Fulfillment-Oriented Location-Inventory Problem with Demand Information Uncertainty

Abstract

Motivated by a cooperation project with Group G, we propose a fulfillment-oriented location-inventory problem under demand information uncertainty, where only the 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 as a mixed-integer conic problem based on copositive cones, and it is tractable with positive semidefinite relaxation. We design an interpretable dual-variable-based heuristic to accelerate the branch-and-bound solution process. Extensive numerical studies demonstrate that compared to a sample-average-approximation model and a robust model with only mean and variance information, our proposed model can achieve a higher fulfillment rate and a lower total cost.

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

1. Introduction

Location-inventory problems are a classic elementary stream of problems that occur in a wide range of applications, including supply chain management, disaster relief, last-mile delivery, and retailing. Daskin et al. (2002) noted that a typical location-inventory problem assumes predetermined potential locations of suppliers and aims to determine the optimal inventory levels and locations of distribution centers (DCs), as well as the deployed inventory allocated to customers. In this kind of problem, the location decision is a strategic decision, typically made for a planning horizon of two to five years. Additionally, 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 quantity of items can have substantial positive economic effects, such as decreased operational cost and increased demand fulfillment rate. The intrinsic characteristic of location-inventory problems of combining strategic supply chain decision making and operational inventory management decisions, makes these problems attractive to the academic community.

With the rapid development of last-mile logistics and e-commerce, the location-inventory problem has received considerable attention from practitioners in recent years in the field of retailing. The retailing industry has undergone significant reforms in recent years. Retailing companies manage to provide customers with a faster and high-level service by deploying 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 orders from customers within a three-kilometer radius. In the U.S., Walmart adopted a similar strategy called "dark store" to quickly fulfill its orders.

Our study is motivated by cooperation with Group G, a leading Chinese retail company selling top brand athletic apparel for babies. Let us take Shanghai, China as an example. Group G has more than 40 stores in Shanghai and sells approximately two thousand stock keeping units (SKUs) per season, including shoes, apparel, and accessories. Before 2018, inventory decisions were made centrally every week by the sales department, and a regional warehouse was responsible for executing these decisions. Although professional, sales experts still make suboptimal decisions occasionally, leading to stock shortages within a given week and loss of sales opportunities. To avoid stock shortages, capture more sales opportunities, and provide quick response ability, Group G has 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 the coverage radius.

Our study's main aims are to select in-city UW locations from a predetermined set of potential locations and to make decisions on weekly inventory levels such that the UWs can fulfill as much of the demand as possible within a given week. The establishment of a new UW entails a location-specified setup cost, and deploying one unit of product entails a location-specific holding cost. These two decisions are referred to as first-stage decisions. Then, during the week, every replenishment request from stores is fully fulfilled as long as the UWs have sufficient capacity. Any unmet demand is counted as a one-unit penalty cost. Although replenishment requests arrive sequentially in practice, it is reasonable to regard the successively arriving demand of each store within a week as an aggregate demand. Therefore, the second-stage problem is a resource allocation problem of minimizing total unmet demand, given the deployed capacity and realized demand. Note that the fulfillment-orientation feature in our problem is the issue of greatest concern for Group G. For simplicity of notations, we temporarily simplify the realistic problem and consider only one selling week and one product. This simplification can then easily be extended to a more realistic situation.

Unfortunately, even though the problem has been simplified, it remains difficult to solve 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 difficult to solve efficiently, early location-inventory studies have proposed various heuristics to which we can refer (Berman et al. (2012), Lim et al. (2013), Nakao et al. (2017), Zhang et al. (2020)). The second issue is more challenging to address. Since each selling season lasts approximately twelve to sixteen weeks, week-level historical sales data are limited. It is impossible to accurately estimate the distribution of demand.

Traditionally, two main approaches have been used to address uncertainty in optimization problems: stochastic programming and robust optimization. In stochastic programming, the uncertain parameters are represented as random variables with a known probability distribution. By contrast, 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 our scenario due to the scarcity of historical data, and the latter is overly conservative.

We adopt a distributionally robust optimization (DRO) technique for our problem. The aim of DRO is to minimize the worst-case expected total cost over a set of feasible demand distributions. Therefore, DRO belongs in an intermediate position between stochastic programming and robust optimization, overcoming the deficiencies of both approaches. Specifically, we adopt moments-based DRO, in which the set of distributions is characterized by the moments of random variables, e.g., the mean and covariance matrix. The moments are easy to estimate from a statistically sufficiently large dataset, such as a dataset containing 30 samples. The covariance matrix of demand offers the possibility to capture correlations between different demand nodes. Notably, the demand correlation is greatest in the case of offline retailing. For example, stores in the same city are likely to achieve lower sales volumes on a rainy day simultaneously. Although DRO address some of the deficiencies of stochastic programming and robust optimization, computational intractability remains another issue. To address this problem, we reformulate our model as a conic programming problem described by copositive cone constraints and exploit semidefinite approximation to approximate copositive cones for computational convenience.

