Fulfillment-oriented Location-Inventory Problem with Demand Information Uncertainty Based on Copositve Programming

Abstract

Motivated by a cooperation project with Group G, we put forward a fulfillment-oriented location-inventory problem under demand information uncertainty where only mean and covariance matrix are known. We formulate this problem as a mixed-integer two-stage distributionally robust optimization model, which is generally NP-hard. The model is further reformulated into a mixed-integer conic problem based on copositive cones. We design an interpretable dual-variable-based heuristic to accelerate the branch and bound solving process. Extensive numerical studies demonstrate that our proposed model can achieve a higher fulfillment rate and a lower total cost than a model utilizing empirical distributions derived from limited historical data points and a robust model ignoring the existence of covariance.

Keywords: Location-inventory problem, Copositive programming, Distributionally robust optimization

1. Introduction

Location-inventory problems refer to an elementary and classical stream of problems in a wide range of applications, including supply chain management, disaster relief, last-mile delivery, and retailing. For example, in supply chain management, Daskin et al. (2002) summarized that a typical location-inventory problem assumes the predetermined location of suppliers and aims to determine the optimal number and the location of distribution centers (DCs), allocate customers to DCs, and optimize the inventory service level at each DC. The location decision is a strategic decision, typically made for a planning horizon of two to five years. And the inventory decision is an operational planning decision that is highly influenced by strategic location decisions (Farahani et al. (2013)). In practice, designing a proper warehouse network by choosing appropriate locations and prepositioning a reasonable amount of items can cause significant positive economic effects such as decreased logistic cost, increased revenue, sales, and profit. The intrinsic characteristic of location-inventory problems, which combines strategic supply chain decision-making and operational inventory management decisions, makes itself attractive to the academic community.

Although the location-inventory problem is among the earliest and best-studied problem in the literature, with the rapid development of last-mile logistics and e-commerce, it has received much attention from practitioners in recent years in the field of retailing. The retailing industry has undergone significant reforms in recent years. Online retailing companies managed to provide customers with a faster and high-level service by deploying their relatively small DCs in cities, making the delivery time drop precipitously. For instance, in China, Freshippo, a supermarket owned by Alibaba Group, equipped its brick-and-mortar shops with packing materials, serving as DCs for online orders from customers within a three-kilometer radius. And in the U.S., Walmart adopted a similar strategy called “dark store” to quickly fulfill its online orders.

Our study is motivated by the cooperation with Group G, a Chinese leading offline retailing company selling baby athletic apparels of a top brand. Let us take Shanghai as an example. It has more than 40 stores in Shanghai, China, and sells about two thousand SKUs (stock keeping units) per season, including shoes, apparels, and accessories. Before 2018, inventory decisions are centrally made every week by the sales department, and a regional warehouse is responsible for executing these decisions. Although professional enough, sales experts still make suboptimal decisions occasionally, leading to stock shortages within that week and further losing sales opportunities. To avoid stock shortages, capture more sales opportunities, and provide the ability of quick response, Group G attempted to reconstruct their supply chains by building in-city small-size Upfront Warehouses (UWs) in Shanghai. Supported by a third-party express company, in-city UWs can deliver products to stores within four hours, which can be regarded as a coverage radius.

Our study’s main aims are choosing in-city UW locations from a predetermined set of potential locations and making decisions on weekly inventory levels so that the UWs can fulfill as much demand as possible within that week. Establishing a new UW needs a location-specified setup cost, and deploying one unit product requires a location-specified holding cost. These two decisions are referred to as first-stage decisions. Then, during the week, every replenishment request from stores will be fully fulfilled as long as there is enough capacity in the UWs. Any unmet demand will be counted as one unit penalty cost. Although replenishment requests come sequentially in practice, it is still reasonable to regard the successively arrived demand of each store within one week as an aggregated demand. Therefore, the second stage problem is a resource allocation problem that minimizes total unmet demand, given the deployed capacity and realized demand. That emphasizes the feature of fulfillment-orientation in our problem. For simplicity of notations, we temporarily simplify the realistic problem, and we only consider one selling week and one product, which can be easily extended to a more realistic situation.

Unfortunately, even the problem has been simplified, it still hard to handle due to the following two issues:

- the complexity of combining strategic location planning and operational inventory management.
- uncertainties in unpredictable factors such as future demand.

While the first issue always involves binary decision variables, making the problem itself hard to solve efficiently, early location-inventory studies have proposed many heuristics to which we can refer. The second issue is more challenging to deal with. Since each selling season only lasts three or four months (about twelve weeks to sixteen weeks), the week-level historical sales data is very limited. It is impossible to accurately estimate the distribution of demand, let alone tackle the censored demand issue.

Traditionally, there have been two main approaches to handle uncertainty in optimization problems, stochastic programming, and robust optimization. In stochastic programming, the uncertain parameters are represented as random variables having a known probability distribution, while in robust optimization, the probability distribution of uncertain parameters is usually ignored, and only the support set is taken into consideration. The former technique is not suitable for the case due to the scarcity of historical data, and the latter one is too conservative.

We adopt a distributionally robust optimization technique for our problem. We want to minimize the worst-case expected total cost over a set of feasible demand distributions characterized by first-moment and second-cross-moment information (i.e., mean value and covariance matrix), which is easy to estimate from a statistically large enough dataset, such as a dataset containing 30 samples. Utilizing the covariance matrix of demand offers us the possibility to capture correlations between different demand nodes. Please note that the existence of demand correlation is most the case in offline retailing. For example, stores in the same city are more likely to achieve lower sales volumes on a rainy day simultaneously. Although distributionally robust optimization relieves deficiencies of stochastic programming and robust optimization, computational intractability is another issue. To deal with this problem, we further equivalently reformulate our model into conic programming described by copositive cone constraints, and exploit semi-definite approximation to approximate copositive cones for computational convenience.

Our main contributions are three-fold. First, we introduced demand correlations to location-inventory problems with distribution uncertainty and established a mixed-integer two-stage distributionally robust optimization model to describe it. Second, we put forward a detailed proof over an equivalent reformulation between the original model and a copositive programming problem by exploiting min-cut max-flow theory in network problem. The equivalent reformulation does not exist in general. Third, we proposed a heuristic to accelerate the solving process, halving computational times.

The structure of this paper is organized as follows. We firstly give a literature review on location-inventory problems, distributionally robust optimization, and copositive programming in Section 2. We formally propose our studied model in Section 3, where we also put forward the reformulation model and an algorithm to accelerate the process of solving. In Section 4, we conduct numerous numerical studies to show the advantages of proposed models. In Section 5, several extensions are proposed to demonstrate our model's compatibility to more complex situations. And in Section 6, we conclude our study.

2. Literature Review

2.1. Location-Inventory Problem under Demand Uncertainty

The location-inventory problem originates from Baumol and Wolfe (1958), who considered a warehouse location model under a single-sourcing setting and proposed a heuristic to solve it. Later, numerous works considered more realistic but complex situations, such as inventory control (Miranda and Garrido (2004), Gabor and van Ommeren (2006)), service level requirement (Mak and Shen (2009), Li et al. (2017)), routing design (Shen and Qi (2007), Javid and Azad (2010)) , network design (Shu et al. (2010), Chou et al. (2011)), reverse logistics (Cardoso et al. (2013)). And the most attractive topic is dealing with uncertain demand, and most of the mentioned works considered demand uncertainty by assuming a known distribution or a set of scenarios. Although describing uncertain demand with known demand scenarios or known distributions is reasonable, it still has the risk of putting the decision maker in a disadvantageous situation of misspecification, which possibly results in an inappropriately designed network and causes massive economic loss.

To handle misspecification, distributionally robust optimization, an emerging modeling tool, is utilized to deal with the classical location-inventory related problem when historical data is scarce. For example, Nakao et al. (2017) formulated a distributionally robust network design by constructing an ambiguity set of the unknown demand distribution based on marginal moment information to minimize the worst-case total cost over all possible distributions. They approximated the objective function with a piece-wise linear function and solved the model with a tailored cutting-plane algorithm. Gourtani et al. (2020) formulated a mixed-integer programming problem: an optimal selection of facility locations in the first stage and an optimal decision on the operation of each facility in the second stage. They developed numerical schemes with the semi-infinite programming approach and approximate semi-infinite constraints with linear decision rules. Saif and Delage (2020) studied a distributionally robust facility location problem with a distributional ambiguity set defined by Wasserstein ball constructed from a small set of empirical sample. They also designed a column-generation algorithm to solve the problem exactly. The above three works indeed considered potential correlations between demand variables, whereas inventory level decisions were not explicitly considered.

Mak et al. (2013) considered electric vehicle charging stations and batteries deployment with a distributionally robust model, and mean and variance of demand are assumed to be known. They also studied the potential impacts of battery standardization and technology advancements on the optimal infrastructure deployment strategy. Zhang et al. (2020) studied a three-layer relief location-inventory problem with multiple objectives, maximizing the equitable distribution of relief supplies and simultaneously minimizing the transportation time and operation cost. Liu et al. (2019) designed an Emergency Medical Service system by optimizing locations, the number of ambulances, and demand assignment with a demand ambiguity set defined by first moment information. Although they considered both location and inventory decisions, the correlation between demand is omitted.

In our study, we explicitly incorporate correlations of demand into our location-inventory models by assuming the covariance matrix is known.

2.2. Moments-based Distributionally Robust Optimization

Moments-based distributionally robust optimization is a robust formulation for stochastic programming, where the objective function is formulated with respect to the worst-case expected penalty cost over the choice of a distribution subjected to an ambiguity set constrained by known support set and moments information. It was arguably first introduced by Scarf (1958), who studied the classical single-product newsvendor problem when only mean and variance of demand are known, and the support set includes all real numbers. The last decade witnessed an explosive growth in this topic after Bertsimas and Popescu (1999) connected moment problem and semidefinite optimization. For example, Bertsimas and Popescu (2005) derived a tight bound to the probability that an univariate random variable with first-k moments known belongs to a region defined by polynomial inequalities.Bertsimas et al. (2004) and Bertsimas et al. (2006) evaluated the expected optimal objective value of a 0-1 optimization problem with objective uncertainties. They showed that the optimal value could be obtained in polynomial time if the deterministic problem is solvable in polynomial time, and they applied it to a project management problem.

The above-mentioned works unanimously assume that only marginal moment information is known, ignoring correlations between random variables and further information such as covariance. Recently, many works incorporated cross-moment information. Popescu (2007) derived robust solutions to specific stochastic optimization problems based on mean-covariance information. They reduced the multivariate problem into a univariate mean-variance robust problem according to a general projection property and then characterized the worst-cast distribution as a point distribution. Mishra et al. (2012) calculated the upper bound on the worst-case maximum of n random variables through enumerating extreme points with semi-definite formulation. Bertsimas et al. (2010) further showed that generalizing this model to general linear programs leads to a NP-hard problem. Delage and Ye (2010) allowed the mean and covariance matrix to be unknown themselves. However, to reduce computational complexity incurred by involving semi-definite matrix, a linear decision rule based technique is introduced. For example, Chen et al. (2008) introduced a piecewise linear decision rule which is more flexible than regular decision rules when only mean, covariance, and support are known. Goh and Sim (2010) extended the set of primitive uncertainties for new linear decision rules of larger dimensions. Wiesemann et al. (2014) proposed a broad class of ambiguity sets with conic-representable expectation constraints. Bertsimas et al. (2019b) considered an adaptive distributionally robust linear optimization problem and equivalently reformulated it into a second-order cone problem when the ambiguity set is second-order-cone representable. Chen et al. (2019) modeled covariance information from an infinitely constrained perspective and further incorporated fourth moment information. Chen et al. (2020)

developed an algebraic modeling toolbox to automatically transform robust optimization problems to linear or second-order conic optimization problems based on linear decision rules.

However, in most practical problems, the support set should be restricted to nonnegative orthant. For example, demand should always be nonnegative, and service time in clinics should be longer than zero. And when the uncertainty is in objective function, Natarajan et al. (2011) showed that given mean and covariance matrix, the problem of computing the worst-case expect value can be reformulated as a conic problem based on completely positive cones. This equivalent reformulation sparks extensive applications based on completely positive cone or its dual cone, copositive program, which we will be further introduced in the next subsection. Our study heavily utilizes their proposed reformulations.

2.3. Copositive Programming

A n-dimensional Copositive Cone is defined as

$$\mathcal{CO}_n := \left\{ \boldsymbol{A} \in S_n \,\middle|\, \boldsymbol{v} \in \mathbb{R}_+^n, \boldsymbol{v}^T \boldsymbol{A} \boldsymbol{v} \geq 0 \right\}$$

where S_n denotes the cone of $n \times n$ symmetric matrics. For $\boldsymbol{A} \in \mathcal{CO}_n$, we write $\boldsymbol{A} \succeq_{co} \mathbf{0}$. A complelete positive cone is the dual cone of a copositive cone (Dickinson (2013)). It is defined as

$$\begin{aligned} \mathcal{CP}_n &:= \left\{ \boldsymbol{A} \in S_n \,\middle|\, \exists \boldsymbol{V} \in \mathbb{R}_+^{n \times m}, s.t. \boldsymbol{A} = \boldsymbol{V}\boldsymbol{V}^T \right\} \\ &:= \left\{ \boldsymbol{A} \in S_n \,\middle|\, \exists \boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_m \in \mathbb{R}_+^n, s.t. \boldsymbol{A} = \sum_{i=1}^{m} \boldsymbol{v}_i \boldsymbol{v}_i^T \right\} \end{aligned}$$

A programming problem with linear constraints on copositive cone or completely positive cone is called copositive program or completely positive program respectively. Recently, copositive program has attracted much attention in optimization community since its powerful modeling technique. Many NP-hard problems have been equivalently reformulated into copositive programs or completely positive programs. For example, the well-known maximum stable set problem has been proved to be equivalent to a copositive program (De Klerk and Pasechnik (2002)) by modifying the feasible region of theta number problem (Lovász and Schrijver (1991)) from a semidefinite cone to a copositive cone. Another example is the maximum weighted clique problem. It is first formulated as a standard quadratic optimization problem by Bomze (1998), and then Bomze et al. (2000) reformulated the quadratic optimization problem into a copositive problem. Dickinson (2013) further provided a new and direct proof for this reformulation. In general, Burer (2009) summarized that any nonconvex quadratic program having a mix of binary and continuous variables can be modeled as a linear program on copositive cone. Natarajan et al. (2011) further introduced copositive program into mixed 0-1 linear programs under objective uncertainty.

Although the original NP-hard problem could be equivalently reformulated into a copositive problem, the difficulty of optimization does not be reduced. Indeed, checking whether a given matrix lies in a copositive cone has been observed to be NP-complete (Murty and Kabadi (1985)). However, such reformulations shift the difficulty of optimization to understanding the mathematical properties of copositive cone. Once we have more knowledge about copositive cone, many optimization problems could be solved more efficiently. Natarajan et al. (2011) proved that with mild conditions, the completely positive program could be exactly solved by a semidefinite program. For more general cases, Lasserre (2011) proposed a hierarchy of approximation involving a sequence of semidefinite cones to solve parametric polynomial optimization problems. A commonly used outer approximation for completely positive cone is "Doubly Nonnegative Cone", defined as $\{\boldsymbol{A}\,|\,\boldsymbol{A} \succeq \boldsymbol{0}, \boldsymbol{A} \geq \boldsymbol{0}\}$. And $\{\boldsymbol{A}\,|\,\boldsymbol{A} = \boldsymbol{A}_1 + \boldsymbol{A}_2, \boldsymbol{A}_1 \succeq \boldsymbol{0}, \boldsymbol{A}_2 \geq \boldsymbol{0}\}$ provides an inner approximation to copositive cone. In our numerical study, we use this inner approximation to approximate copositive cone.

During recent years, researchers found that many operation management problems can be properly handled with copositive program and its approximation techniques. For example, Natarajan et al. (2011) applied copositive program on a project management problem to estimate the expected completion time and the persistence of each activity. Kong et al. (2013) handled a healthcare appointment-scheduling problem where only the mean and covariance of service durations are known. They formulated the problem as a convex copositive optimization problem with a tractable semidefinite relaxation. Li et al. (2014) studied the sequencing problem with random costs. Utilizing copositve program technique, Yan et al. (2018) designed a roving team deployment plan for Singapore Changi Airport. Kong et al. (2020) further incorporated patients' no-show behaviors into a healthcare appointment scheduling problem and reformulated the problem as a copositive program. We refer interested readers to Dickinson (2013) for general analytic properties of copositve cone, to Li et al. (2014) and Bomze (2012) for a comprehensive review over applications with copositive program.

