.47	3210.8185	3.29	3172.8185	3600*	3203.0110	0.04	3203.0110	0.81
4	0.70	3275.9670	18.68	3275.9670	3.45	3275.9670	0.06	3275.9670	0.06	3275.9670	2.18	3275.9670	1.81	3275.9670	3.12	3275.9670	1.33	3195.9670	3600*	3267.9300	0.04	3267.9300	0.55
5	0.70	3210.8005	27.18	3210.7735	1.96	3208.8005	0.06	3210.7735	0.07	3210.7735	0.91	3210.7735	2.02	3210.7735	4.67	3210.7735	1.14	3112.7735	3600*	3202.9390	0.03	3202.9390	0.73
1	0.75	3354.1785	20.48	3354.1785	1.78	3354.1785	0.06	3354.1785	0.07	3354.1785	0.91	3354.1785	0.79	3354.1785	2.56	3354.1785	4.80	3322.1785	3600*	3345.9210	0.03	3345.9210	0.72
2	0.75	3210.8680	33.16	3210.7870	5.56	3210.7870	0.07	3210.7870	0.06	3196.7870	1.07	3210.7870	0.82	3210.7870	3.72	3210.7870	1.35	3184.7870	3600*	3203.0520	0.03	3203.0520	0.73
3	0.75	3263.0255	22.80	3262.9715	2.32	3262.9715	0.06	3262.9715	0.06	3262.9715	1.06	3262.9715	2.31	3262.9715	3.46	3262.9715	1.94	3224.9715	3600*	3255.0560	0.03	3255.0560	0.78
4	0.75	3347.6740	18.54	3347.6470	0.74	3315.6470	0.06	3339.6740	0.06	3347.6470	2.80	3347.6470	2.27	3347.6470	3.65	3347.6470	1.07	3315.6470	3600*	3339.4120	0.03	3339.4120	0.76
5	0.75	3282.4580	30.17	3282.4310	4.14	3282.4310	0.08	3282.4310	0.06	3282.4310	4.84	3282.4310	2.80	3282.4310	3.64	3282.4310	4.38	3244.4310	3600*	3274.4210	0.03	3274.4210	0.67
1	0.80	3425.8540	20.02	3425.8540	1.58	3425.8540	0.06	3425.8540	0.06	3425.8540	2.12	3425.8540	1.41	3425.8540	3.35	3425.8540	2.89	3393.8540	3600*	3417.3980	0.03	3417.3980	0.81
2	0.80	3295.5435	16.90	3295.5165	0.68	3295.5165	0.07	3257.5165	0.06	3295.5165	0.92	3295.5165	1.00	3293.5165	2.72	3295.5165	1.99	3275.5165	3600*	3287.5290	0.03	3287.5290	0.77
3	0.80	3334.6875	14.61	3334.6605	0.53	3332.6605	0.07	3334.6605	0.06	3334.6605	0.83	3326.6605	0.87	3334.6605	3.07	3334.6605	3.30	3314.6605	3600*	3326.5430	0.04	3326.5430	0.88
4	0.80	3406.3450	28.16	3406.3450	6.05	3406.2910	0.08	3406.2910	0.06	3406.2910	1.51	3406.2910	0.65	3406.2910	2.22	3406.2910	1.29	3386.2910	3600*	3397.9340	0.04	3397.9340	0.95
5	0.80	3347.6245	23.84	3347.5705	1.46	3347.5705	0.06	3347.5705	0.06	3347.5705	0.96	3347.5705	1.04	3347.5705	5.02	3347.5705	2.75	3309.5705	3600*	3339.3810	0.03	3339.3810	0.69
1	0.85	3530.1420	18.31	3530.1150	1.47	3530.1150	0.06	3530.1150	0.06	3530.1150	1.05	3530.1150	1.38	3530.1150	2.73	3530.1150	2.08	3510.1150	3600*	3521.3310	0.03	3521.3310	0.80
2	0.85	3393.2685	17.36	3393.2415	1.88	3393.2415	0.06	3393.2415	0.08	3393.2415	4.60	3393.2415	3.10	3393.2415	3.53	3393.2415	1.79	3373.2415	3600*	3384.9660	0.03	3384.9660	0.71