Our main contributions are threefold:

1. We introduce demand correlations to location-inventory problems by specifying the demand covariance matrix. This problem is described as a mixed-integer two-stage distributionally robust optimization model (see Model (FOLI)).

2. We reformulate the original model as a copositive programming problem by exploiting min-cut max-flow theory (see Proposition 3). The equivalent reformulation does not exist in general.

3. We propose an interpretable branching method to accelerate the branch-and-bound solving process, thereby cutting the computational time in half (see Algorithm 1).

The rest of this paper is organized as follows. We first present a literature review of location-inventory problems, DRO, and copositive programming in Section 2. We formally propose our model in Section 3, where we also propose the reformulated model and an algorithm to accelerate the solution process. In Section 4, we conduct numerous numerical studies to illustrate the advantages of the proposed models. In Section 5, several extensions are proposed to demonstrate our model's applicability to more complex situations. Finally, in Section 6, we conclude our study.

Notations. Throughout this paper, we use lowercase ***bold*** and UPPERCASE ***bold*** characters to denote vectors and matrices, respectively. Corresponding normal characters denote component-wise elements. Variables with tilde symbols, such as $\tilde{\boldsymbol{d}}$, represent random variables. We use $\boldsymbol{A} \succeq \mathbf{0}$ to indicate that matrix $\boldsymbol{A}$ is positive semidefinite. $\boldsymbol{A} \succeq_{co} \mathbf{0}$ and $\boldsymbol{A} \succeq_{cp} \mathbf{0}$ require that $\boldsymbol{A}$ belongs to a copositive or completely positive cone. $\mathbb{R}^n_+$ denotes an n-dimensional nonnegative real number, and $[n]$ represents the set $\{1, 2, 3, \ldots, n\}$.

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 the problem. 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)), and reverse logistics (Cardoso et al. (2013)). The most popular topic is to address uncertain demand: most of the mentioned works consider 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, the approach risks placing the decision maker in a disadvantageous situation of misspecification, which could result in an inappropriately designed network and cause massive economic loss.

To handle misspecification, DRO, an emerging modeling tool, is utilized to address the classic location-inventory-related problem when historical data are 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 as 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 via semi-infinite programming and approximated 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 the Wasserstein ball constructed from a small set of empirical samples. They also designed a column-generation algorithm to solve the problem exactly. The above three works considered potential correlations between demand variables, but inventory-level decisions were not explicitly accounted for.

Mak et al. (2013) considered electric vehicle charging station and battery deployment with a distributionally robust model, where the mean and variance of demand were 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 the locations, 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 was omitted.

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

2.2. Moment-based Distributionally Robust Optimization and Copositive Programming

Moment-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 subject to an ambiguity set constrained by a known support set and moment information. The past decade witnessed explosive growth in research on this topic after Bertsimas and Popescu (1999) connected the moment problem and semidefinite optimization (see Bertsimas and Popescu (2005), Bertsimas et al. (2006), Popescu (2007), Bertsimas et al. (2010), Mishra et al. (2012)). Most of works assume that the known support set is the real number. However, in practical problems, the support set should be restricted to a nonnegative orthant. For example, demand should always be nonnegative, and service time in clinics should be greater than zero. Natarajan et al. (2011) showed that, given the mean and covariance matrix, the problem of computing the worst-case expected value can be reformulated as a conic problem based on completely positive cones. This equivalent reformulation sparked extensive applications based on a completely positive cone or its dual cone, i.e., copositive cone.

An 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 an $n \times n$ symmetric matrix. For $\boldsymbol{A} \in \mathcal{CO}_n$, we write $\boldsymbol{A} \succeq_{co} \boldsymbol{0}$. A completely positive cone is the dual cone of a copositive cone (Dickinson (2013)) and is defined as

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

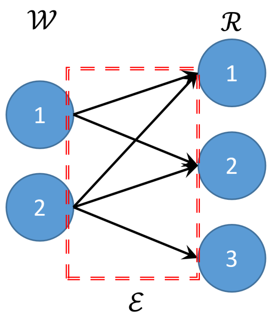
A programming problem with linear constraints on the copositive cone or completely positive cone is called copositive programming or completely positive programming, respectively. Recently, copositive programming has attracted considerable attention in the optimization community as a powerful modeling technique. Many NP-hard problems have been reformulated via copositive programming or completely positive programming. For example, the well-known maximum stable set problem (Rebennack et al. (2012)) has been proved to be equivalent to a copositive program by modifying the feasible region of the theta-number problem from a semidefinite cone to a copositive cone (De Klerk and Pasechnik (2002)). 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 a copositive cone. Additionally, Natarajan et al. (2011) introduced copositive programming into mixed 0-1 linear programs under objective uncertainty.