3. Fulfillment-Oriented Network Design Problem

Consider a fulfillment-oriented warehouse network design problem with a set of potential supply nodes denoted by $\mathcal{W} = \{1, 2, \ldots, m\}$, and a set of demand nodes denoted by $\mathcal{R} = \{1, 2, \ldots, n\}$. The road links is denoted by $\mathcal{E}$, and its cardinality is $|\mathcal{E}| = r$. We denote $\Gamma(i)$ as the set of adjacent nodes for location i when the network is $\mathcal{E}$. Warehouse $i \in \mathcal{W}$ is able to deliver products to all adjacent demand nodes $\Gamma(i)$ through roads. Correspondingly, a demand node $j \in \mathcal{R}$ can receive products from its adjacent warehouses $\Gamma(j)$ when the network structure is $\mathcal{E}$. Suppose the decision maker has only limited information about the demand, i.e., only the first-moment, second-moment, and support set are known. Facing the uncertainty of future demand, the decision maker should select a subset of $\mathcal{W}$ to build large-enough warehouses and proactively deploy products accordingly so that after demand realized, the decision maker could allocate pre-deployed products to satisfy as much demand as possible (or equivalently as less unmet demand as possible). Each built warehouse is associated with a fixed setup cost $f_i, i \in [m]$, and each prepositioning unit product causes a holding cost $h_i, i \in [m]$. Since we focus on the fulfillment-oriented circumstance, each unmet demand will trigger one unit penalty cost. We want to minimize the total cost, including the first-stage setup cost and the worst expected second-stage penalty cost. We temporarily assume there is only one kind of product, and we will consider the multi-item situation in Section 5.1.

Notations. Throughout the paper, we use ***bold*** characters to denote vectors or matrixs. Corresponding normal characters denote its component-wise element. For example, $\boldsymbol{I}$ is the inventory vector, and I_i represents the inventory level at i-th warehouse. Variables with tilde symbol represent random variables, such as $\tilde{\boldsymbol{d}}$. We use $\boldsymbol{A} \succeq \mathbf{0}$ to show the matrix $\boldsymbol{A}$ is a semi-definite matrix. $\boldsymbol{A} \succeq_{co} \mathbf{0}$ and $\boldsymbol{A} \succeq_{cp} \mathbf{0}$ require $\boldsymbol{A}$ to be a copositive cone or completely positive cone. $\mathbb{R}^n_+$ denotes n-demiensional non-negative real number. $[n]$ means the set $\{1, 2, 3, \ldots, n\}$.

3.1. Model

The fulfillment-oriented location-inventory design (FOLI) problem under demand distribution uncertainty, given that the ambiguity set is characterized by first- and second-cross-moment, can be formulated as:

$$
\begin{aligned}
(FOLI) \quad \min_{\boldsymbol{I},\boldsymbol{Z}} \quad & \kappa(\boldsymbol{f}^T\boldsymbol{Z} + \boldsymbol{h}^T\boldsymbol{I}) + \sup_{\mathbb{P}\in\mathscr{F}(\boldsymbol{\mu},\boldsymbol{\Sigma})} \mathbb{E}_{\mathbb{P}}\left[g(\boldsymbol{I},\boldsymbol{Z},\tilde{\boldsymbol{d}})\right] \\
\text{s.t.} \quad & \boldsymbol{I} \leq U\boldsymbol{Z}, \\
& \boldsymbol{I} \in \mathbb{R}^m_+, \boldsymbol{Z} \in \{0,1\}^m
\end{aligned} \tag{1}
$$

where the distribution ambiguity set is:

$$
\mathscr{F}(\boldsymbol{\mu},\boldsymbol{\Sigma}) = \left\{ \mathbb{P} \in \mathcal{P}\left(\mathbb{R}^n_+\right) \;\middle|\; \begin{array}{l} \tilde{\boldsymbol{d}} \sim \mathbb{P} \\ \mathbb{E}_{\mathbb{P}}[\tilde{\mathbf{d}}] = \boldsymbol{\mu} \\ \mathbb{E}_{\mathbb{P}}[\tilde{\boldsymbol{d}}\tilde{\boldsymbol{d}}^T] = \boldsymbol{\Sigma} \\ \mathbb{P}[\tilde{\boldsymbol{d}} \in \mathbb{R}^n_+] = 1 \end{array} \right\}
$$

In Model (1), $\boldsymbol{f}$ is the setup cost of building a new warehouse, $\boldsymbol{h}$ is the holding cost for each unit product, and $\kappa \geq 0$ is a risk attitude parameter balancing costs of two stages. $\boldsymbol{I}$ is the decision variable, meaning the number of relief materials that should be deployed at each location. Another decision variable, $\boldsymbol{Z}$, represents whether a new warehouse should be built. The first constraint guarantees that materials can be stored at locations where a warehouse is established, where U is a positive large enough number. The remaining constraints are standard nonnegative and binary constraints. $g(\boldsymbol{I},\boldsymbol{Z},\tilde{\boldsymbol{d}})$ is the number of unmet demand obtained by solving an classical allocation problem after demand $\tilde{\boldsymbol{d}}$ is realized. And the allocation problem can be described as:

$$
g(\boldsymbol{I},\boldsymbol{Z},\tilde{\boldsymbol{d}}) = \min_{\boldsymbol{x}\in\Omega(\boldsymbol{I},\boldsymbol{Z})} \sum_{j\in[n]} \tilde{d}_j - \sum_{(i,j)\in\mathcal{E}(\boldsymbol{Z})} x_{ij} \tag{2}
$$

where $\Omega(\boldsymbol{I}, \boldsymbol{Z})$ is the feasible region characterized by

$$\Omega(\boldsymbol{I}, \boldsymbol{Z}) := \{\boldsymbol{x} \in \mathbb{R}_+^{|\mathcal{E}(\boldsymbol{Z})|} | \sum_{i:(i,j)\in\mathcal{E}(\boldsymbol{Z})} x_{ij} \leq \tilde{d}_j, \forall j \in [n]; \sum_{j:(i,j)\in\mathcal{E}(\boldsymbol{Z})} x_{ij} \leq I_i, \forall i \in [m]\}$$

$\mathcal{E}(\boldsymbol{Z})$ is the accessible roads based on decision $\boldsymbol{Z}$.

It is clear to see, Model (2) depends both on $\boldsymbol{I}$ and $\boldsymbol{Z}$, and accessible network is varied under different $\boldsymbol{Z}$, which hampers us from solving Model (1). Fortunately, in fact, (2) could get rid of the dependency on $\boldsymbol{Z}$ at the cost of lifting the dimension of feasible region from $|\mathcal{E}(\boldsymbol{Z})|$ to r.

Proposition 1. *Given a feasible solution* $(\boldsymbol{I}, \boldsymbol{Z})$ *to* (1)*, for any demand realization* $\tilde{\boldsymbol{d}} \in \mathbb{R}_+^n$*, the value of* (2) *is equal to the objective value of the following problem.*

$$g(\boldsymbol{I}, \tilde{\boldsymbol{d}}) = \min_{\boldsymbol{x}\in\Omega(\boldsymbol{I})} \sum_{j\in[n]} \tilde{d}_j - \sum_{(i,j)\in\mathcal{E}} x_{ij} \tag{3}$$

where

$$\Omega(\boldsymbol{I}) := \{\boldsymbol{x} \in \mathbb{R}_+^r | \sum_{i:(i,j)\in\mathcal{E}} x_{ij} \leq \tilde{d}_j, \forall j \in [n]; \sum_{j:(i,j)\in\mathcal{E}} x_{ij} \leq I_i, \forall i \in [m]\}$$

Proof. See Appendix A.1 □

According to Proposition 1, instead of considering both $(\boldsymbol{I}, \boldsymbol{Z})$, we only need to take $\boldsymbol{I}$ into consideration. Getting rid of $\boldsymbol{Z}$ does not harm the optimality, however, at the cost of boosting the dimension of recourse decision variables $\boldsymbol{x}$. Surprisingly, additional variables make further reformulation easier and provide more insights on selection locations for warehouses, accelerating the solving process. We will further explain this in Section 3.3.

From now on, we replace $g(\boldsymbol{I}, \boldsymbol{Z}, \tilde{\boldsymbol{d}})$ with $g(\boldsymbol{I}, \tilde{\boldsymbol{d}})$. Therefore, we have a reduced model (rFOLI) as follows:

$$\begin{aligned}
(rFOLI) \quad \min_{\boldsymbol{I},\boldsymbol{Z}} \quad & \kappa(\boldsymbol{f}^T\boldsymbol{Z} + \boldsymbol{h}^T\boldsymbol{I}) + \sup_{\mathbb{P}\in\mathscr{F}(\boldsymbol{\mu},\boldsymbol{\Sigma})} \mathbb{E}_{\mathbb{P}}\left[g(\boldsymbol{I}, \tilde{\boldsymbol{d}})\right] \\
\text{s.t.} \quad & \boldsymbol{I} \leq U\boldsymbol{Z}, \\
& \boldsymbol{I} \in \mathbb{R}_+^m, \boldsymbol{Z} \in \{0,1\}^m
\end{aligned} \tag{4}$$

With the more tractable recourse problem, we next equivalently reformulate Model (4) into a mixed-integer copositive programming problem.

3.2. Reformulation

Obviously, the difficulty of Model (1) comes from evaluating the worst expected second-stage unmet demand. Denote the worst-case expected second-stage problem as

$$L(\boldsymbol{I}) := \sup_{\mathbb{P}\in\mathscr{F}(\boldsymbol{\mu},\boldsymbol{\Sigma})} \mathbb{E}_{\mathbb{P}}\left[g(\boldsymbol{I}, \tilde{\boldsymbol{d}})\right] \tag{5}$$

which still only depends on $\boldsymbol{I}$ since the expectation operator does not invalidate Proposition 1. We first analyze the inner allocation problem, which has many properties help up simply the whole problem. Equivalently, $g(\boldsymbol{I}, \tilde{\boldsymbol{d}})$ can be rewritten as

$$
\begin{aligned}
g(\boldsymbol{I}, \tilde{\boldsymbol{d}}) = & \quad \min_{\boldsymbol{x} \in \Omega(\boldsymbol{I})} \sum_{j \in [n]} \tilde{d}_j - \sum_{(i,j) \in \mathcal{E}} x_{ij} \\
= & \quad \sum_{j \in [n]} \tilde{d}_j - \max_{\boldsymbol{x} \in \Omega(\boldsymbol{I})} \sum_{(i,j) \in \mathcal{E}} x_{ij} \\
= & \quad \sum_{j \in [n]} \tilde{d}_j - \min_{(\boldsymbol{u}, \boldsymbol{v}) \in \mathcal{L}} \tilde{\boldsymbol{d}}^T \boldsymbol{u} + \boldsymbol{I}^T \boldsymbol{v} \\
= & \quad - \min_{(\boldsymbol{u}, \boldsymbol{v}) \in \mathcal{L}} \tilde{\boldsymbol{d}}^T (\boldsymbol{u} - \mathbf{1}) + \boldsymbol{I}^T \boldsymbol{v} \\
= & \quad \max_{(\boldsymbol{u}, \boldsymbol{v}) \in \mathcal{L}} \tilde{\boldsymbol{d}}^T (\mathbf{1} - \boldsymbol{u}) - \boldsymbol{I}^T \boldsymbol{v}
\end{aligned}
$$

The first equation is exactly the minimizing unmet demand problem after $\tilde{\boldsymbol{d}}$ is realized. The second equation is by simply exchanging minimizing operator and negative symbol. The third equation strictly follows the classical min-cut max-flow theorem, where the feasible region is

$$
\mathcal{L} := \{(\boldsymbol{u}, \boldsymbol{v}) \in \{0,1\}^{n+m} | u_j + v_i \geq 1, \forall (i,j) \in \mathcal{E}\}
$$

The fourth equation holds by combining realized demand vector $\tilde{\boldsymbol{d}}$, where $\mathbf{1}$ is a vector with all elements equal to 1. And the last equation also comes from exchanging the maximizing operator and the negative symbol. One notable reformulation trick here is that, at the third equation, directly taking expectation on $\tilde{d}_j$ is doable if the real mean value is captured by the outer ambiguity set, which temporarily is the case we consider. However, We will further relax the ambiguity set to a more general case. And the reformulation we adopt here would be more compatible with further study. Additionally, from the perspective of stochastic programming, the feasible region of $\mathcal{L}$ is complete recourse, so that the region is always nonempty.

Due to the max-flow theorem, the feasible region $\mathcal{L}$ can be relaxed to a convex polyhedron without losing optimality by relaxing binary variables to continuous variables. The relaxed region is

$$
\mathcal{L}_{lp} := \{(\boldsymbol{u}, \boldsymbol{v}) \in [0,1]^{n+m} | u_j + v_i \geq 1, \forall (i,j) \in \mathcal{E}\}
$$

For further analytics, we modify $\mathcal{L}$ by replacing $\mathbf{1} - \boldsymbol{u}$ with decision variables $\hat{\boldsymbol{u}} = \mathbf{1} - \boldsymbol{u}$, and define the new region with $\hat{\boldsymbol{u}}$ as

$$
\hat{\mathcal{L}} := \{(\hat{\boldsymbol{u}}, \boldsymbol{v}) \in \{0,1\}^{n+m} | v_i \geq \hat{u}_j, \forall (i,j) \in \mathcal{E}\}
$$

Moreover, the linear relaxation of the feasible region is

$$
\hat{\mathcal{L}}_{lp} := \left\{ (\hat{\boldsymbol{u}}, \boldsymbol{v}) \in \mathbb{R}_+^{n+m} \;\middle|\; \begin{array}{l} v_i \geq \hat{u}_j, \forall (i,j) \in \mathcal{E} \\ \hat{\boldsymbol{u}} \leq \mathbf{1} \\ \boldsymbol{v} \leq \mathbf{1} \end{array} \right\}
$$

Following the classical min-cut max-flow theorem, the optimal solution of $g(\boldsymbol{I}, \tilde{\boldsymbol{d}})$ is the same no matter we optimize in region $\hat{\mathcal{L}}$ or $\hat{\mathcal{L}}_{lp}$. Therefore, $g(\boldsymbol{I}, \tilde{\boldsymbol{d}})$ has the same value, as well as the same optimal solution, among the following problems:

$$
\begin{aligned}
g(\boldsymbol{I}, \tilde{\boldsymbol{d}}) &= \max_{(\boldsymbol{u}, \boldsymbol{v}) \in \mathcal{L}} \tilde{\boldsymbol{d}}^T (\mathbf{1} - \boldsymbol{u}) - \boldsymbol{I}^T \boldsymbol{v} && (6)\\
&= \max_{(\hat{\boldsymbol{u}}, \boldsymbol{v}) \in \hat{\mathcal{L}}} \tilde{\boldsymbol{d}}^T \hat{\boldsymbol{u}} - \boldsymbol{I}^T \boldsymbol{v} && (7)\\
&= \max_{(\hat{\boldsymbol{u}}, \boldsymbol{v}) \in \hat{\mathcal{L}}_{lp} \cap \{0,1\}^{n+m}} \tilde{\boldsymbol{d}}^T \hat{\boldsymbol{u}} - \boldsymbol{I}^T \boldsymbol{v} && (8)\\
&= \max_{(\hat{\boldsymbol{u}}, \boldsymbol{v}) \in \hat{\mathcal{L}}_{lp}} \tilde{\boldsymbol{d}}^T \hat{\boldsymbol{u}} - \boldsymbol{I}^T \boldsymbol{v} && (9)
\end{aligned}
$$

Unfortunately, though we have relaxed the inner allocation problem from an integer programming problem to a linear programming problem with a fixed feasible region $\hat{\mathcal{L}}_{lp}$ and it is irrelative to first-stage decision $\boldsymbol{I}$, the difficulty of evaluating (5) does not reduce. Actually, Proposition 2 shows that problem (5), even with feasible region $\hat{\mathcal{L}}_{lp}$, is still a NP-hard problem.