3	0.85	3399.8945	33.16	3399.8405	2.35	3399.8405	0.07	3399.8405	0.07	3399.8405	0.76	3399.8405	0.81	3399.8405	1.96	3399.8405	1.49	3385.8405	3600*	3391.5520	0.03	3391.5520	0.84
4	0.85	3490.9935	32.10	3490.9935	0.76	3490.9935	0.06	3490.9935	0.07	3490.9935	2.12	3490.9935	1.66	3490.9935	2.17	3490.9935	1.58	3482.9935	3600*	3482.4030	0.02	3482.4030	0.88
5	0.85	3412.7955	25.85	3412.7415	1.41	3386.7415	0.08	3412.7415	0.06	3412.7415	1.77	3412.7415	0.91	3412.7415	2.89	3412.7415	3.59	3386.7415	3600*	3404.4030	0.03	3404.4030	0.81
1	0.90	3614.8355	13.44	3614.8085	2.14	3614.8085	0.06	3614.8085	0.07	3614.8085	0.92	3614.8085	2.60	3614.8085	4.58	3614.8085	2.13	3606.8085	3600*	3605.8130	0.03	3605.8130	0.91
2	0.90	3491.0070	16.15	3490.9530	0.74	3476.9530	0.07	3488.9800	0.06	3490.9530	0.94	3490.9530	1.10	3490.9530	2.75	3490.9530	2.40	3476.9530	3600*	3482.3850	0.04	3482.3850	0.82
3	0.90	3510.6330	22.20	3510.5790	1.39	3510.5790	0.06	3510.5790	0.07	3510.5790	2.46	3510.5790	1.48	3510.5790	1.56	3508.5790	1.55	3508.5790	3600*	3501.9840	0.03	3501.9840	0.81
4	0.90	3634.3850	15.73	3634.3310	2.24	3634.3310	0.06	3634.3310	0.06	3634.3310	2.93	3634.3310	2.35	3634.3310	2.45	3634.3310	1.29	3632.3310	3600*	3625.2950	0.02	3625.2950	1.02
5	0.90	3503.9845	17.71	3503.9575	0.78	3503.9575	0.08	3465.9575	0.06	3503.9575	0.87	3503.9575	1.77	3503.9575	2.50	3503.9575	1.29	3489.9575	3600*	3495.3940	0.03	3495.3940	1.03
1	0.95	3777.7720	11.82	3777.7180	1.94	3777.7180	0.06	3777.7180	0.07	3777.7180	0.86	3775.7180	1.12	3777.7180	1.59	3777.7180	1.21	3777.7180	3600*	3768.2320	0.03	3768.2320	1.17
2	0.95	3686.5065	10.86	3686.4525	0.86	3686.4525	0.06	3686.4525	0.06	3686.4525	1.38	3686.4525	1.91	3686.4525	1.27	3686.4525	1.54	3684.4525	3600*	3677.2320	0.03	3677.2320	0.90
3	0.95	3647.4075	34.22	3647.3805	0.72	3647.3805	0.08	3647.3805	0.06	3647.3805	1.75	3647.3805	1.51	3647.3805	1.02	3647.3805	1.36	3647.3805	3600*	3638.4250	0.04	3638.4250	0.96
4	0.95	3849.3620	23.95	3849.3350	0.81	3849.3350	0.06	3849.3350	0.07	3849.3350	1.29	3849.3350	2.34	3849.3350	2.06	3849.3350	2.12	3849.3350	3600*	3839.6380	0.03	3839.6380	1.06
5	0.95	3686.3580	27.24	3686.3580	1.44	3686.3580	0.07	3666.3580	0.06	3686.3580	2.78	3686.3580	0.85	3686.3580	3.02	3686.3580	3.40	3546.3580	3600*	3677.2680	0.03	3677.2680	1.13

C.2 Class Two

Table 12 Results for Class Two with $\hat{d} = 2$.