Although the original NP-hard problem could be reformulated as a copositive problem, the difficulty of optimization is not 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 cones. Complete knowledge of the copositive cone could be used to solve many optimization problems more efficiently. Natarajan et al. (2011) proved that under mild conditions, completely positive programming problems could be solved exactly via semidefinite programming. 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 cones is the "doubly nonnegative cone", defined as $\{\boldsymbol{A} \,|\, \boldsymbol{A} \succeq \boldsymbol{0}, \boldsymbol{A} \geq \boldsymbol{0}\}$. Moreover, $\{\boldsymbol{A} \,|\, \boldsymbol{A} = \boldsymbol{A}_1 + \boldsymbol{A}_2, \boldsymbol{A}_1 \succeq \boldsymbol{0}, \boldsymbol{A}_2 \geq \boldsymbol{0}\}$ provides an inner approximation to copositive cones. In our numerical study, we use this inner approximation to approximate copositive cones.

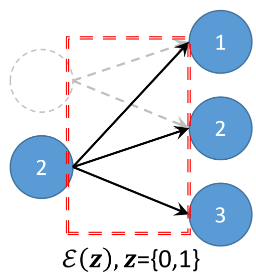
In recent years, researchers have found that many operation management problems can be properly handled via copositive programming and its approximation techniques. For example, Natarajan et al. (2011) applied copositive programming to a project management problem to estimate the expected completion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

3. Fulfillment-Oriented Location-Inventory Problem

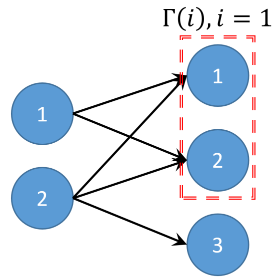
Consider a fulfillment-oriented location-inventory network design problem with a set of potential supply nodes denoted by $\mathcal{W} = \{1, 2, \ldots, m\}$ and a set of demand nodes denoted by $\mathcal{R} = \{1, 2, \ldots, n\}$. The set of road links (or network structure) is denoted by $\mathcal{E}$ (see Figure 1a), and its cardinality is $|\mathcal{E}| = r$. In the first stage, the decision maker designs the accessible network $\mathcal{E}(\boldsymbol{z})$ by selecting a subset of potential supply nodes to build warehouses and deploys an appropriate number of products at location i. Mathematically, we define the binary decision variable $\boldsymbol{z} = (z_1, z_2 \ldots, z_m)$ to indicate whether location i is selected ($z_i = 1$) or not ($z_i = 0$). An example is shown in Figure 1b. Building a warehouse at location i entails a fixed setup cost f_i, and $\boldsymbol{f} = (f_1, f_2, \ldots, f_m)$. We define another decision variable $\boldsymbol{y} = (y_1, y_2, \ldots, y_m)$ to represent the inventory levels at supply nodes. Each deployed product at supply node i entails a fixed holding cost h_i, and we denote $\boldsymbol{h} = (h_1, h_2, \ldots, h_m)$. Additionally, $\Gamma(i)$ is the set of adjacent nodes for location i when the network is $\mathcal{E}$ (see Figure 1c). Interchangeably, $\Gamma(i) = \{j : (i, j) \in \mathcal{E}\}$. We use $\Gamma_{\boldsymbol{z}}(i)$ to emphasize the set of i's adjacent nodes when $\boldsymbol{z}$ is explicitly given. Warehouse $i \in \mathcal{W}$ can deliver products to all adjacent demand nodes $\Gamma(i)$ via 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 demand, i.e., only the finite first



(a) Links: $\mathcal{E}$



(b) Links: $\mathcal{E}(\boldsymbol{z})$



(c) Adjacent Nodes: $\Gamma(i)$

Figure 1: Notation Illustration

moment $\boldsymbol{\mu}$, finite second moment $\boldsymbol{\Sigma}$, and support set $\mathbb{R}_+^n$ are known.