Proposition 2. *Given $g(\boldsymbol{I}, \tilde{\boldsymbol{d}})$ is in the form of* (9)*, and the feasible region is $\hat{\mathcal{L}}_{lp}$, calculating the value of problem* (5) *is still a NP-hard problem.*

Proof. See Appendix A.2 □

While the problem is NP-hard, we can reformulate it into a conic programming problem and then solve it with positive semi-definite relaxation.

Through adding slackness variables $(\boldsymbol{s}^\dagger, \hat{\boldsymbol{u}}^\dagger, \boldsymbol{v}^\dagger) \in \mathbb{R}_+^{r+n+m}$, we can modify $\hat{\mathcal{L}}_{lp}$ to a system only consist of equalities:

$$
\hat{\mathcal{L}}_{lp}^\dagger := \left\{ (\hat{\boldsymbol{u}}, \boldsymbol{v}, \boldsymbol{s}, \hat{\boldsymbol{u}}^\dagger, \boldsymbol{v}^\dagger) \in \mathbb{R}_+^{2n+2m+r} \;\middle|\; \begin{array}{l} \hat{u}_j - v_i + s_{ij}^\dagger = 0, \forall (i,j) \in \mathcal{E} \\ \hat{\boldsymbol{u}} + \hat{\boldsymbol{u}}^\dagger = \mathbf{1} \\ \boldsymbol{v} + \boldsymbol{v}^\dagger = \mathbf{1} \end{array} \right\}
$$

Therefore, the expected second-stage penalty value under the worst distribution can be equivalently reformulated as:

$$
L(\boldsymbol{I}) = \sup_{\mathbb{P} \in \mathscr{F}(\boldsymbol{\mu}, \boldsymbol{\Sigma})} \mathbb{E}_{\mathbb{P}} \left[g(\boldsymbol{I}, \tilde{\boldsymbol{d}}) \right] = \sup_{\mathbb{P} \in \mathscr{F}(\boldsymbol{\mu}, \boldsymbol{\Sigma})} \mathbb{E}_{\mathbb{P}} \left[\max_{(\tilde{\boldsymbol{u}}, \boldsymbol{v}, \boldsymbol{s}, \hat{\boldsymbol{u}}^\dagger, \boldsymbol{v}^\dagger) \in \hat{\mathcal{L}}_{lp}^\dagger \cap \{0,1\}^{2n+2m+r}} \tilde{\boldsymbol{d}}^T \hat{\boldsymbol{u}} - \boldsymbol{I}^T \boldsymbol{v} \right]
$$

Please note that uncertainties of the inner problem only exist in the objective function. For simplicity of notations, we denote, with a little bit of abuse of notations, the decision variables by $\boldsymbol{x} := (\hat{\boldsymbol{u}}, \boldsymbol{v}, \boldsymbol{s}, \hat{\boldsymbol{u}}^\dagger, \boldsymbol{v}^\dagger) \in \hat{\mathcal{L}}_{lp}^\dagger \subset \mathbb{R}_+^N$, where $N = 2m + 2n + r$. The constraints of $\hat{\mathcal{L}}_{lp}^\dagger$ can also be written in a more general form as $\{\boldsymbol{x} \geq \mathbf{0} \mid \boldsymbol{a}_i^T \boldsymbol{x} = b_i, \forall i \in [M]\}$, where $M = r + n + m$. Interestingly, the first r constraints of $\hat{\mathcal{L}}_{lp}$ only involve roads in $\mathcal{E}$, and following n and m constraints only involve demand nodes and supply nodes, respectively.

Now, it is easy to demonstrate that several key assumptions in Natarajan et al. (2011) always hold in our (FOLI) problem, and we formally summarize these satisfied assumptions as follows:

1. $\boldsymbol{x} \in \hat{\mathcal{L}}_{lp}^{\dagger} := \left\{ \boldsymbol{x} \geq \mathbf{0} \,\middle|\, \boldsymbol{a}_i^T \boldsymbol{x} = \boldsymbol{b}_i, \forall i \in [M] \right\} \implies \boldsymbol{x} \leq \mathbf{1}$

2. The feasible region of $\hat{\mathcal{L}}_{lp}^{\dagger} \cap \{0,1\}^N$ is nonempty and bounded

Checking the definition of $\hat{\mathcal{L}}_{lp}^{\dagger}$ naturally results in these two conclusions. To reformulate the problem (rFOLI) into a conic programming problem based on copositive cone, another key assumption is necessary.

Assumption 1. The random coeffcent $\tilde{\boldsymbol{d}}$ has finite first-two moments $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. And the moment matrix $\begin{pmatrix} 1 & \boldsymbol{\mu}^T \\ \boldsymbol{\mu} & \boldsymbol{\Sigma} \end{pmatrix}$ lies in the interior of a $(1+n) \times (1+n)$-dimensional completely positive cone.

Under assumption 1, we can equivalently reformulate problem (FOLI) into a conic problem (CO).

$$
\begin{aligned}
(CO) \quad L_{CO} = \quad & \min_{\boldsymbol{I}, \boldsymbol{Z}, \alpha_i, \beta_i, \theta_j, \tau, \boldsymbol{\xi}, \boldsymbol{\eta}} \boldsymbol{f}^T \boldsymbol{I} + \boldsymbol{h}^T \boldsymbol{Z} + \sum_{i \in [M]} (b_i \alpha_i + b_i^2 \beta_i) + \tau + 2\boldsymbol{\mu}^T \boldsymbol{\xi} + \boldsymbol{\Sigma} \bullet \boldsymbol{\eta} \\
s.t. \quad & \boldsymbol{I} \leq U \boldsymbol{Z} \\
& \boldsymbol{I} \geq \mathbf{0}, \boldsymbol{Z} \in \{0,1\}^m \\
& \begin{pmatrix} \tau & \boldsymbol{\xi}^T & \frac{1}{2}\left(\sum_{i \in [M]} \boldsymbol{a}_i \alpha_i - \boldsymbol{\theta} \right)^T \\ \boldsymbol{\xi} & \boldsymbol{\eta} & \mathbf{0} \\ \frac{1}{2}\left(\sum_{i \in [M]} \boldsymbol{a}_i \alpha_i - \boldsymbol{\theta} \right) & \mathbf{0} & \sum_{i \in [M]} \boldsymbol{a}_i \boldsymbol{a}_i^T \beta_i + \boldsymbol{Diag(\theta)} \end{pmatrix} + \frac{1}{2} \begin{pmatrix} \mathbf{0} & \mathbf{0} & \begin{bmatrix} \mathbf{0} \\ \boldsymbol{I} \\ \mathbf{0} \end{bmatrix}^T \\ \mathbf{0} & \mathbf{0} & \begin{bmatrix} -\boldsymbol{e}^{[n]} \\ \mathbf{0} \end{bmatrix}^T \\ \begin{bmatrix} \mathbf{0} \\ \boldsymbol{I} \\ \mathbf{0} \end{bmatrix} & \begin{bmatrix} -\boldsymbol{e}^{[n]} \\ \mathbf{0} \end{bmatrix} & \mathbf{0} \end{pmatrix} \succeq_{co} \mathbf{0}
\end{aligned}
$$

where $\boldsymbol{I} \in \mathbb{R}_+^m$, $\boldsymbol{Z} \in \{0,1\}^m$, $\boldsymbol{\alpha} \in \mathbb{R}^M$, $\boldsymbol{\beta} \in \mathbb{R}^M$, $\boldsymbol{\theta} \in \mathbb{R}^N$, $\tau \in \mathbb{R}$, $\boldsymbol{\xi} \in \mathbb{R}^n$, $\boldsymbol{\eta} \in \mathbb{R}^{n \times n}$ are decision variables. We use superscribe $^{[k]}$ to define a k-by-k matrix variable. For example, $\boldsymbol{e}^{[n]}$ is a n-dimensional identity matrix. $\boldsymbol{Diag(\theta)}$ represents a diagnal matrix with i-th diagnal element equal to θ_i. The first-two constraints are standard big-M constraint and non-negative constraint, where U is a big enough positive number. The third constraint with $\succeq_{co}$ is derived from $L(\boldsymbol{I})$. Formally, we propose this equivalent reformulation in Proposition 3.

Proposition 3. *Under assumption 1, problem (rFOLI) can be equivalently reformulated into a copositive-cone based mixed-integer conic problem (CO).*

Proof. Before diving into detailed proof, we concisely summarize the roadmap to prove this proposition. We first transform the second-stage problem $L(\boldsymbol{I})$ into a maximizing completely positive conic problem according to Natarajan et al. (2011) Theorem 3.3. Then taking dual to obtain a minimizing problem on copositive cone. We further show that strong duality holds between the maximizing and the minimizing problem. Finally, combining the second-stage minimizing problem with the first-stage problem gives (CO).

Step1: Transform $L(\boldsymbol{I})$

Before we transform $L(\boldsymbol{I})$ to a conic program, we first define three new variables.

- $\boldsymbol{p} := \mathbb{E}_{\mathbb{P}}[\boldsymbol{x}(\tilde{\boldsymbol{d}})] \in \mathbb{R}_+^N$
- $\boldsymbol{Y} := \mathbb{E}_{\mathbb{P}}[\boldsymbol{x}(\tilde{\boldsymbol{d}})\tilde{\boldsymbol{d}}^T] \in \mathbb{R}_+^{N\times n}$
- $\boldsymbol{X} := \mathbb{E}_{\mathbb{P}}[\boldsymbol{x}(\tilde{\boldsymbol{d}})\boldsymbol{x}(\tilde{\boldsymbol{d}})^T] \in \mathbb{R}_+^{N\times N}$

According to Natarajan et al. (2011) Theorem 3.3, since three assumptions are satisfied, we can reformulate $L(\boldsymbol{I})$ to a completely positive program $L_{CP}(\boldsymbol{I})$ as follows:

$$
\begin{aligned}
L(\boldsymbol{I}) = \quad L_{CP}(\boldsymbol{I}) = \quad & \max_{\boldsymbol{p},\boldsymbol{Y},\boldsymbol{X}} \begin{bmatrix} \boldsymbol{e}^{[n]} \\ \mathbf{0}_{(N-n)\times n} \end{bmatrix} \bullet \boldsymbol{Y} + \begin{bmatrix} \mathbf{0}_{n\times 1} \\ -\boldsymbol{I}_{m\times 1} \\ \mathbf{0}_{M\times 1} \end{bmatrix}^T \boldsymbol{p} \\
s.t. \quad & \boldsymbol{a}_i^T \boldsymbol{p} = b_i, \forall i \in [M] && (10) \\
& (\boldsymbol{a}_i \boldsymbol{a}_i^T) \bullet \boldsymbol{X} = b_i^2, \forall i \in [M] && (11) \\
& \boldsymbol{e}_{(j,j)} \bullet \boldsymbol{X} - \boldsymbol{e}_{(j)}^T \boldsymbol{p} = 0, \forall j \in [N] && (12) \\
& \begin{pmatrix} 1 & \boldsymbol{\mu}^T & \boldsymbol{p}^T \\ \boldsymbol{\mu} & \boldsymbol{\Sigma} & \boldsymbol{Y}^T \\ \boldsymbol{p} & \boldsymbol{Y} & \boldsymbol{X} \end{pmatrix} \succeq_{cp} \mathbf{0} && (13)
\end{aligned}
$$

We use superscribe $^{[k]}$ to define a k-by-k matrix variable. $\boldsymbol{e}^{[n]}$ is a n-dimensional identity matrix. We further use $\boldsymbol{e}_{(j,j)}$ to represent a matrix with (j,j)-position element equal to 1. And $\boldsymbol{e}_{(j)}$ is the unit vector with j-th element equal to 1. $\succeq_{cp}$ means that the matrix should belong to a completely positive cone. Except further explanation, $\mathbf{0}$ should be a zero vector or zero matrix with proper shape.

Step 2: Take duality on $L_{CP}(\boldsymbol{I})$

Taking Duality on $L_{CP}(\boldsymbol{I})$ gives the following results

$$L_{CO}(\boldsymbol{I}) = \min_{\alpha_i,\beta_i,\theta_j,\tau,\boldsymbol{\xi},\boldsymbol{\varphi},\boldsymbol{\psi},\boldsymbol{\eta},\boldsymbol{w}} \sum_{i\in[M]} b_i\alpha_i + b_i^2\beta_i + \tau + 2\boldsymbol{\mu}^T\boldsymbol{\xi} + \boldsymbol{\Sigma}\bullet\boldsymbol{\eta}$$

$$s.t. \quad \sum_{i\in[M]} \boldsymbol{a}_i\alpha_i - \sum_{j\in[N]} \boldsymbol{e}_{(j)}\theta_j - 2\boldsymbol{\varphi} = \begin{bmatrix} \mathbf{0}_{n\times 1} \\ -\boldsymbol{I}_{m\times 1} \\ \mathbf{0}_{M\times 1} \end{bmatrix} \tag{14}$$

$$\sum_{i\in[M]} \boldsymbol{a}_i\boldsymbol{a}_i^T\beta_i + \sum_{j\in[N]} \boldsymbol{e}_{(j,j)}\theta_j - \boldsymbol{w} = \mathbf{0} \tag{15}$$

$$-2\boldsymbol{\psi} = \begin{bmatrix} \boldsymbol{e}^{[n]} \\ \mathbf{0} \end{bmatrix} \tag{16}$$

$$\begin{pmatrix} \tau & \boldsymbol{\xi}^T & \boldsymbol{\varphi}^T \\ \boldsymbol{\xi} & \boldsymbol{\eta} & \boldsymbol{\psi}^T \\ \boldsymbol{\varphi} & \boldsymbol{\psi} & \boldsymbol{w} \end{pmatrix} \succeq_{co} \mathbf{0} \tag{17}$$

α_i, β_i, and θ_j is the dual variable for constraint (10), (11), and (12) respectively. Variable $\begin{pmatrix} \tau & \boldsymbol{\xi}^T & \boldsymbol{\varphi}^T \\ \boldsymbol{\xi} & \boldsymbol{\eta} & \boldsymbol{\psi}^T \\ \boldsymbol{\varphi} & \boldsymbol{\psi} & \boldsymbol{w} \end{pmatrix}$ is the dual variables for completely positive cone constraint (13). By weak duality theorem, $L_{CO}(\boldsymbol{I}) \geq L_{CP}(\boldsymbol{I})$.

Step 3: Strong duality holds, i.e. $L_{CO}(\boldsymbol{I}) = L_{CP}(\boldsymbol{I})$

We first divide the solution to (9) into two parts. One is composed of decision variables $\boldsymbol{\theta}$, and the other includes all of the slack variables $\boldsymbol{s}$, i.e., $\boldsymbol{x} = \left(\begin{smallmatrix} \boldsymbol{\theta} \\ \boldsymbol{s} \end{smallmatrix}\right), \boldsymbol{\theta} \in \mathbb{R}^{(n+m)}, \boldsymbol{s} \in \mathbb{R}^{(r+n+m)}$. According to Yan et al. (2018) Theorem 2, $L_{CO}(\boldsymbol{I}) = L_{CP}(\boldsymbol{I})$ holds as long as following two conditions hold:

1. The moment matrix $\left(\begin{smallmatrix} 1 & \boldsymbol{\mu}^T \\ \boldsymbol{\mu} & \boldsymbol{\Sigma} \end{smallmatrix}\right)$ lies in the interior of a $(1+n)\times(1+n)$-dimensional completely positive cone.

2. There exists a set of feasible solution $\boldsymbol{x}^{(k)} = \left(\begin{smallmatrix} \boldsymbol{\theta}^{(k)} \\ \boldsymbol{s}^{(k)} \end{smallmatrix}\right)$, $\forall k \in [K]$ to $\hat{\mathcal{L}}_{lp}^{\dagger}$ such that $\text{span}\{\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(K)}\} = \mathbb{R}^{n+m}$, and at least one of them is strictly positive; i.e., $\exists \boldsymbol{\theta}^{(l)} \in \{\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(K)}\}$ such that $\boldsymbol{\theta}^{(l)} > \mathbf{0}$

Since the first condition holds by assuming that Assumption 1 holds, we only need to prove the second condition is always satisfied. It is each to check that $\boldsymbol{x}^{(0)} = \left(\begin{smallmatrix} \boldsymbol{\theta}^{(0)} \\ \boldsymbol{s}^{(0)} \end{smallmatrix}\right) = \left(\begin{smallmatrix} \mathbf{1} \\ \mathbf{0} \end{smallmatrix}\right)$ is a feasible solution, and $\boldsymbol{\theta}^{(0)}$ is strictly positive.