Instance	α	SM		SM_{cut}		FA		FA_{cut}		FASM		$FASM_{cut}$		SMFA		$SMFA_{cut}$	
		LB	Time	LB	Time	LB	Time	LB	Time	LB	Time	LB	Time	LB	Time	LB	Time
1	0.60	7267.3480	3,600*	7916.5475	2058.92	7862.0270	1.35	7873.0260	2.87	7911.1070	710.75	7917.6005	3684.51	7911.4155	1298.38	7912.1060	781.86
2	0.60	6956.1380	3,600*	7688.2445	1433.67	7641.9110	1.83	7647.2070	1.94	7684.1270	378.22	7690.5530	3021.92	7687.9475	2740.84	7689.3245	2543.55
3	0.60	7325.9195	3,600*	7954.2615	3601.16	7908.0225	1.68	7859.0080	2.28	7952.7205	931.24	7951.6830	568.36	7920.3445	1442.14	7951.4130	688.00
4	0.60	7280.9015	3,600*	8051.5310	1486.01	7987.9805	1.41	8021.5890	2.50	8001.7085	395.36	8030.1155	53.81	7998.4665	462.70	8050.7750	2024.77
5	0.60	7073.2625	3,600*	7805.3470	1035.70	7764.9555	2.40	7776.6295	1.62	7808.7915	573.01	7801.0210	317.87	7757.6035	565.90	7808.2920	747.46
1	0.65	7540.7560	3,600*	8013.0785	1815.64	7964.8510	1.31	8000.8220	1.73	8021.2675	766.10	8021.7535	496.57	8019.0110	1584.59	8022.3745	1248.43
2	0.65	7377.5765	3,600*	7798.5770	646.23	7783.1170	1.60	7811.4005	1.38	7842.0040	3548.13	7839.9520	682.47	7801.3445	1418.32	7805.5355	350.07
3	0.65	7502.4905	3,600*	8039.1030	847.90	7987.5215	1.49	8023.2900	1.43	8061.2705	3,600*	8059.4885	1533.99	8058.3950	3613.92	8056.7075	1437.79
4	0.65	7600.1795	3,600*	8148.6885	2890.53	8097.3675	1.71	8120.5150	1.80	8151.6180	2629.44	8147.8650	714.19	8148.3510	2462.60	8147.4735	1064.22
5	0.65	7377.8105	3,600*	7912.5965	2442.18	7856.1125	1.46	7879.3140	1.61	7912.0295	929.14	7909.9080	337.20	7915.7825	2688.74	7893.5165	661.25
1	0.70	7814.1655	3,600*	8116.2320	2001.57	8040.6545	1.32	8037.0480	1.70	8111.5610	1043.03	8118.7295	3,600*	8039.6130	992.59	8113.0460	993.76
2	0.70	7540.5445	3,600*	7925.8335	1793.96	7848.9040	1.53	7847.6215	1.51	7899.0120	1110.75	7923.6600	688.59	7871.5885	2258.02	7900.2000	1429.84
3	0.70	7827.5075	3,600*	8195.8005	972.12	8118.6915	1.22	8182.7860	1.48	8191.4590	703.47	8205.2505	2446.51	8176.4985	1861.02	8202.2535	2984.55
4	0.70	7983.3850	3,600*	8284.8715	2924.26	8201.8765	1.12	8254.7000	1.46	8282.3605	655.05	8282.6440	592.86	8255.4425	1274.56	8283.2110	2070.38
5	0.70	7835.5600	3,600*	8051.1335	2557.83	7974.6110	1.27	7978.7595	1.83	8049.5540	1480.82	8048.2985	471.43	8051.0390	2972.67	8050.5800	2412.50
1	0.75	7958.3580	3,600*	8234.7475	3261.37	8127.5635	1.19	8136.5375	2.17	8200.9040	487.07	8228.4295	344.60	8230.9675	1393.58	8186.0725	2797.50
2	0.75	7703.1210	3,600*	8076.3125	3039.99	7987.1555	1.40	7979.8575	1.56	8046.0620	920.85	8075.9885	2486.15	8047.6010	1235.10	8073.3965	2004.15
3	0.75	8114.7575	3,600*	8314.7415	3282.65	8217.6000	1.54	8220.6685	1.65	8308.0725	1212.08	8315.0925	3,600*	8243.4340	1464.71	8313.4185	3376.02
4	0.75	8224.9105	3,600*	8415.0505	3,600*	8332.3120	1.18	8266.0535	1.47	8377.9300	1388.26	8411.3110	789.04	8349.3920	806.24	8390.1730	1431.63
5	0.75	7895.7460	3,600*	8209.3380	1355.35	8104.5280	1.55	8128.0690	1.66	8153.2725	455.23	8201.7450	416.81	8129.5270	1122.04	8211.6870	1526.73