In the second stage, after demand $\tilde{\boldsymbol{d}} = (\tilde{d}_1, \tilde{d}_2, \ldots, \tilde{d}_n)$ is realized, the decision maker can allocate predeployed items through network $\mathcal{E}(\boldsymbol{z})$ to satisfy as much of the demand as possible (or equivalently to minimize unmet demand). Since we focus on a fulfillment-oriented scenario, each unit of unmet demand will trigger a one-unit penalty cost. The ultimate objective is to minimize the total cost, including the setup cost, holding cost and the worst-case expected second-stage penalty cost. We temporarily assume there is only one kind of product: we consider multiple products in Section 5.1.

3.1. Model

Now, we are ready to model the fulfillment-oriented location-inventory (FOLI) problem under demand distribution uncertainty. The formulation can be expressed as follows:

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

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 (FOLI), $\boldsymbol{y}$ and $\boldsymbol{z}$ are the first-stage decision variables, $\kappa \geq 0$ is a risk attitude parameter that balances the costs of the two stages, and U is a sufficiently large positive number. The first constraint guarantees that materials can be stored at locations where a warehouse is established; the remaining constraints are standard nonnegative and binary constraints. $g(\boldsymbol{I}, \boldsymbol{z}, \tilde{\boldsymbol{d}})$, the penalty cost of unmet demand, is obtained by solving a classic allocation problem after demand $\tilde{\boldsymbol{d}}$ is realized. Note that, without loss of generality, the unit penalty cost of each unit of unmet demand is one. Therefore, given $\boldsymbol{y}$, $\boldsymbol{z}$ and $\tilde{\boldsymbol{d}}$, the second-stage problem can be defined as:

$$g(\boldsymbol{y}, \boldsymbol{z}, \tilde{\boldsymbol{d}}) = \min_{\boldsymbol{t} \in \Omega(\boldsymbol{y}, \boldsymbol{z})} \sum_{j \in [n]} \tilde{d}_j - \sum_{(i,j) \in \mathcal{E}(\boldsymbol{z})} t_{ij} \tag{1}$$

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

$$\Omega(\boldsymbol{y}, \boldsymbol{z}) := \{\boldsymbol{t} \in \mathbb{R}_+^{|\mathcal{E}(\boldsymbol{z})|} | \sum_{i:(i,j) \in \mathcal{E}(\boldsymbol{z})} t_{ij} \leq \tilde{d}_j, \forall j \in [n]; \sum_{j:(i,j) \in \mathcal{E}(\boldsymbol{z})} t_{ij} \leq y_i, \forall i \in [m]\}$$

, $\mathcal{E}(\boldsymbol{z})$ is the accessible roads based on decision $\boldsymbol{z}$, and t_{ij} is the number of items delivered from location i to demand node j. The first constraint of $\Omega(\boldsymbol{y}, \boldsymbol{z})$ ensures that the total number of items delivered to demand node j does not exceed the demand, and the second constraint is a capacity constraint for each warehouse i.

Clearly, Model (1) depends on both $\boldsymbol{y}$ and $\boldsymbol{z}$, and the accessible network varies under different $\boldsymbol{z}$, which prevents us from solving (FOLI). Fortunately, (1) avoids the dependency on $\boldsymbol{z}$ at the cost of lifting the dimension of the feasible region from $|\mathcal{E}(\boldsymbol{z})|$ to r.

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

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

where

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

Proof. See Appendix A.1 □

According to Proposition 1, instead of considering $(\boldsymbol{y}, \boldsymbol{z})$, we need to take only $\boldsymbol{y}$ into consideration. Removing $\boldsymbol{z}$ does not harm the optimality but comes at the cost of increasing the dimension of the recourse decision variables $\boldsymbol{t}$. Surprisingly, additional variables make further reformulation easier and provide greater insight into the selection of locations for warehouses, accelerating the solution process. We further explain this result in Section 3.3.

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

$$
\begin{aligned}
(rFOLI) \quad \min_{\boldsymbol{y},\boldsymbol{z}} \quad & \kappa(\boldsymbol{f}^T\boldsymbol{z} + \boldsymbol{h}^T\boldsymbol{y}) + \sup_{\mathbb{P}\in\mathscr{F}(\boldsymbol{\mu},\boldsymbol{\Sigma})} \mathbb{E}_{\mathbb{P}}\left[g(\boldsymbol{y}, \tilde{\boldsymbol{d}})\right] \\
\text{s.t.} \quad & \boldsymbol{y} \leq U\boldsymbol{z}, \\
& \boldsymbol{y} \in \mathbb{R}^m_+, \boldsymbol{z} \in \{0,1\}^m
\end{aligned} \tag{3}
$$

With the more tractable recourse problem, we next equivalently reformulate Model (3) as a mixed-integer copositive programming problem.