There are another two sets of feasible solutions:

$$\mathcal{S}_1 := \left\{ \begin{bmatrix} \boldsymbol{\theta} \\ \boldsymbol{s} \end{bmatrix} \in \mathbb{R}^N \;\middle|\; \begin{bmatrix} \boldsymbol{\theta}^{(i)} \\ \boldsymbol{s}^{(i)} \end{bmatrix} = \begin{bmatrix} \begin{pmatrix} \mathbf{0} \\ \boldsymbol{e}_{(i)} \end{pmatrix} \\ \begin{pmatrix} \sum_{\gamma \in idx(\mathcal{E},i)} \boldsymbol{e}_{(\gamma)} \\ \mathbf{1} \\ \mathbf{1} - \boldsymbol{e}_{(i)} \end{pmatrix} \end{bmatrix}, \ \forall i \in [m] \right\}$$

where $idx(\mathcal{E}, i)$ is the index set that contains all indice of roads that are connected to node i, ensuring the first constraint in $\hat{\mathcal{L}}_{lp}^{\dagger}$ is met, and

$$\mathcal{S}_2 := \left\{ \begin{bmatrix} \boldsymbol{\theta} \\ \boldsymbol{s} \end{bmatrix} \in \mathbb{R}^N \;\middle|\; \begin{bmatrix} \boldsymbol{\theta}^{(j)} \\ \boldsymbol{s}^{(j)} \end{bmatrix} = \begin{bmatrix} \begin{pmatrix} \boldsymbol{e}_{(j)} \\ \sum_{i \in \Gamma(j)} \boldsymbol{e}_{(i)} \end{pmatrix} \\ \begin{pmatrix} \mathbf{0} \\ \mathbf{1} - \boldsymbol{e}_{(j)} \\ \mathbf{1} - \sum_{i \in \Gamma(j)} \boldsymbol{e}_{(i)} \end{pmatrix} \end{bmatrix}, \ \forall j \in [n] \right\}$$

It is also easy to check that all vectors $\boldsymbol{\theta}$ in $\mathcal{S}_1 \cup \mathcal{S}_2$ spans the space $\mathbb{R}^{(n+m)}$ by Gaussian Elimination. Therefore $\mathcal{S}_1 \cup \mathcal{S}_2 \cup \{\boldsymbol{x}^{(0)}\}$ forms a satisfied solution set to the second condition, which completes current proof step.

Step 4: Combine with first-stage problem

We further combine $L_{CO}(\boldsymbol{I})$ with first-stage problem and replace a part of decision variables in (17) with equations (14), (15), and (16). Now, we obtain the (rFOLI) problem in a Mixed-Integer Copositive Cone form, i.e. problem (CO).

□

3.3. Speed Up Branch&Bound

Although we have reformulated (rFOLI) into a convex conic problem based on copositive cone, Model (CO) still cannot be efficiently solved, not only because of the complexity of copositive cone itself but also the mixed-integer programming. The former issues could be practically tackled with a semi-definite positive relaxation technique, while the latter issue is not that trivial. In this subsection, we adopt the classical branch and bound framework with deep-first search to search optimal solutions. To speed up branch and bound procedure, we exploit known information over moments to figure out an interpretable initial solution instead of randomly generating an initial solution. Additionally, we put forward a dual-variable based heuristic to find out a better branch in each iteration to reduce nodes needed to visit by quickly finding a tighter linear relaxation bound. As the only integer decision variable is $\boldsymbol{Z}$, the acceleration technique proposed in this subsection only applied to $\boldsymbol{Z}$.

Since the main aim of building a new warehouse is to counter future uncertainty, an intuitive idea of determining an initial solution is ranking weighted average demand upper bound, then selecting the top-$\lfloor \frac{m}{2} \rfloor$ locations as an initial solution. Following this idea, we can figure out supply nodes that have the potential to relieve the uncertainty most. We first calculate a potential demand upper bound by $\bar{d}_j = \mu_j + \sigma_j, \forall j \in [n]$, where μ_j is the mean value and σ_j is the standard deviation, both of which are easy to obtain from moments information. Then calculate the weighted average demand upper bound for each supply node, $w_i = \frac{\sum_{j:\Gamma(i)} \bar{d}_j}{|\Gamma(i)|}, \forall i \in [m]$. Finally, we select $\lfloor \frac{m}{2} \rfloor$ supply nodes as the initial location for warehouses according to the following criterion.

$$
\boldsymbol{Z}^{init} = \begin{cases} Z_i = 1, & if \ (i) \leq \lfloor \frac{m}{2} \rfloor \\ Z_i = 0, & o.w. \end{cases} \tag{18}
$$

where (i) is the ranking index of supply node i in terms of w_i.

Another issue is about branching. Please note that the decision variable $\boldsymbol{\alpha}$ is the dual variable to first-M constraints of equation system $\hat{\mathcal{L}}^{\dagger}_{lp}$. And more specifically, $\alpha_k, \forall k \in [r]$ is the dual variable to first-r constraints, which is directly imposed on network structure. According to the duality theorem, economically, the optimal solution $\alpha^*_k, \forall k \in [r]$ can be interpreted as the value of corresponding road in current iteration. Similar arguments have been proposed in Yan et al. (2018), in which the authors' main purpose is designing a sparse network structure by iteratively deleting arcs from a full-flexible system. Since α^*_k is the value of i-th arc, they greedily delete the arc with the smallest value in each iteration. As to our branch and bound procedure, we can exploit the value of α^*_k as well. Intuitively, in each iteration, we greedily select one supply node, having the highest aggregated value of all roads originate from it, from remaining supply nodes as the next branch. In other words, we always choose the most valuable supply node as the next branch. To describe our idea mathematically clearer, we first define an index mapping from road (i,j) to a scalar index, representing the roads index in [r]. For example we use $(ij), \forall (i,j) \in \mathcal{E}$ to represent road (i,j)'s index in set [r]. Let $\mathcal{W}^t$ be the set of supply nodes that have not been explored before t-th iteration. Mathematically, we choose i^* as the next branch according to (19).

$$
i^* = \arg\max_{i \in \mathcal{W}^t} \sum_{j \in \Gamma(i)} \alpha_{(ij)} \tag{19}
$$

One thing worthy of attention is that the fixed cost and the holding cost are not considered. It is possible to involve them as a denominator for calculating the return of investment (ROI) of each supply node. However, since $\boldsymbol{\alpha}$ only reflects values of roads in current iteration, in which there is no inventory stored at the chosen location, ROI does not accurately reflect the importance of supply nodes. In some extreme cases, regarding ROI as a selection criterion leads to unnecessary searches in inexpensive but less connected supply nodes. From a practical perspective, adopting (19) as the criterion has already shortened computational time significantly.

Formally, we summarize the proposed accelerated branch and bound algorithm as Algorithm 1. We will further conduct numerical experiments to show the effect on reducing solving time.

Algorithm 1 Accelerated Branch and Bound Algorithm

Set $\boldsymbol{Z}^{init}$ according to (18)
Solve linear relaxation (CO) with $\boldsymbol{Z}^{init}$ to get L^{init}
Initialize $\boldsymbol{Z}^* \leftarrow \boldsymbol{Z}^{init}$, $L^* \leftarrow L^{init}$
Initialize $\boldsymbol{Z} \leftarrow \{0\}_{i\in[m]}$
Run DFS($\boldsymbol{Z}$)

```
function DFS(Z)
    Solve linear relaxation (CO) with Z to get current objective value L
    if L < L* then
        if Z is integer then
            L* ← L, Z* ← Z
        else
            Select branch i according to (19)
            Update Z_i = 1
            Run DFS(Z)
            Update Z_i = 0
            Run DFS(Z)
        end if
    end if
end function
```

4. Numerical Study

In this section, we conduct numerous experiments to examine the advantages of the proposed two-stage distributionally robust model, as well as the equivalent conic reformulation. First, We run our models in hundreds of randomly generated balanced but asymmetric networks with predetermined parameters. Second, we conduct sensitivity analysis on three dimensions, including correlation parameter ρ, coefficient of variation c_v, and risk attitude parameter κ. Finally, we extend our experiments to more general network structures and re-run all experiments in the first two steps, trying to obtain a comprehensive understanding of proposed models in a more general setting. A stochastic model and a mean-variance model are introduced as benchmarks throughout this section, which will be clearly illustrated in the next subsection. Experiments are conducted on a computer with 16GB memory and AMD 3700X CPU. We use Mosek 9.2 to solve conic problems and Gurobi 9.0.3 to solve mix-integer linear problems. All codes are open-sourced[1], and interested readers can easily reproduce the results of our experiments.

4.1. Experiment Setting

In this subsection, we first introduce a two-stage stochastic model and a mean-variance distributionally robust model as benchmarks. Then, a detailed explanation is given to describe how we generate synthetic data for simulation experiments.

4.1.1. benchmark

We introduce a two-stage stochastic model and a mean-variance distributionally robust model as benchmarks. The former model fully exploits all demand samples while the latter one only involves two statistics, mean and variance, which does not consider correlations between demand nodes. Since our model only exploits first-two moments, the information we have is slightly more than the mean-variance model has, but less than the stochastic model has.

SAA with Empirical Distribution

When the demand is uncertain, and the decision maker is risk-neutral, the network design problem can be formulated as a two-stage stochastic model as much literature did. Our risk-neutral two-stage stochastic programming model is:

$$
\begin{aligned}
(SAA) \quad \min_{\boldsymbol{I},\boldsymbol{Z}} \quad & \boldsymbol{f^T Z} + \boldsymbol{h^T I} + \mathbb{E}_{\widehat{\mathbb{P}}}\left[g(\boldsymbol{I}, \tilde{\boldsymbol{d}})\right] \\
\text{s.t.} \quad & \boldsymbol{I} \leq U\boldsymbol{Z}, \\
& \boldsymbol{I} \in \mathbb{R}^m_+, \boldsymbol{Z} \in \{0,1\}^m
\end{aligned}
$$

where $g(\boldsymbol{I}, \tilde{\boldsymbol{d}})$ is a recouse allocation problem for any given $\boldsymbol{I}$ and realized $\tilde{\boldsymbol{d}}$, exactly the same as Model (3). $(\boldsymbol{I}, \boldsymbol{Z})$ are the inventory and location selection decision variables. The point that distinguishes (SAA) from (rFOLI) is the distribution $\widehat{\mathbb{P}}$, which, for convenience, is the discrete empirical distribution. That is, the uniform distribution on the known training samples,

$$
\widehat{\mathbb{P}} = \frac{1}{N}\sum_{i=1}^{N} \delta_{\hat{\xi}_i}
$$

where $\delta_{\hat{\xi}_i}$ denotes the Dirac point mass at the i-th training sample $\hat{\xi}_i$. Since (SAA) model involves training samples, and by tradition, we regarded this benchmark as sample average approximation.

Mean-Variance

[1]Source codes are available at GitHub: Hyperlink is temporarily disabled for peer review

Another benchmark is a distributionally robust model with only mean $\boldsymbol{\mu}$ and variance $\boldsymbol{\sigma}^2$ information, i.e., ignore information on covariance.

$$
\begin{aligned}
(MV) \quad \min_{\boldsymbol{I},\boldsymbol{Z}} \quad & \boldsymbol{f^T Z} + \boldsymbol{h^T I} + \sup_{\mathbb{P}\in\mathscr{F}_{mv}(\boldsymbol{\mu},\boldsymbol{\sigma^2})} \mathbb{E}_{\mathbb{P}}\left[g(\boldsymbol{I},\tilde{\boldsymbol{d}})\right] \\
\text{s.t.} \quad & \boldsymbol{I} \le U\boldsymbol{Z}, \\
& \boldsymbol{I} \in \mathbb{R}_+^m, \boldsymbol{Z} \in \{0,1\}^m
\end{aligned}
$$

where the ambiguity set is

$$
\mathscr{F}_{mv}(\boldsymbol{\mu},\boldsymbol{\sigma}) = \left\{ \mathbb{P} \in \mathcal{P}\left(\mathbb{R}_+^n\right) \left| \begin{array}{l} \tilde{\boldsymbol{d}} \sim \mathbb{P} \\ \mathbb{E}_{\mathbb{P}}[\tilde{\mathbf{d}}] = \boldsymbol{\mu} \\ \mathbb{E}_{\mathbb{P}}[(\tilde{d}_j - \mu_j)^2] = \sigma_j^2 \\ \mathbb{P}[\tilde{\boldsymbol{d}} \in \mathbb{R}_+^n] = 1 \end{array} \right. \right\}
$$

$\sigma_j^2, \forall j \in [n]$ is the variance of $\tilde{d}_i$. Compared to (rFOLI), the most noticeble distinction is the construction of distribution ambiguity set. Although (MV) is still intractable in most cases, we can equivalent transform $\mathscr{F}_{mv}(\boldsymbol{\mu},\boldsymbol{\sigma})$ to $\mathscr{F}(\boldsymbol{\mu},\boldsymbol{\Sigma}_{mv})$, where $\boldsymbol{\Sigma}_{mv} = \boldsymbol{\mu}\boldsymbol{\mu}^T + \boldsymbol{Diag(\sigma^2)}$. Then, every analysis still goes through as shown in Proposition 3. And replaceing $\boldsymbol{\Sigma}$ in (CO) with $\boldsymbol{\Sigma}_{mv}$ gives the resulting conic model. Actually, for some special cases such appointment scheduling problem with $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}^2$, a hidden tractable reformulation is available (see Mak et al. (2015)).

4.1.2. Data Generation

Suppose we consider a network with m potential warehouses and n demand nodes. We randomly generate dozens of or hundreds of networks, and each network is defined by a six-elements tuple $(\mathcal{W}, \mathcal{R}, \mathcal{E}, \boldsymbol{f}, \boldsymbol{h}, \boldsymbol{\mu})$. $\mathcal{W}$ and $\mathcal{R}$ are the set of potential warehouse nodes and demand nodes. Temporarily, we assume $|\mathcal{W}| = |\mathcal{R}| = 8$. $\mathcal{E}$ is a set of links. According to Ni et al. (2018), the average node degree of a typical road network is about 2.4. This is also align with industrial practices in manufacturing, disaster management, and school bus routing design (Jordan and Graves (1995), Mete and Zabinsky (2010), and Bertsimas et al. (2019a)). Therefore, we require $|\mathcal{E}| = 1.2(m+n)$. The following two constraints on the generated network should be satisfied: (1) at least one road linked to each supply node (2) at least one road linked to each demand node. These conditions help avoid the appearance of isolated nodes. $\boldsymbol{f}$ is the fixed setup cost randomly draw from a uniform distribution U[10, 25]. $\boldsymbol{h}$ is the holding cost drawn from U[0.1, 0.2]. And $\boldsymbol{\mu}$ is the first moment, i.e. the mean value, of demand, drawn from U[400, 600].

Initially, we set the correlation coefficient $\rho = 0.3$, coefficient of variation $c_v = 0.3$, and risk attitude coefficient $\kappa = 1.0$, all of which will be further modified to conduct sensitivity analysis. Therefore, the standard deviation of each demand node is $\sigma_j = \mu_j c_v, \forall j \in [n]$. For simplicity, we assume the correlation parameter ρ and coefficient of variation c_v are applied to all demand node pairs or demand node, which means the covariance of any two demand nodes (i, j) is $\mu_i \mu_j c_v^2 \rho,\ \forall i, j \in \mathcal{R}$. We further assume the underlying demand distribution follows a multivariate gaussian distribution, whose domain is defined as $\mathbb{R}_+^n$, the mean value is $\boldsymbol{\mu}$, and the covariance matrix is $\boldsymbol{\Sigma}_{\rho,c_v}$ as defined above. We use the subscripts $_{\rho,c_v}$ to emphasize that the covariance matrix depends on ρ and c_v.

For model (CO), we suppose the first-moment $\boldsymbol{\mu}$ and second-moment matrix $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_{\rho,c_v} + \boldsymbol{\mu}\boldsymbol{\mu}^T$ are known to decision makers. For model (MV), only mean and variance of demand are known. And for model (SAA), we assume there are 30 historical samples available. The sample size is statistically large enough to obtain unbiased estimations of first-two moments.