1	0.80	8173.2535	3,600*	8370.2525	2098.84	8286.9950	1.65	8368.6055	1.90	8370.7925	440.13	8370.2660	564.81	8345.5910	1108.58	8370.9545	994.04
2	0.80	7989.4355	3,600*	8224.4570	902.49	8130.2120	1.32	8229.8630	1.54	8209.0490	415.52	8239.0895	762.24	8185.3055	1850.82	8180.9950	1419.58
3	0.80	8197.0965	3,600*	8435.3277	967.09	8351.2980	1.11	8428.7580	1.97	8400.9180	572.87	8437.5529	760.82	8437.6605	2025.71	8405.4195	444.62
4	0.80	8328.3380	3,600*	8530.4585	1064.40	8475.7970	1.63	8551.1490	1.64	8530.8480	726.58	8555.3090	490.68	8555.3495	1573.80	8555.8085	1508.28
5	0.80	8225.5645	3,600*	8334.9115	3073.15	8248.4005	1.17	8332.5895	1.51	8297.0060	927.25	8229.3880	53.67	8308.4950	1288.07	8282.0185	3088.01
1	0.85	8400.6040	3,600*	8462.7967	746.46	8436.3815	1.42	8512.3565	2.69	8532.8885	3339.82	8488.0305	680.78	8497.5290	1379.86	8525.7335	866.58
2	0.85	8411.4910	3,600*	8434.0195	1388.12	8346.1120	1.25	8428.5595	1.53	8435.6530	653.53	8439.4060	1332.05	8395.5410	2529.69	8436.0850	729.18
3	0.85	8421.3150	3,600*	8613.8925	1813.11	8490.8010	1.59	8532.0855	1.88	8580.5445	549.63	8580.3940	1093.58	8584.1665	1030.62	8584.5000	1493.54
4	0.85	8575.9855	3,600*	8730.7805	2270.60	8638.4585	1.12	8638.2405	1.55	8729.5250	453.29	8740.9325	3,600*	8735.6810	3191.33	8730.1055	1276.66
5	0.85	8455.5670	3,600*	8474.9130	1845.57	8384.1570	1.31	8471.9835	1.93	8474.4930	688.93	8477.4645	1804.06	8406.6160	1111.42	8473.2255	761.33
1	0.90	8669.8140	3,600*	8699.7850	2459.54	8610.8805	1.53	8688.1650	1.58	8699.0985	547.67	8700.4870	1390.98	8675.1870	2839.46	8703.6325	3311.32
2	0.90	8705.7715	3,600*	8753.0320	2123.85	8664.0600	1.62	8743.8845	1.52	8754.1950	1054.18	8697.8845	727.29	8701.9945	576.05	8753.7900	899.70
3	0.90	8789.5875	3,600*	8868.7830	3455.78	8737.6685	1.44	8731.2635	1.50	8863.2075	1165.07	8781.5085	271.35	8760.8486	1061.69	8860.1970	635.28
4	0.90	8873.8580	3,600*	8894.0315	1504.55	8827.5590	1.35	8910.2090	1.72	8919.2235	512.06	8922.6815	542.76	8923.5165	1880.70	8922.1125	1342.83
5	0.90	8623.4750	3,600*	8636.5840	1301.74	8571.1780	1.10	8639.3130	1.48	8652.9350	514.07	8593.2320	637.66	8605.4635	885.31	8636.6110	646.80
1	0.95	9052.3510	3286.68	9052.7560	876.81	8991.0010	1.63	9000.4455	1.23	8989.1920	354.73	9052.8370	497.45	9057.5620	2213.79	9050.9200	427.01
2	0.95	9083.0690	3410.62	9075.1365	585.25	8977.1210	1.34	8983.2445	1.47	9084.1490	540.71	9062.0005	573.22	9049.3390	730.14	9082.8665	658.89
3	0.95	9169.5850	3575.90	9152.3180	336.50	9083.8435	1.58	9073.6545	1.51	9172.8925	1243.71	9169.8415	831.35	9129.1415	825.43	9168.0077	576.18
4	0.95	9295.3935	3,600*	9292.6260	654.64	9206.1650	1.34	9199.1930	1.09	9292.9770	1007.36	9295.6770	385.68	9292.6645	1447.57	9284.2730	1220.90
5	0.95	8967.3290	3,600*	8965.6550	657.06	8884.5460	1.62	8855.7205	1.28	8966.3860	888.74	8967.4930	710.82	8933.2345	1265.17	8969.6780	1200.09

C.3 Class Three

Table 13 Results for Class Three with $\hat{d} = 2$.