After obtaining optimal solutions through solving three models, we randomly generate 2000 demand realizations to conduct the second-stage simulations. 1000 demand realizations are drawn from underlying true multivariate gaussian distribution. Another 1000 demand realizations are drawn from a mixture distribution with four independent and equally weighted components, including two-point distribution $\mathbb{P}(\tilde{\boldsymbol{d}} = \boldsymbol{\mu} - \boldsymbol{\sigma}) = \mathbb{P}(\tilde{\boldsymbol{d}} = \boldsymbol{\mu} + \boldsymbol{\sigma}) = 0.5$, element-wise independent uniform distribution $U\left[\boldsymbol{\mu} - \sqrt{3}\boldsymbol{\sigma}, \boldsymbol{\mu} + \sqrt{3}\boldsymbol{\sigma}\right]$, independent normal distribution $d_j \sim N(\mu_j, \sigma_j), \forall j \in [n]$, and log-normal distribution with mean $\boldsymbol{\mu}$ and standard deviatio $\boldsymbol{\sigma}$. It is easy to check that the mixture distribution has mean values and standard deviations approximately around true $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ but ignores correlations. We separately analyze results on Multivariate Gaussian Distribution (MGD) and Mixture Distribution (MixD), and for each network, we regard the average (or worst-case) result on 1000 simulations as the corresponding network's performance.

4.2. (8, 8) Network

4.2.1. (ρ, c_v, κ)=(0.3, 0.3, 1.0)

Following the initial setting as depicted in the above subsection, we first restrict ourselves to the (8, 8) structure for a closer look at the model's performance. We randomly generate 200 networks with $(\rho, c_v, \kappa) = (0.3, 0.3, 1.0)$. Except for further explanations, all analyses in this subsection rely on the 200 networks: for each network, we construct a measurement based on simulation results, for example, the average unmet demand, to represent that network's performance. And then, we calculate the average on 200 networks.

First-stage Deployment

First of all, we compare the first-stage deployments and characteristics of designed networks. Table 1 summarized the first-stage decisions and resulting networks. The first column, named by *total inv*, represents the average total inventory of 200 networks. The number in the parenthesis shows the standard deviation. It is clear that Model (CO) results in the highest inventory level while Model (MV) has the lowest inventory level. The difference comes from whether the strong correlation of demand is taken into account. The second column, *# of w.h.*, reveals the number of established warehouses on average. Model (SAA) selects almost half of potential locations to build warehouses. The third column is the number of roads in designed

Table 1: Summary on Designed Networks

	total inv.	# of w.h.	# of roads	w.h. degree	d.n. degree
SAA	4903.43	4.06	11.57	2.91	1.45
	(300.33)	(0.78)	(1.57)	(0.45)	(0.20)
CO	4944.17	3.79	10.96	2.96	1.37
	(201.06)	(0.72)	(1.52)	(0.48)	(0.19)
MV	4638.28	3.77	11.15	3.02	1.39
	(194.14)	(0.72)	(1.44)	(0.48)	(0.18)

networks, which implies that Model (CO) designs the most sparse network. The last two columns exhibit warehouse nodes' and demand nodes' degree, respectively.

Cost

We next check the cost performances, including first-stage deployment cost and second-stage penalty due to unmet demand, which are the most concerned. Table 2 shows the average costs on 200 networks under

Table 2: Cost Comparison, $(m, n) = (8, 8), (\rho, c_v, \kappa) = (0.3, 0.3, 1.0)$

			MGD		MixD	
	f	**h**	**p**	total cost	**p**	total cost
SAA	67.7	660.2	64.5	792.5	70.0	798.0
	(14.9)	(70.5)	(31.7)	(78.2)	(38.8)	(81.0)
CO	63.4	668.5	53.1	785.0	57.3	789.2
	(13.9)	(66.5)	(12.2)	(78.5)	(15.0)	(82.0)
MV	63.3	636.1	96.1	795.5	108.7	808.1
	(13.7)	(66.1)	(15.4)	(80.1)	(15.8)	(81.3)

different models with MGD and MixD realizations respectively. The number in parenthesis represents the standard deviation. First two columns represent the setup cost $\boldsymbol{f}$ and the holding cost $\boldsymbol{h}$. $\boldsymbol{p}$ is the penalty cost, or equivalently, the number of unsatisfied demand. *total cost* is the aggregated cost. It is clear that Model (CO) always achieves the lowest total cost. One remarkable decrease appears in $\boldsymbol{p}$. It decreases from about 64.5 (by SAA) to 53.1 (by CO), i.e., 17.7% decreasing in unmet demand under the in-sample simulation. For MixD simulation, the decrease is also astounding, from 70.0 (by SAA) to 57.3, about 18.4%, or from 108.7 (by MV) to 57.3, about 47.3%. Although the absolute decreasing value is marginal, the relative decreasing is remarkable, which underlines the advantages of our proposed conic model and reformulation technique: we can further significantly reduce the amount of unmet demand even it is already small. This managerial insight is quite attractive, especially to disaster management experts, or fulfillment-oriented warehouse managers, since their priority is reducing unmet demand without boosting expenditure.

We next scrutinize the unmet demand more deeply. Figure 1 shows the histograms about average unmet demand, where the horizontal axis shows the average unmet demand obtaining through simulations, and the vertical axis represents the number of networks falling in the corresponding bins. In the left graph, when

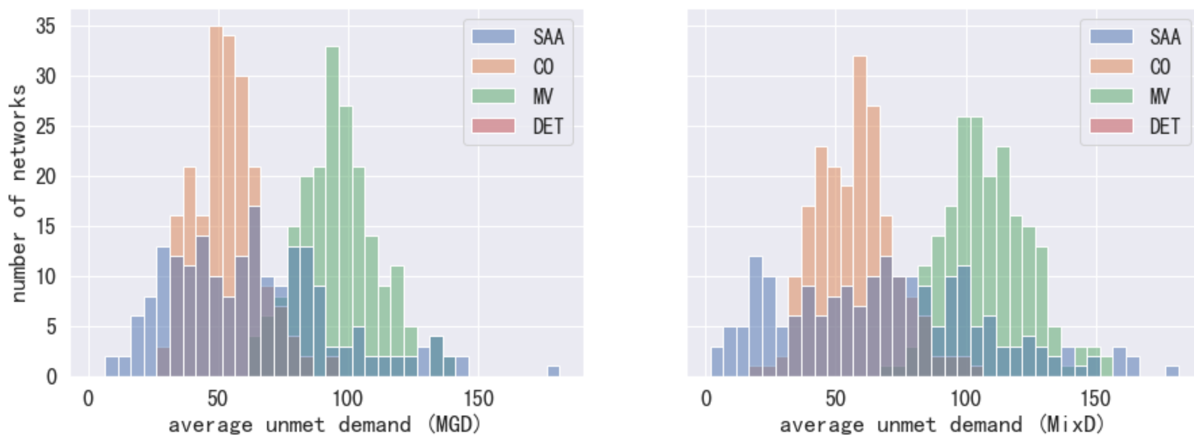

Figure 1: Histogram on average unmet demand, $(\rho, c_v, \kappa) = (0.3, 0.3, 1.0)$

the true demand distribution aligns with the distribution that decision makers know, it is easy to see that Model (CO) dominates Model (MV) except in some instances overlapping at around 75. Although Model (CO) achieves a higher level of concentration around 60, the least amount of unmet demand is achieved by Model (SAA), whose performance is more scattered, entailing several worst cases at 150 levels. The right graph shows the results from MixD simulations. Similarly, Model (CO) dominates Model (MV) most of the time. However, the most interesting change is the increasing dispersion of Model (SAA). Clear to see, the performance of Model (SAA) becomes unstable in different networks while (CO) and (MV) still are concentrated in their former mean values. The changes in standard deviations of $\boldsymbol{p}$ in Table 2 also verify our findings.

Service Level

Another critical measurement is the service level. Table 3 summarizes two kinds of service levels. Type-1 service level, or α service level, is an event-oriented performance criterion. It measures the probability that all coming demand will be completely delivered from stock on hand. Type-2 service level, or β service level, is a quantity-oriented performance measure describing the proportion of satisfied demand to total demand. It is obvious that Model (CO) performs a little bit better than Model (SAA) in terms of type 2 service level. As to type 1 service level, Model (CO) outperforms both other models about 1%.

Table 3: Service Level, $(m, n) = (8, 8), (\rho, c_v, \kappa) = (0.3, 0.3, 1.0)$

	MGD		MixD	
service level (%)	type 1	type2	type1	type2
SAA	94.79	98.76	93.75	98.59
CO	95.61	98.99	94.32	98.85
MV	93.82	98.16	93.10	97.85

Value-at-Risk

Furthermore, we calculate the 99.9%, 95%, 90%, 85%, and 80% quantile of unmet demand to show the ability of models to hedge long-tail risk. When the true distribution is known to decision makers, these quantiles could be regarded as estimators of the value-at-risk (VaR) of unmet demand after implementing the corresponding first-stage solutions. Table 4 shows the VaR values under different quantiles. *MGD VaR* shows that Model (CO) dominates other models under all circumstances, which underlines its ability to hedge long-tail risk comes from the true underlying distribution. Surprisingly, the columns *MixD VaR* in

Table 4: Unmet Demand with Various Quantiles

	MGD VaR					MixD VaR				
quantile	99.9%	95%	90%	85%	80%	99.9%	95%	90%	85%	80%
SAA	1323.3	406.4	212.8	111.0	56.3	798.3	335.4	325.9	250.0	119.8
CO	1280.0	350.9	163.2	71.8	27.5	761.8	267.9	265.0	224.8	97.5
MV	1585.0	617.8	365.5	206.2	100.2	968.8	569.7	569.7	398.3	171.5

Table 4 further reveals that Model (CO) is robust enough to the misspecification in distribution. In terms of 95% quantile, Model (CO)'s the amount of unmet demand under MixD is 20% lower than that of Model (SAA), while the gap is only 13.6% under MGD situation. This phenomenon complementarily verifies the advantages of Model (CO) and distributionally robust technique, especially when historical data is limited, and misspecification in distribution is likely to happen.

CPU Time

Computational time is also essential in practice. Although the problem we consider is strategy-level, a reasonable computational time is indeed necessary. Therefore, we record the CPU time for solving each optimization problem, underlining the superiority of proposed branch and bound acceleration algorithm. Please note that, since we run experiments with four parallel tasks synchronously on one computer, the CPU time will be slightly longer. Table 5 summarizes the time needed to solve each model. We use *(acce)* to represent the model is solved with accelerated branch and bound technique proposed in Section 3.3. The

Table 5: CPU Time to Solve Models

	CPU Time (s)	Node
SAA	0.05	-
CO	748.79	98.80
CO (acce)	337.30	44.09
MV (acce)	355.55	43.88

first column named as *CPU Time (s)* reports computational time in seconds to obtain the optimal solution, and the second column, *Node*, exhibits the number of visited branch and bound node before terminating the algorithm. Obviously, for Model (CO), the accelerated algorithm reduces more than half time by selecting better branches, finding tighter bounds, and visiting fewer nodes. The decreasing computational time also demonstrates that involving more variables into reformulated model in Proposition 1 is worthy.

4.2.2. Sensitivity Analysis

We modify some factors (correlation parameter ρ, coefficient of variation c_v, and risk attitude κ) to conduct sensitivity analysis. Since we have explored concerned measurement in detail, throughout this part, we only consider 50 graphs for each parameter combination. And we focus on MixD simulations where the true underlying distribution deviates from the distribution used in training sample generation process. This situation is more likely to happen in reality, and hence a robust enough warehouse network is needed. Results show that all the conclusions proposed earlier still remain the same tendency.

Correlation Parameter ρ

We first show a sensitivity analysis on the correlation parameter ρ. We modify it from 0.05 to 0.5. Figure

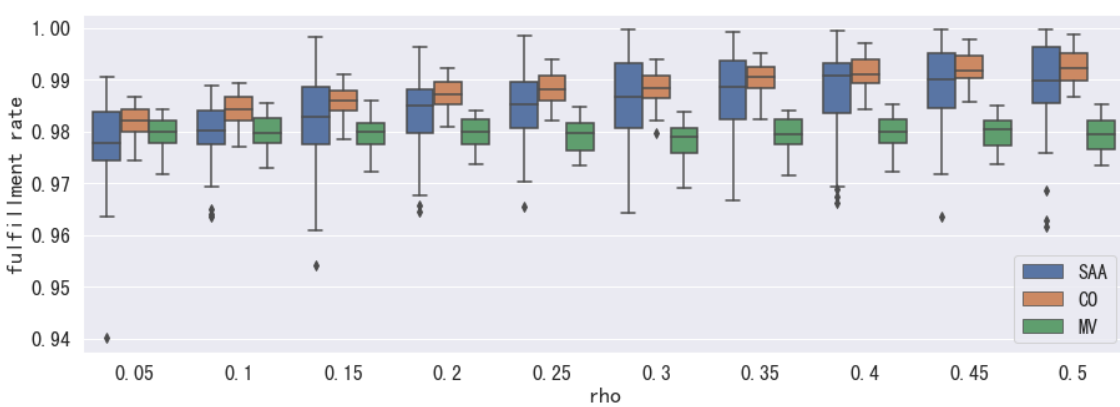

Figure 2: Sensitivity Analysis on ρ

2 shows fulfillment rates of three models under different parameter ρ. Fulfillment rate, or type 2 service level, measures what proportion of demand has been satisfied. It is evident that Model (CO) always has a higher fulfillment rate, on average, than other models have. Moreover, the perturbation of Model (CO) is much less severe than others', implying (CO) is robust to network structure. An interesting trend is that the fulfillment rate of Model (MV) is stable over ρ while that of other models increases. The difference comes from models' abilities to utilize correlation information. While (MV) does not consider any correlation between demand nodes, (CO) and (SAA) take that into consideration explicitly or implicitly, respectively. Another noteworthy phenomenon is that outliers only appear in Model (SAA), reflecting its weakness of hedging extreme risks.

Figure 3 exhibits the total cost of three models under different ρ. Model (CO) can always achieve the

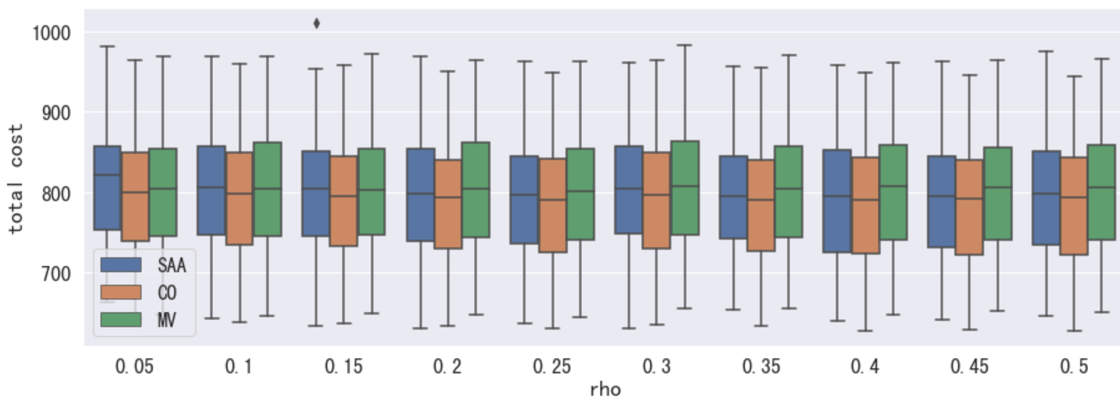

Figure 3: Sensitivity Analysis on ρ

lowest total cost.

Coefficient of Variation c_v

Same as the analysis for ρ, we first look at the fulfillment rate. Figure 4 depicts fulfillment rates under varied c_v. As c_v grows, fulfillment rates decrease in general, and variances increase. However, among three models, (CO) has the lowest decreasing rate and the smallest variance, which further emphasizes the robustness of Model (CO). Figure 5 shows the changes of total costs over c_v. It increases as c_v becomes larger since more warehouses and products are needed to counter the increasing uncertainty.