Instance	α	SM_{cut}		FA		FA_{cut}		FASM		$FASM_{cut}$		SMFA		$SMFA_{cut}$	
		LB	Time	LB	Time	LB	Time	LB	Time	LB	Time	LB	Time	LB	Time
1	0.60	15676.6680	3,600*	15544.1740	10.60	15682.5450	11.35	15688.6060	1921.92	15641.8720	3,600*	15560.2180	3,600*	15685.7860	3,600*
2	0.60	15260.0940	3,600*	15157.0870	11.66	15212.4910	11.50	15276.7810	1800.94	15216.4160	2990.07	15206.3370	3,600*	15245.4010	3,600*
3	0.60	15759.6540	3,600*	15623.3220	11.30	15739.0950	14.87	15824.2670	3,600*	15753.9580	3,600*	15633.4470	3,600*	15755.5550	3,600*
4	0.60	15995.8910	3,600*	15866.3960	10.70	15987.5210	14.97	16054.7740	3123.58	16006.9690	2679.63	15929.9090	3,600*	15993.3600	3,600*
5	0.60	15444.1780	3,600*	15328.9550	11.63	15431.5260	13.30	15478.6220	1640.82	15407.4630	2211.78	15345.7280	2404.98	15437.3860	3,600*
1	0.65	15766.5370	3,600*	15642.7270	10.70	15700.9900	11.50	15824.4390	3,600*	15748.8660	3449.91	15568.2220	3,600*	15722.4110	3,600*
2	0.65	15437.1190	3,600*	15322.3910	13.18	15474.1250	12.24	15480.3190	3,600*	15453.4750	3,600*	15271.6520	3,600*	15381.0380	3,600*
3	0.65	15896.3200	3,600*	15784.5460	9.96	15853.5430	10.48	15950.0760	3318.27	15837.8030	677.77	15692.6800	3,600*	15872.5090	3,600*
4	0.65	16162.1640	3,600*	16041.1300	10.41	16131.2760	10.24	16224.5630	3,600*	16172.4770	3,600*	16009.4530	3,600*	16109.5250	3,600*
5	0.65	15577.8040	3,600*	15448.2280	13.31	15525.8270	16.05	15616.7590	2186.56	15572.6300	3,600*	15416.1820	3,600*	15516.0470	3,600*
1	0.70	15973.8450	3,600*	15795.8820	8.89	15986.5770	8.67	15982.5570	3,600*	15956.8670	1429.66	15870.6660	3,600*	15907.9890	3,600*
2	0.70	15646.3610	3,600*	15526.6680	11.24	15585.0030	9.62	15691.1700	3272.46	15591.2230	1461.09	15409.4430	3,600*	15593.2900	3,600*
3	0.70	16102.4760	3,600*	15995.5030	14.86	16059.9100	16.17	16146.8350	1274.32	16103.1650	3,600*	15926.8000	3,600*	16066.7370	3,600*
4	0.70	16266.3980	2864.71	16146.9660	10.05	16229.2200	10.51	16331.2790	3392.56	16232.7520	3,600*	16072.2000	3,600*	16213.7680	3,600*
5	0.70	15731.9060	3,600*	15636.6270	10.80	15685.1400	13.42	15806.9400	3,600*	15743.5310	3,600*	15591.6900	3,600*	15703.8340	3,600*
1	0.75	16054.9810	3,600*	15960.5840	9.29	15960.3950	12.50	16105.6910	3,600*	15943.8030	1024.43	15812.6030	3,600*	16030.5760	3,600*
2	0.75	15862.4600	3,600*	15754.1330	11.70	15769.3970	9.47	15897.0620	3,600*	15830.1550	3,600*	15616.5590	3,600*	15794.4870	3,600*
3	0.75	16330.7940	3,600*	16219.4590	9.59	16266.7840	10.63	16385.0540	3,600*	16319.8250	3,600*	16118.9020	3,600*	16294.6140	3,600*
4	0.75	16449.4510	3,600*	16355.6120	9.33	16416.4130	10.26	16490.5120	1333.79	16473.4810	3,600*	16267.4780	3,600*	16417.5860	3,600*