Risk Attitude Parameter κ

Figure 6 It represents the relationship between fulfillment rate and risk attitude κ. As κ grows, which implies the decision maker focuses more on first-stage cost, the fulfillment rate decreases significantly. This is a direct result of changing risk attitude since larger κ reflects a more aggressive attitude to future risk. And therefore, corresponding prepositioning decisions would become less conservative. Another noteworthy phenomenon

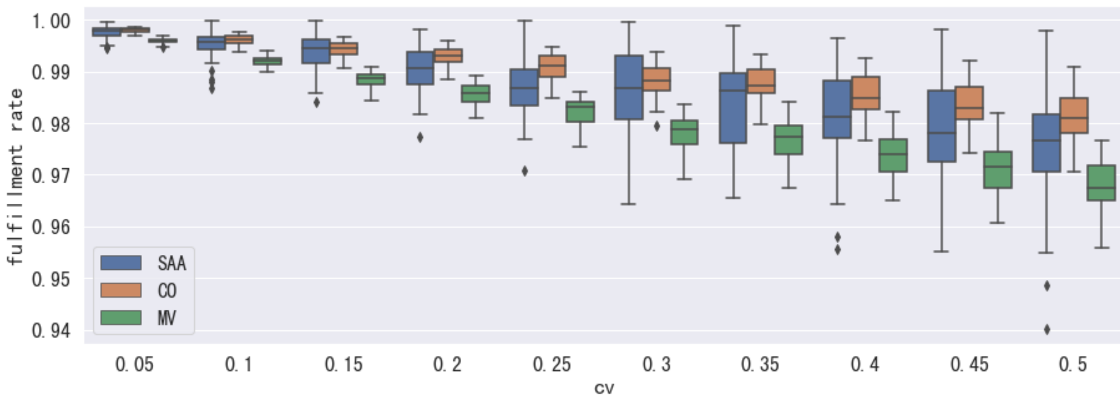

Figure 4: Sensitivity Analysis on c_v

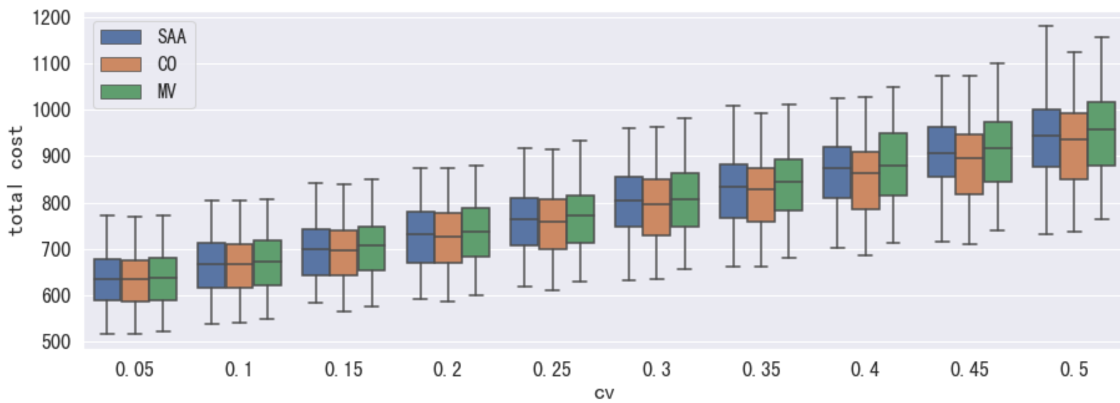

Figure 5: Sensitivity Analysis on c_v

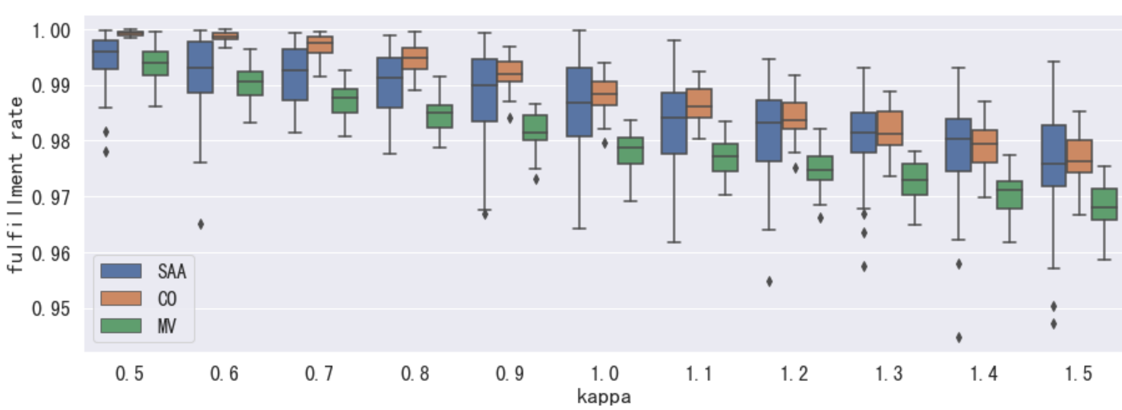

Figure 6: Sensitivity Analysis on κ

is the diminishing performance gap between (CO) and (SAA) as κ increases. Since more attention shifts to deployment cost, the advantage of (CO) in dealing with demand uncertainty is gradually cannibalized. Therefore, (CO)'s superiority in fulfillment rate diminishes. Figure 7 expresses total costs of models when κ

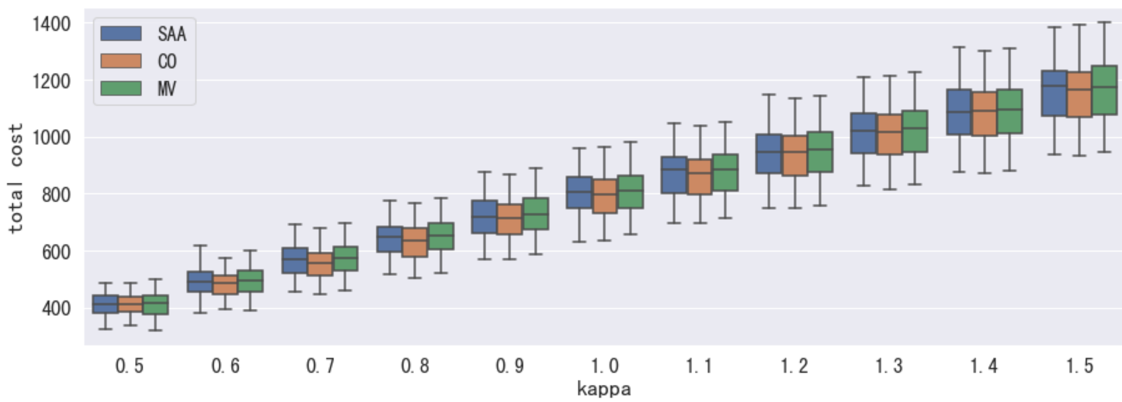

Figure 7: Sensitivity Analysis on κ

increases. We find that (CO) has a slightly lower total cost, no matter how κ changes.

To sum up this subsection, we conduct numerous experiments on a given balanced system to evaluate the performance of the proposed copositive cone based conic model. We find that the conic model is robust enough to demand uncertainty in most cases and outperforms SAA-based model when the historical data is very limited but statistically large enough to obtain moments statistics. We also find that it is ultimately essential to incorporate correlation information into modeling for situations like disaster management where demand correlation indeed exists and plays an unignorable role. As variances or correlations grows, they become more influential in consequent second-stage fulfillment performance. We also demonstrated that the acceleration algorithm indeed shortens computational times.

4.3. General Network

In this subsection, we conduct more simulation experiments on general unbalanced networks and compare these results with $(8,8)$ networks. More specifically, we consider $(4,8)$ and $(8,12)$ respectively. Except for changes in the number of potential supply nodes and demand nodes, other parameters remain the same. We also modify three parameters, including correlation parameter ρ, coefficient of variation c_v, and risk attitude parameter κ to conduct sensitivity analysis. Generally speaking, we re-run all experiments in this section but on different networks. Table 6 summarizes the performance of different networks in terms of Fulfillment Rate(FR). We regard the performance of SAA as the benchmark, and ratios in Table 6 are calculated according to $\frac{FR_x - FR_{SAA}}{FR_{SAA}} \times 100\%, x \in \{CO, MV\}$.

Table 6: Fulfillment rate comparison on general networks

Fulfillment Rate			In-sample									Out-sample								
Relative Changes			(4, 8)			(8, 8)			(8, 12)			(4, 8)			(8, 8)			(8, 12)		
$\boldsymbol{\rho}$	$\boldsymbol{c_v}$	$\boldsymbol{\kappa}$	SAA	CO	MV	SAA	CO	MV	SAA	CO	MV	SAA	CO	MV	SAA	CO	MV	SAA	CO	MV
0.05	0.3	1.0	-	0.30%	0.12%	-	0.35%	0.21%	-	0.31%	0.15%	-	0.71%	-0.03%	-	0.47%	0.21%	-	0.47%	0.17%
0.10	0.3	1.0	-	0.29%	-0.07%	-	0.32%	0.04%	-	0.32%	-0.02%	-	0.48%	-0.91%	-	0.43%	-0.02%	-	0.45%	-0.06%
0.15	0.3	1.0	-	0.18%	-0.36%	-	0.27%	-0.16%	-	0.22%	-0.30%	-	0.26%	-1.68%	-	0.34%	-0.29%	-	0.28%	-0.41%
0.20	0.3	1.0	-	0.33%	-0.40%	-	0.33%	-0.25%	-	0.09%	-0.62%	-	0.81%	-1.66%	-	0.40%	-0.36%	-	0.08%	-0.77%
0.25	0.3	1.0	-	0.17%	-0.71%	-	0.26%	-0.45%	-	0.28%	-0.59%	-	0.15%	-2.76%	-	0.34%	-0.56%	-	0.32%	-0.63%
0.30	0.3	1.0	-	0.33%	-0.74%	-	0.21%	-0.68%	-	0.31%	-0.73%	-	0.66%	-2.71%	-	0.19%	-0.84%	-	0.34%	-0.75%
0.35	0.3	1.0	-	0.27%	-0.95%	-	0.21%	-0.80%	-	0.21%	-1.00%	-	0.68%	-3.04%	-	0.26%	-0.83%	-	0.23%	-0.95%
0.40	0.3	1.0	-	0.41%	-0.96%	-	0.30%	-0.85%	-	0.18%	-1.19%	-	0.92%	-3.23%	-	0.34%	-0.82%	-	0.21%	-1.05%
0.45	0.3	1.0	-	0.16%	-1.40%	-	0.34%	-0.97%	-	0.19%	-1.33%	-	0.57%	-3.87%	-	0.36%	-0.88%	-	0.21%	-1.13%
0.50	0.3	1.0	-	0.11%	-1.55%	-	0.29%	-1.14%	-	0.20%	-1.47%	-	0.45%	-4.32%	-	0.33%	-1.00%	-	0.22%	-1.20%
0.3	**0.05**	1.0	-	0.06%	-0.20%	-	0.05%	-0.16%	-	0.09%	-0.17%	-	0.09%	-0.63%	-	0.05%	-0.16%	-	0.08%	-0.15%
0.3	**0.10**	1.0	-	0.16%	-0.32%	-	0.11%	-0.30%	-	0.11%	-0.38%	-	0.35%	-1.03%	-	0.11%	-0.31%	-	0.11%	-0.33%
0.3	**0.15**	1.0	-	0.15%	-0.52%	-	0.09%	-0.48%	-	0.11%	-0.58%	-	0.23%	-1.73%	-	0.08%	-0.51%	-	0.09%	-0.54%
0.3	**0.20**	1.0	-	0.19%	-0.64%	-	0.22%	-0.48%	-	0.20%	-0.65%	-	0.48%	-2.01%	-	0.24%	-0.49%	-	0.21%	-0.60%
0.3	**0.25**	1.0	-	0.39%	-0.55%	-	0.33%	-0.47%	-	0.27%	-0.70%	-	0.98%	-1.96%	-	0.38%	-0.51%	-	0.29%	-0.64%
0.3	**0.30**	1.0	-	0.33%	-0.74%	-	0.21%	-0.68%	-	0.31%	-0.73%	-	0.66%	-2.71%	-	0.19%	-0.84%	-	0.34%	-0.75%
0.3	**0.35**	1.0	-	0.24%	-0.91%	-	0.35%	-0.59%	-	0.24%	-0.88%	-	0.46%	-3.20%	-	0.40%	-0.70%	-	0.22%	-0.97%
0.3	**0.40**	1.0	-	0.47%	-0.73%	-	0.30%	-0.67%	-	0.15%	-1.00%	-	1.16%	-2.83%	-	0.39%	-0.82%	-	0.17%	-1.10%
0.3	**0.45**	1.0	-	0.36%	-0.87%	-	0.43%	-0.60%	-	0.46%	-0.75%	-	1.15%	-3.04%	-	0.52%	-0.76%	-	0.52%	-0.88%
0.3	**0.50**	1.0	-	0.47%	-0.84%	-	0.40%	-0.64%	-	0.39%	-0.84%	-	1.59%	-2.87%	-	0.54%	-0.84%	-	0.49%	-1.01%
0.3	0.3	**0.5**	-	0.47%	-0.10%	-	0.45%	-0.02%	-	0.47%	-0.09%	-	1.60%	-1.12%	-	0.46%	-0.09%	-	0.58%	-0.14%
0.3	0.3	**0.6**	-	0.46%	-0.27%	-	0.51%	-0.07%	-	0.40%	-0.32%	-	1.99%	-1.73%	-	0.68%	-0.15%	-	0.54%	-0.44%
0.3	0.3	**0.7**	-	0.47%	-0.39%	-	0.37%	-0.32%	-	0.45%	-0.38%	-	1.97%	-2.03%	-	0.56%	-0.44%	-	0.64%	-0.47%
0.3	0.3	**0.8**	-	0.29%	-0.66%	-	0.36%	-0.42%	-	0.25%	-0.67%	-	1.11%	-2.75%	-	0.50%	-0.53%	-	0.37%	-0.76%
0.3	0.3	**0.9**	-	0.34%	-0.68%	-	0.32%	-0.51%	-	0.33%	-0.65%	-	0.91%	-2.67%	-	0.39%	-0.62%	-	0.39%	-0.71%
0.3	0.3	**1.0**	-	0.33%	-0.74%	-	0.21%	-0.68%	-	0.31%	-0.73%	-	0.66%	-2.71%	-	0.19%	-0.84%	-	0.34%	-0.75%
0.3	0.3	**1.1**	-	0.15%	-0.94%	-	0.29%	-0.61%	-	0.11%	-0.98%	-	0.22%	-2.91%	-	0.31%	-0.67%	-	0.07%	-0.99%
0.3	0.3	**1.2**	-	0.32%	-0.78%	-	0.24%	-0.67%	-	-0.02%	-1.11%	-	0.23%	-2.70%	-	0.23%	-0.72%	-	-0.07%	-1.10%
0.3	0.3	**1.3**	-	0.09%	-1.03%	-	0.13%	-0.81%	-	0.24%	-0.87%	-	0.07%	-2.65%	-	0.08%	-0.85%	-	0.21%	-0.78%
0.3	0.3	**1.4**	-	0.21%	-0.91%	-	0.10%	-0.82%	-	0.01%	-1.09%	-	0.11%	-2.45%	-	0.03%	-0.86%	-	-0.05%	-1.02%
0.3	0.3	**1.5**	-	0.04%	-1.05%	-	0.08%	-0.84%	-	0.09%	-1.00%	-	-0.27%	-2.68%	-	0.06%	-0.80%	-	0.02%	-0.93%

In general, Model (CO) always achieves higher fulfillment rates than (SAA) does except in several situations, while Model (MV) achieves lower fulfillment rates than (SAA) does. When we gradually increase ρ from 0.05 to 0.5 as shown in the table's first part, the performance of (MV) gradually becomes worse, which implies the importance of considering correlations between nodes when correlations indeed exist. As c_v grows, shown in the second part of Table 6, the relative performance increase in terms of fulfillment rate also rises. This phenomenon emphasizes that when the variance of demand is high enough, explicitly considering cross-moments information, instead of only marginal-moments, can further improve fulfillment rate.

Moreover, we also summarize relative changes in Total Cost(TC) compared to (SAA) as described in Table 7. The relative changes are calculated according to $\frac{TC_x - TC_{SAA}}{TC_{SAA}} \times 100\%, x \in \{CO, MV\}$. Similarly, (CO) always dominates (SAA) by achieving lower costs except for several cases. Surprisingly, the maximum decrease in total cost can reach 14.96%.