5	0.75	15846.0630	3,600*	15766.3410	9.31	15781.3910	13.48	15922.6290	3,600*	15789.4540	3,600*	15629.6670	3,600*	15797.9550	3,600*
1	0.80	16327.6630	3,600*	16221.3380	12.45	16337.2140	11.63	16343.9140	3,600*	16303.6030	3,600*	16095.9530	3,600*	16255.2940	3,600*
2	0.80	16074.5480	3,600*	15986.3220	12.28	16016.2590	7.46	16102.3560	3,600*	16008.9250	3,600*	15843.9450	3,600*	16085.5220	3,600*
3	0.80	16499.5360	3,600*	16417.3370	9.61	16428.9680	12.28	16559.9310	3,600*	16443.3730	2735.08	16269.1130	3,600*	16449.0680	3,600*
4	0.80	16650.5760	3578.21	16549.4840	14.06	16572.9020	8.83	16702.2670	3,600*	16556.8560	3,600*	16406.3240	3,600*	16581.9460	3,600*
5	0.80	16079.6200	3,600*	15985.3580	8.47	15960.7210	8.67	16103.0860	3,600*	15917.3130	749.02	15872.9090	3,600*	16016.9090	3,600*
1	0.85	16453.5630	3,600*	16372.1290	8.94	16357.6970	8.68	16487.8010	1738.24	16368.5560	3,600*	16258.8340	3,600*	16400.9510	3,600*
2	0.85	16246.0280	3,600*	16171.8560	9.54	16174.4880	8.50	16288.1420	2851.41	16159.3990	3,600*	16050.7230	3,600*	16221.0920	3,600*
3	0.85	16782.2000	3,600*	16697.9740	11.56	16687.9990	8.44	16792.1430	1217.71	16700.1560	2092.14	16561.0750	3,600*	16725.0570	3,600*
4	0.85	16893.2080	2497.15	16803.2250	8.73	16809.9720	10.50	16933.3600	3,600*	16845.4830	3,600*	16659.7050	3,600*	16839.8310	3,600*
5	0.85	16256.9200	3,600*	16157.7650	8.53	16148.5320	9.19	16287.0070	3,600*	16176.1580	3,600*	15995.9570	3,600*	16204.1280	3,600*
1	0.90	16771.4290	3,600*	16703.2280	10.63	16721.8080	7.95	16846.2630	3124.44	16733.8190	3,600*	16551.3770	3,600*	16782.7540	3,600*
2	0.90	16521.9270	3,600*	16443.1300	10.68	16459.8990	9.24	16571.9580	1819.29	16471.7390	3,600*	16355.4950	3,600*	16528.0659	3,600*
3	0.90	17273.6125	3,600*	17214.0680	8.50	17220.3490	7.90	17284.9500	1410.60	17206.0160	3,600*	17081.5040	3,600*	17259.5813	3,600*
4	0.90	17151.7420	3,600*	17118.4950	8.06	17118.9540	8.04	17199.0890	1295.15	17074.5870	563.97	16984.1400	3,600*	17185.6610	3,600*
5	0.90	16397.4630	3,600*	16324.2520	13.49	16313.6370	10.06	16419.0904	598.66	16272.1660	1351.37	16199.5010	3,600*	16402.2300	3,600*
1	0.95	17300.8730	3,600*	17220.7400	7.29	17243.5760	6.96	17326.8570	1963.63	17194.5620	1842.58	17071.8380	3,600*	17314.5400	3,600*
2	0.95	17121.1670	3,600*	17091.1350	8.66	17084.4780	9.92	17169.0430	3047.38	17054.6640	2530.02	16926.3030	3,600*	17115.8600	3,600*
3	0.95	17850.6070	3,600*	17772.4850	7.58	17747.3620	7.72	17825.8800	594.89	17738.5660	3,600*	17649.7700	3,600*	17755.6330	3,600*
4	0.95	17773.5530	3,600*	17684.8710	7.54	17729.8090	7.87	17808.8530	3,600*	17612.6250	1013.60	17578.0320	3,600*	17773.0040	3,600*
5	0.95	16899.8850	3,600*	16896.9150	6.98	16926.6110	7.71	16977.2890	1879.92	16823.6180	3,600*	16725.4080	3,600*	16957.0820	3,600*