5. Extensions

5.1. Multi-items and Multi-periods

In this subsection, we extend our fulfillment-oriented location-inventory model to a multiple-items and multiple-periods model. Supposed that there are K different kinds of items, and the total planning period is S. And further, we assume demand variables are uncorrelated over items and periods. Under this assumption, the problem is separatable and can be modified as:

$$\begin{aligned} \min_{\boldsymbol{I}_{ks}, \boldsymbol{Z}} \quad & \kappa(\boldsymbol{f}^T \boldsymbol{Z} + \sum_{k \in [K]} \sum_{s \in [S]} \boldsymbol{h}^T \boldsymbol{I}_{ks}) + \sum_{k \in [K]} \sum_{s \in [S]} \sup_{\mathbb{P} \in \mathscr{F}(\boldsymbol{\mu}_{ks}, \boldsymbol{\Sigma}_{ks})} \mathbb{E}_{\mathbb{P}} \left[g_{ks}(\boldsymbol{I}_{ks}, \tilde{\boldsymbol{d}}) \right] \\ \text{s.t.} \quad & \boldsymbol{I}_{ks} \leq U \boldsymbol{Z}, \ \forall k \in [K], s \in [S] \\ & \boldsymbol{I}_{ks} \in \mathbb{R}_+^m, \boldsymbol{Z} \in \{0,1\}^m \end{aligned} \tag{20}$$

Let $L_{ks}(\boldsymbol{I}) = \sup_{\mathbb{P} \in \mathscr{F}(\boldsymbol{\mu}_{ks}, \boldsymbol{\Sigma}_{ks})} \mathbb{E}_{\mathbb{P}} \left[g_{ks}(\boldsymbol{I}, \tilde{\boldsymbol{d}}) \right]$. Following the analysis in Section 3.2, we can equivalently reformulate $L_{ks}(\boldsymbol{I})$ as a conic programming. And the resulting model will contain $|K| \times |S|$ copositive cone constraints, which practically would slow down the computational process but theoretically does not make the model intractable.

5.2. Capacitated Warehouses

Capacitated Warehouses is most the case in reality, and the setup cost on warehouses $\boldsymbol{f}$ is possibly capacity-specified. Since these constraints are imposed on first-stage, they are easy to incorporate into our model. Suppose K types of warehouses are available, and each of them has a limited capacity C_k with a

Table 7: Total cost comparison on general networks

Total Cost			In-sample									Out-sample								
Relative Changes			(4, 8)			(8, 8)			(8, 12)			(4, 8)			(8, 8)			(8, 12)		
ρ	c_v	κ	SAA	CO	MV	SAA	CO	MV	SAA	CO	MV	SAA	CO	MV	SAA	CO	MV	SAA	CO	MV
0.05	0.3	1.0	-	-0.56%	-0.57%	-	-0.95%	-1.03%	-	-0.64%	-0.84%	-	-2.46%	0.30%	-	-1.69%	-1.01%	-	-1.66%	-0.94%
0.10	0.3	1.0	-	-0.92%	-0.68%	-	-0.48%	-0.48%	-	-0.74%	-0.69%	-	-1.75%	3.65%	-	-1.21%	-0.08%	-	-1.56%	-0.40%
0.15	0.3	1.0	-	-0.72%	-0.13%	-	-1.03%	-0.73%	-	-0.49%	-0.11%	-	-1.03%	6.80%	-	-1.47%	0.10%	-	-0.84%	0.64%
0.20	0.3	1.0	-	-0.86%	0.17%	-	-0.77%	-0.18%	-	-0.78%	0.13%	-	-3.26%	6.73%	-	-1.25%	0.54%	-	-0.69%	1.12%
0.25	0.3	1.0	-	-0.75%	0.66%	-	-0.68%	0.29%	-	-0.77%	0.52%	-	-0.55%	11.93%	-	-1.13%	1.02%	-	-1.01%	0.83%
0.30	0.3	1.0	-	-1.18%	0.86%	-	-1.14%	0.39%	-	-1.11%	0.71%	-	-2.93%	11.44%	-	-1.03%	1.42%	-	-1.27%	0.86%
0.35	0.3	1.0	-	-0.72%	1.71%	-	-0.80%	1.08%	-	-0.81%	1.54%	-	-2.99%	13.24%	-	-1.13%	1.25%	-	-0.91%	1.27%
0.40	0.3	1.0	-	-1.70%	1.21%	-	-0.90%	1.44%	-	-0.72%	2.23%	-	-4.53%	13.65%	-	-1.10%	1.28%	-	-0.91%	1.37%
0.45	0.3	1.0	-	-1.18%	2.46%	-	-1.18%	1.70%	-	-1.14%	2.32%	-	-3.54%	16.61%	-	-1.34%	1.13%	-	-1.25%	1.05%
0.50	0.3	1.0	-	-0.79%	3.09%	-	-1.13%	2.22%	-	-0.87%	3.14%	-	-2.84%	19.44%	-	-1.40%	1.22%	-	-1.03%	1.37%
0.3	**0.05**	1.0	-	-0.17%	0.50%	-	-0.16%	0.38%	-	-0.25%	0.48%	-	-0.38%	3.23%	-	-0.15%	0.38%	-	-0.21%	0.32%
0.3	**0.10**	1.0	-	-0.41%	0.77%	-	-0.35%	0.64%	-	-0.37%	0.94%	-	-1.56%	4.99%	-	-0.34%	0.69%	-	-0.33%	0.64%
0.3	**0.15**	1.0	-	-0.56%	0.99%	-	-0.45%	0.83%	-	-0.50%	1.15%	-	-1.01%	8.09%	-	-0.43%	1.09%	-	-0.42%	0.95%
0.3	**0.20**	1.0	-	-0.64%	1.15%	-	-0.61%	0.86%	-	-0.67%	1.23%	-	-2.18%	8.93%	-	-0.73%	0.96%	-	-0.73%	0.99%
0.3	**0.25**	1.0	-	-1.09%	0.71%	-	-0.95%	0.59%	-	-0.75%	1.23%	-	-4.19%	8.34%	-	-1.27%	0.84%	-	-0.88%	0.92%
0.3	**0.30**	1.0	-	-1.18%	0.86%	-	-1.14%	0.39%	-	-1.11%	0.71%	-	-2.93%	11.44%	-	-1.03%	1.42%	-	-1.27%	0.86%
0.3	**0.35**	1.0	-	-0.94%	1.11%	-	-0.93%	0.56%	-	-0.77%	1.02%	-	-2.12%	13.44%	-	-1.27%	1.22%	-	-0.71%	1.62%
0.3	**0.40**	1.0	-	-1.20%	0.69%	-	-0.73%	0.53%	-	-0.84%	0.75%	-	-4.63%	11.48%	-	-1.24%	1.47%	-	-0.98%	1.35%
0.3	**0.45**	1.0	-	-0.64%	0.99%	-	-1.22%	0.04%	-	-1.11%	0.38%	-	-4.65%	12.00%	-	-1.74%	1.04%	-	-1.44%	1.17%
0.3	**0.50**	1.0	-	-1.12%	0.61%	-	-0.89%	0.12%	-	-0.84%	0.48%	-	-6.45%	10.57%	-	-1.68%	1.37%	-	-1.37%	1.49%
0.3	0.3	**0.5**	-	-0.71%	-0.64%	-	-0.40%	-0.66%	-	-0.27%	-0.43%	-	-13.14%	10.46%	-	-0.24%	0.21%	-	-1.43%	0.32%
0.3	0.3	**0.6**	-	-1.15%	0.00%	-	-1.04%	-0.54%	-	-0.93%	0.13%	-	-14.96%	13.27%	-	-2.61%	0.30%	-	-2.34%	1.30%
0.3	0.3	**0.7**	-	-0.97%	0.86%	-	-0.70%	0.40%	-	-0.92%	0.47%	-	-12.62%	13.37%	-	-2.31%	1.45%	-	-2.62%	1.40%
0.3	0.3	**0.8**	-	-1.16%	0.84%	-	-0.88%	0.55%	-	-0.72%	1.00%	-	-6.75%	15.27%	-	-1.93%	1.41%	-	-1.65%	1.76%
0.3	0.3	**0.9**	-	-1.10%	0.91%	-	-0.91%	0.54%	-	-1.16%	0.64%	-	-4.47%	13.01%	-	-1.40%	1.34%	-	-1.51%	1.08%
0.3	0.3	**1.0**	-	-1.18%	0.86%	-	-1.14%	0.39%	-	-1.11%	0.71%	-	-2.93%	11.44%	-	-1.03%	1.42%	-	-1.27%	0.86%
0.3	0.3	**1.1**	-	-0.70%	1.19%	-	-0.82%	0.58%	-	-0.85%	0.96%	-	-0.94%	10.99%	-	-0.94%	0.96%	-	-0.66%	1.07%
0.3	0.3	**1.2**	-	-1.27%	0.46%	-	-0.71%	0.59%	-	-0.78%	0.90%	-	-0.84%	9.06%	-	-0.68%	0.85%	-	-0.55%	0.84%
0.3	0.3	**1.3**	-	-0.56%	1.05%	-	-0.59%	0.75%	-	-0.93%	0.64%	-	-0.45%	7.80%	-	-0.38%	0.94%	-	-0.81%	0.27%
0.3	0.3	**1.4**	-	-0.83%	0.66%	-	-0.59%	0.58%	-	-0.68%	0.77%	-	-0.38%	6.62%	-	-0.32%	0.78%	-	-0.40%	0.50%
0.3	0.3	**1.5**	-	-0.79%	0.49%	-	-0.49%	0.65%	-	-0.62%	0.70%	-	0.42%	6.46%	-	-0.41%	0.52%	-	-0.36%	0.42%
0.3	0.3	**1.4**	-	-0.83%	0.66%	-	-0.59%	0.58%	-	-0.68%	0.77%	-	-0.38%	6.62%	-	-0.32%	0.78%	-	-0.40%	0.50%
0.3	0.3	**1.5**	-	-0.79%	0.49%	-	-0.49%	0.65%	-	-0.62%	0.70%	-	0.42%	6.46%	-	-0.41%	0.52%	-	-0.36%	0.42%

fixed setup cost $\boldsymbol{f}_k$. Now, we can revise the reduced model (4) as:

$$\begin{aligned}
\min_{\boldsymbol{I},\boldsymbol{Z}_k} \quad & \kappa(\sum_{k\in[K]} \boldsymbol{f}_k^T \boldsymbol{Z}_k + \boldsymbol{h}^T \boldsymbol{I}) + \sup_{\mathbb{P}\in\mathscr{F}(\boldsymbol{\mu},\boldsymbol{\Sigma})} \mathbb{E}_{\mathbb{P}} \left[g(\boldsymbol{I},\boldsymbol{Z},\tilde{\boldsymbol{d}}) \right] \\
\text{s.t.} \quad & \boldsymbol{I} \leq \sum_{k\in[K]} C_k \boldsymbol{Z}_k, \ \forall k \in [K] \\
& \sum_{k\in[K]} \boldsymbol{Z}_k \leq \mathbf{1}, \\
& \boldsymbol{I} \in \mathbb{R}_+^m, \boldsymbol{Z}_k \in \{0,1\}^m
\end{aligned} \tag{21}$$

The objective function is self-evident. The first constraint requires that the number of stored products cannot exceed warehouse capacity limits, while the second constraint restricts only one warehouse for each location. We can still follow the same analysis in Section 3.2 to reformulate the *sup* problem. And the resulting conic model is still tractable under relaxation, although it involves extra binary variables. One notable thing here is that imposing additional linear constraints on first-stage location-inventory problem does not theoretically increase the difficulty of solving. Therefore, our proposed model is compatible enough to first-stage feasible region as long as it is described by polynomial-time-solvable constraints such as linear constraints, second-order cone constraints, and positive semi-definite constraints.

5.3. Uncertainty in First- and Second-moments

We further introduce uncertainty in first-two moments to our model by utilizing the technique proposed in Delage and Ye (2010). Suppose that mean value $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ themselves are uncertain, and the uncertainty set can be characterized as:

$$(\boldsymbol{\mu},\boldsymbol{\Sigma}) \in \mathscr{D} := \left\{ (\boldsymbol{\mu},\boldsymbol{\Sigma}) \,\middle|\, \begin{array}{l} (\boldsymbol{\mu}-\boldsymbol{\mu}_0)^T \boldsymbol{\Sigma}_0^{-1} (\boldsymbol{\mu}-\boldsymbol{\mu}_0) \leq \gamma_1 \\ \boldsymbol{\Sigma} \preceq \gamma_2 \boldsymbol{\Sigma}_0 + \boldsymbol{\mu}\boldsymbol{\mu}_0^T + \boldsymbol{\mu}_0\boldsymbol{\mu}^T - \boldsymbol{\mu}_0\boldsymbol{\mu}_0^T \end{array} \right\}$$

where $\boldsymbol{\mu}_0$ and $\boldsymbol{\Sigma}_0$ are given vector and matrix, describing an ellipsoid. Parameter $\gamma_1 \geq 0$ and $\gamma_2 \geq 1$ naturally quantify one's confidence in $\boldsymbol{\mu}_0$ and $\boldsymbol{\Sigma}_0$. The first inequality requires $\boldsymbol{\mu}$ lies in an ellipsoid centered at $\boldsymbol{\mu}_0$ within radius γ_1, and the second constraint restricts covariance matrix $\boldsymbol{\Sigma}$. By Schur's Complementary theory, the first inequality can be expressed as a positive semi-definite constraint:

$$\begin{bmatrix} \boldsymbol{\Sigma}_0 & (\boldsymbol{\mu}-\boldsymbol{\mu}_0) \\ (\boldsymbol{\mu}-\boldsymbol{\mu}_0)^T & \gamma_1 \end{bmatrix} \succeq \mathbf{0} \tag{22}$$

The second constraint itself is tractable:

$$\boldsymbol{\Sigma} \preceq \gamma_2 \boldsymbol{\Sigma}_0 + \boldsymbol{\mu}\boldsymbol{\mu}_0^T + \boldsymbol{\mu}_0\boldsymbol{\mu}^T - \boldsymbol{\mu}_0\boldsymbol{\mu}_0^T \tag{23}$$

Therefore, incorporating (22) and (23) into Model (rFOLI) gives the following model:

$$\begin{aligned} \min_{\boldsymbol{I},\boldsymbol{Z}} \quad & \kappa(\boldsymbol{f^T Z} + \boldsymbol{h^T I}) + \max_{(\boldsymbol{\mu},\boldsymbol{\Sigma})\in\mathscr{D}} \sup_{\mathbb{P}\in\mathscr{F}(\boldsymbol{\mu},\boldsymbol{\Sigma})} \mathbb{E}_{\mathbb{P}}\left[g(\boldsymbol{I},\tilde{\boldsymbol{d}})\right] \\ \text{s.t.} \quad & \boldsymbol{I} \le U\boldsymbol{Z}, \\ & \boldsymbol{I}\in\mathbb{R}^m_+, \boldsymbol{Z}\in\{0,1\}^m \end{aligned} \tag{24}$$

Recall that, $\sup_{\mathbb{P}\in\mathscr{F}(\boldsymbol{\mu},\boldsymbol{\Sigma})} \mathbb{E}_{\mathbb{P}}\left[g(\boldsymbol{I},\tilde{\boldsymbol{d}})\right]$ can be reformulated as a maximizing completely positive programming, and it can further combine with the decision $(\boldsymbol{\mu},\boldsymbol{\Sigma})\in\mathscr{D}$. Since $\mathscr{D}$ only involves positive semi-definite constraints, the strong duality obviously still hold between minimizing copositive programming $L_{CO}(\boldsymbol{I})$ and maximizing completely positive programming $L_{CP}(\boldsymbol{I})$. And the remaining analysis is the same.

6. Conclusion

Motivated by a cooperation project with Group G, we studied a fulfillment-oriented location-inventory problem under demand uncertainty. The problem is formulated as a mixed-integer two-stage distributionally robust optimization with an ambiguity set characterized by first-two moments. Utilizing min-cut max-flow theory, we proposed an equivalent reformulation to transfer the model into a mixed-integer conic problem based on copositive cones. We also put forward a dual-variable-based selection criterion to speed up branch and bound process. Extensive numerical studies demonstrate the necessity of incorporating demand correlation when it exists, and a larger coefficient of variation will amplify the impact of correlations. Moreover, Model (CO) is also robust enough and can always achieve the lowest total cost.

Our study also demonstrated the superiority of copositve programming and its advantages in tackling operation management problems. We hope our study could stimulate further research involving copositive programming and its applications in the field of operation management. A potential research direction is considering a more realistic recourse problem, including shipping cost, ordering cost, location-specified penalty cost, etc. Interested readers are also encouraged to explore analytic properties of copositive cone mathematically.

References

Baumol, W. J., Wolfe, P., 1958. A warehouse-location problem. Operations research 6 (2), 252–263.

Bertsimas, D., Delarue, A., Martin, S., 2019a. Optimizing schools' start time and bus routes. Proceedings of the National Academy of Sciences 116 (13), 5943–5948.

Bertsimas, D., Doan, X. V., Natarajan, K., Teo, C.-P., 2010. Models for minimax stochastic linear optimization problems with risk aversion. Mathematics of Operations Research 35 (3), 580–602.

Bertsimas, D., Natarajan, K., Teo, C.-P., 2004. Probabilistic combinatorial optimization: Moments, semidefinite programming, and asymptotic bounds. SIAM Journal on Optimization 15 (1), 185–209.

Bertsimas, D., Natarajan, K., Teo, C.-P., 2006. Persistence in discrete optimization under data uncertainty. Mathematical programming 108 (2-3), 251–274.

Bertsimas, D., Popescu, I., 1999. Optimal inequalities in probability: A convex optimization approach. SIAM J. Optim., to appear.

Bertsimas, D., Popescu, I., 2005. Optimal inequalities in probability theory: A convex optimization approach. SIAM Journal on Optimization 15 (3), 780–804.

Bertsimas, D., Sim, M., Zhang, M., 2019b. Adaptive distributionally robust optimization. Management Science 65 (2), 604–618.

Bomze, I. M., 1998. On standard quadratic optimization problems. Journal of Global Optimization 13 (4), 369–387.

Bomze, I. M., 2012. Copositive optimization–recent developments and applications. European Journal of Operational Research 216 (3), 509–520.

Bomze, I. M., Dür, M., De Klerk, E., Roos, C., Quist, A. J., Terlaky, T., 2000. On copositive programming and standard quadratic optimization problems. Journal of Global Optimization 18 (4), 301–320.

Burer, S., 2009. On the copositive representation of binary and continuous nonconvex quadratic programs. Mathematical Programming 120 (2), 479–495.

Cardoso, S. R., Barbosa-Póvoa, A. P. F., Relvas, S., 2013. Design and planning of supply chains with integration of reverse logistics activities under demand uncertainty. European journal of Operational Research 226 (3), 436–451.

Chen, X., Sim, M., Sun, P., Zhang, J., 2008. A linear decision-based approximation approach to stochastic programming. Operations Research 56 (2), 344–357.

Chen, Z., Sim, M., Xiong, P., 2020. Robust stochastic optimization made easy with rsome. Management Science.

Chen, Z., Sim, M., Xu, H., 2019. Distributionally robust optimization with infinitely constrained ambiguity sets. Operations Research 67 (5), 1328–1344.

Chou, M. C., Chua, G. A., Teo, C.-P., Zheng, H., 2011. Process flexibility revisited: The graph expander and its applications. Operations research 59 (5), 1090–1105.

Daskin, M. S., Coullard, C. R., Shen, Z.-J. M., 2002. An inventory-location model: Formulation, solution algorithm and computational results. Annals of operations research 110 (1-4), 83–106.

De Klerk, E., Pasechnik, D. V., 2002. Approximation of the stability number of a graph via copositive programming. SIAM Journal on Optimization 12 (4), 875–892.

Delage, E., Ye, Y., 2010. Distributionally robust optimization under moment uncertainty with application to data-driven problems. Operations research 58 (3), 595–612.

Dickinson, P. J., 2013. The copositive cone, the completely positive cone and their generalisations. Citeseer.

Farahani, R. Z., Miandoabchi, E., Szeto, W. Y., Rashidi, H., 2013. A review of urban transportation network design problems. European Journal of Operational Research 229 (2), 281–302.

Gabor, A. F., van Ommeren, J. C., 2006. An approximation algorithm for a facility location problem with stochastic demands and inventories. Operations research letters 34 (3), 257–263.

Goh, J., Sim, M., 2010. Distributionally robust optimization and its tractable approximations. Operations research 58 (4-part-1), 902–917.

Gourtani, A., Nguyen, T.-D., Xu, H., 2020. A distributionally robust optimization approach for two-stage facility location problems. EURO Journal on Computational Optimization, 1–32.

Javid, A. A., Azad, N., 2010. Incorporating location, routing and inventory decisions in supply chain network design. Transportation Research Part E: Logistics and Transportation Review 46 (5), 582–597.

Jordan, W. C., Graves, S. C., 1995. Principles on the benefits of manufacturing process flexibility. Management Science 41 (4), 577–594.

Kong, Q., Lee, C.-Y., Teo, C.-P., Zheng, Z., 2013. Scheduling arrivals to a stochastic service delivery system using copositive cones. Operations research 61 (3), 711–726.

Kong, Q., Li, S., Liu, N., Teo, C.-P., Yan, Z., 2020. Appointment scheduling under time-dependent patient no-show behavior. Management Science.

Lasserre, J. B., 2011. A new look at nonnegativity on closed sets and polynomial optimization. SIAM Journal on Optimization 21 (3), 864–885.

Li, X., Natarajan, K., Teo, C.-P., Zheng, Z., 2014. Distributionally robust mixed integer linear programs: Persistency models with applications. European Journal of Operational Research 233 (3), 459–473.

Li, Y., Shu, J., Song, M., Zhang, J., Zheng, H., 2017. Multisourcing supply network design: two-stage chance-constrained model, tractable approximations, and computational results. INFORMS Journal on Computing 29 (2), 287–300.

Liu, K., Li, Q., Zhang, Z.-H., 2019. Distributionally robust optimization of an emergency medical service station location and sizing problem with joint chance constraints. Transportation research part B: methodological 119, 79–101.

Lovász, L., Schrijver, A., 1991. Cones of matrices and set-functions and 0–1 optimization. SIAM journal on optimization 1 (2), 166–190.

Mak, H.-Y., Rong, Y., Shen, Z.-J. M., 2013. Infrastructure planning for electric vehicles with battery swapping. Management Science 59 (7), 1557–1575.

Mak, H.-Y., Rong, Y., Zhang, J., 2015. Appointment scheduling with limited distributional information. Management Science 61 (2), 316–334.

Mak, H.-Y., Shen, Z.-J. M., 2009. A two-echelon inventory-location problem with service considerations. Naval Research Logistics (NRL) 56 (8), 730–744.

Mete, H. O., Zabinsky, Z. B., 2010. Stochastic optimization of medical supply location and distribution in disaster management. International Journal of Production Economics 126 (1), 76–84.

Miranda, P. A., Garrido, R. A., 2004. Incorporating inventory control decisions into a strategic distribution network design model with stochastic demand. Transportation Research Part E: Logistics and Transportation Review 40 (3), 183–207.

Mishra, V. K., Natarajan, K., Tao, H., Teo, C.-P., 2012. Choice prediction with semidefinite optimization when utilities are correlated. IEEE Transactions on Automatic Control 57 (10), 2450–2463.

Murty, K. G., Kabadi, S. N., 1985. Some np-complete problems in quadratic and nonlinear programming. Tech. rep.

Nakao, H., Shen, S., Chen, Z., 2017. Network design in scarce data environment using moment-based distributionally robust optimization. Computers & operations research 88, 44–57.

Natarajan, K., Teo, C. P., Zheng, Z., 2011. Mixed 0-1 linear programs under objective uncertainty: A completely positive representation. Operations research 59 (3), 713–728.

Ni, W., Shu, J., Song, M., 2018. Location and emergency inventory pre-positioning for disaster response operations: Min-max robust model and a case study of yushu earthquake. Production and Operations Management 27 (1), 160–183.

Popescu, I., 2007. Robust mean-covariance solutions for stochastic optimization. Operations Research 55 (1), 98–112.

Saif, A., Delage, E., 2020. Data-driven distributionally robust capacitated facility location problem. European Journal of Operational Research.

Scarf, H., 1958. A min-max solution of an inventory problem. Studies in the mathematical theory of inventory and production.

Shen, Z.-J. M., Qi, L., 2007. Incorporating inventory and routing costs in strategic location models. European journal of operational research 179 (2), 372–389.

Shu, J., Ma, Q., Li, S., 2010. Integrated location and two-echelon inventory network design under uncertainty. Annals of operations research 181 (1), 233–247.

Wiesemann, W., Kuhn, D., Sim, M., 2014. Distributionally robust convex optimization. Operations Research 62 (6), 1358–1376.

Yan, Z., Gao, S. Y., Teo, C. P., 2018. On the design of sparse but efficient structures in operations. Management Science 64 (7), 3421–3445.

Zhang, P., Liu, Y., Yang, G., Zhang, G., 2020. A multi-objective distributionally robust model for sustainable last mile relief network design problem. Annals of Operations Research, 1–42.

Appendix A. Proofs

Appendix A.1. Proof for Proposition 1

Proof. We prove this proposition by showing $g(\boldsymbol{I}, \boldsymbol{Z}, \tilde{\boldsymbol{d}}) \geq g(\boldsymbol{I}, \tilde{\boldsymbol{d}})$ and $g(\boldsymbol{I}, \boldsymbol{Z}, \tilde{\boldsymbol{d}}) \leq g(\boldsymbol{I}, \tilde{\boldsymbol{d}})$ both hold. For clearer notations, we rewrite both problems as follows:

$$\begin{aligned} g(\boldsymbol{I}, \boldsymbol{Z}, \tilde{\boldsymbol{d}}) = \quad \min_{\boldsymbol{x}} \quad & \sum_{j\in[n]} \tilde{d}_j - \sum_{(i,j)\in\mathcal{E}(\boldsymbol{Z})} x_{ij} \\ s.t. \quad & \sum_{i:(i,j)\in\mathcal{E}(\boldsymbol{Z})} x_{ij} \leq \tilde{d}_j, \forall j \in [n] \\ & \sum_{j:(i,j)\in\mathcal{E}(\boldsymbol{Z})} x_{ij} \leq I_i, \forall i \in [m] \\ & x_{ij} \geq 0, \forall (i,j) \in \mathcal{E}(\boldsymbol{Z}) \end{aligned} \tag{A.1}$$

Define a indices set $\boldsymbol{Z}^+$ containing all location indices that a warehouse is built, i.e. $\boldsymbol{Z}^+ := \{i \,\big|\, Z_i = 1,\ \forall i \in [m]\}$.

$$g(\boldsymbol{I}, \tilde{\boldsymbol{d}}) = \quad \min_{\boldsymbol{x}} \quad \sum_{j\in[n]} \tilde{d}_j - \sum_{(i,j)\in\mathcal{E}(\boldsymbol{Z})} x_{ij} - \sum_{(i,j)\notin\mathcal{E}(\boldsymbol{Z})} x_{ij} \tag{A.2}$$

$$s.t. \quad \sum_{i:(i,j)\in\mathcal{E}(\boldsymbol{Z})} x_{ij} + \sum_{i:(i,j)\notin\mathcal{E}(\boldsymbol{Z})} x_{ij} \leq \tilde{d}_j, \forall j \in [n] \tag{A.3}$$

$$\sum_{j:(i,j)\in\mathcal{E}} x_{ij} \leq I_i, \forall i \in \boldsymbol{Z}^+$$

$$\sum_{j:(i,j)\in\mathcal{E}} x_{ij} \leq 0, \forall i \notin \boldsymbol{Z}^+ \tag{A.4}$$

$$x_{ij} \geq 0, \forall (i,j) \in \mathcal{E}$$

In $g(\boldsymbol{I}, \tilde{\boldsymbol{d}})$, the right-hand-side coefficient of the third constraint is zero since only establishing a warehouse is the prerequisite to store products there.

1. $g(\boldsymbol{I}, \boldsymbol{Z}, \tilde{\boldsymbol{d}}) \geq g(\boldsymbol{I}, \tilde{\boldsymbol{d}})$

   It is easy to see that the optimal solution $\boldsymbol{x}^*$ to (A.1) is always a feasible solution to (A.2) by setting other $x_{ij} = 0, (i,j) \notin \mathcal{E}(\boldsymbol{Z})$. Therefore, $g(\boldsymbol{I}, \boldsymbol{Z}, \tilde{\boldsymbol{d}}) \geq g(\boldsymbol{I}, \tilde{\boldsymbol{d}})$ holds.

2. $g(\boldsymbol{I}, \boldsymbol{Z}, \tilde{\boldsymbol{d}}) \leq g(\boldsymbol{I}, \tilde{\boldsymbol{d}})$

   Now, suppose $\boldsymbol{x}^*$ is the optimal solution to (A.2). We split the solution into two parts according to whether $i \in \boldsymbol{Z}^+$, i.e. $\boldsymbol{x}^* = (\boldsymbol{x}^+, \boldsymbol{x}^0)$ where $\boldsymbol{x}^+$ contains allocation decision from warehouses while $\boldsymbol{x}^0$ contains allocation decisions from places no warehouse is built. Because of Constraint (A.4), we have $\boldsymbol{x}^0 = \boldsymbol{0}$. It is easy to check $\boldsymbol{x}^+$ is a feasible solution to (A.1). Therefore, $g(\boldsymbol{I}, \boldsymbol{Z}, \tilde{\boldsymbol{d}}) \leq g(\boldsymbol{I}, \tilde{\boldsymbol{d}})$ always holds.

Combining above two statements, we finished the proof. □

Appendix A.2. Proof for Proposition 2

Proof. We prove Proposition 2 by equivalently reformulating (5) into a minimizing problem with only one constraint. And conducting optimization on the unique constraint has been proved to be NP-hard in Bertsimas et al. (2010).

Firstly, we take dual on (5) as follows:

$$L(\boldsymbol{I}) = \sup_{\mathbb{P}\in\mathscr{F}(\boldsymbol{\mu},\boldsymbol{\Sigma})} \mathbb{E}_{\mathbb{P}}[g(\boldsymbol{I},\tilde{\boldsymbol{d}})] = \begin{aligned} \sup_{m(\tilde{\boldsymbol{d}})} & \int_{\mathbb{R}^n_+} g(\boldsymbol{I},\tilde{\boldsymbol{d}})\ m(\tilde{\boldsymbol{d}}) \\ s.t. & \int_{\mathbb{R}^n_+} 1\ m(\tilde{\boldsymbol{d}}) = 1 \\ & \int_{\mathbb{R}^n_+} \tilde{\boldsymbol{d}}\ m(\tilde{\boldsymbol{d}}) = \boldsymbol{\mu} \\ & \int_{\mathbf{R}^n_+} \tilde{\boldsymbol{d}}\tilde{\boldsymbol{d}}^T\ m(\tilde{\boldsymbol{d}}) = \boldsymbol{\Sigma} \\ & m(\tilde{\boldsymbol{d}}) \in \mathscr{M}_+(\mathbb{R}^n_+) \end{aligned}$$

where $m(\tilde{\boldsymbol{d}})$ is naturally a probability measure on $\mathbb{R}^n_+$. The dual problem is:

$$\begin{aligned} & \inf_{s,\boldsymbol{t},\boldsymbol{\eta}} s + \boldsymbol{\mu}^T\boldsymbol{t} + \boldsymbol{\Sigma}\bullet\boldsymbol{\eta} \\ & s.t. \min_{(\hat{\boldsymbol{u}},\boldsymbol{v})\in\hat{\mathcal{L}_{lp}},\tilde{\boldsymbol{d}}\in\mathbb{R}^n_+} \tilde{\boldsymbol{d}}\boldsymbol{\eta}\tilde{\boldsymbol{d}} + \tilde{\boldsymbol{d}}^T(\boldsymbol{t}-\hat{\boldsymbol{u}}) + \boldsymbol{I}^T\boldsymbol{v} + s \geq 0 \end{aligned} \tag{A.5}$$

Obviously, strong duality holds between the maximizing problem and the dual minimizing problem. According to Bertsimas et al. (2010) Theorem 3.1, the separation problem, i.e. for given $\boldsymbol{t}$, $\boldsymbol{I}$, and s, check if the constraint (A.5) is satisfied and if not, find a feasible $(\hat{\boldsymbol{u}},\boldsymbol{v}) \in \hat{\mathcal{L}_{lp}}, \tilde{\boldsymbol{d}} \in \mathbb{R}^n_+$ satisfying (A.5), is NP-Complete. Because of the equivalence of separation and optimization, the minimizing problem is NP-hard, as well as evaluating the value of $L(\boldsymbol{I})$. $\square$