3.2. Reformulation

Clearly, the difficulty of rFOLI comes from evaluating the worst-case expected second-stage unmet demand. Denote the worst-case expected second-stage problem as

$$
L(\boldsymbol{y}) := \sup_{\mathbb{P}\in\mathscr{F}(\boldsymbol{\mu},\boldsymbol{\Sigma})} \mathbb{E}_{\mathbb{P}}\left[g(\boldsymbol{y}, \tilde{\boldsymbol{d}})\right] \tag{4}
$$

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

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

The first equation is exactly the unmet demand minimization problem after $\tilde{\boldsymbol{d}}$ is realized. The second equation is obtained by simply exchanging the minimizing operator and negative symbol. The third equation strictly follows the classic 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 the first term and the minimization problem in the third equation, where $\mathbf{1}$ is a vector with all elements equal to 1. The last equation comes from exchanging the maximizing operator and the negative symbol. Additionally, from the perspective of stochastic programming, the feasible region of $\mathcal{L}$ is "complete recourse", so the region is always nonempty.

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

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

Therefore, $g(\boldsymbol{y}, \tilde{\boldsymbol{d}})$ is equivalent among the following formulations:

$$
\begin{aligned}
g(\boldsymbol{y}, \tilde{\boldsymbol{d}}) &= \max_{(\boldsymbol{u}, \boldsymbol{v}) \in \mathcal{L}} \tilde{\boldsymbol{d}}^T (\mathbf{1} - \boldsymbol{u}) - \boldsymbol{y}^T \boldsymbol{v} \\
&= \max_{(\hat{\boldsymbol{u}}, \boldsymbol{v}) \in \hat{\mathcal{L}}} \tilde{\boldsymbol{d}}^T \hat{\boldsymbol{u}} - \boldsymbol{y}^T \boldsymbol{v} \\
&= \max_{(\hat{\boldsymbol{u}}, \boldsymbol{v}) \in \hat{\mathcal{L}}_{lp} \cap \{0,1\}^{n+m}} \tilde{\boldsymbol{d}}^T \hat{\boldsymbol{u}} - \boldsymbol{y}^T \boldsymbol{v}
\end{aligned} \tag{5}
$$

Unfortunately, although the feasible region of $g(\boldsymbol{y}, \tilde{\boldsymbol{d}})$ is fixed and not related to the first-stage decision $\boldsymbol{y}$, the difficulty of evaluating (4) is not reduced. Indeed, Proposition 2 shows that problem (4) is at least NP-hard.

Proposition 2. *Given $g(\boldsymbol{y}, \tilde{\boldsymbol{d}})$ in the form of* (5)*, calculating the solution to problem* (4) *is at least NP-hard.*

Proof. See Appendix A.2 □

While the problem is at least NP-hard, we can reformulate it as a conic programming problem and then solve it via positive semi-definite relaxation.

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

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

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

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

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

Now, several key assumptions in Natarajan et al. (2011) can be shown to always hold in our rFOLI problem, and we formally summarize these assumptions as follows:

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

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

The definition of $\hat{\mathcal{L}}_{lp}^{\dagger}$ naturally leads to these two conclusions. Then, we can reformulate problem rFOLI into a conic problem CO under a mild condition.

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

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

Proposition 3. *Given that 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, problem (rFOLI) can be equivalently reformulated as a copositive-cone-based mixed-integer conic problem CO.

Proof. Before presenting the detailed proof, we concisely summarize the roadmap to prove this proposition. We first transform the second-stage problem $L(\boldsymbol{y})$ into a completely positive conic maximization problem according to Theorem 3.3 in Natarajan et al. (2011). Then, we take the dual to obtain a minimization problem on the copositive cone. We further show that strong duality holds between the maximization and the minimization problems. Finally, we combine the second-stage minimization problem with the first-stage problem to obtain CO.

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

Before we transform $L(\boldsymbol{y})$ 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 Theorem 3.3 in Natarajan et al. (2011), we can equivalently reformulate $L(\boldsymbol{y})$ to a completely positive program $L_{CP}(\boldsymbol{y})$ as follows:

$$
\begin{aligned}
L(\boldsymbol{y}) = \quad L_{CP}(\boldsymbol{y}) = \quad & \max_{\boldsymbol{p},\boldsymbol{Y},\boldsymbol{X}} \begin{bmatrix} \boldsymbol{E}^{[n]} \\ \boldsymbol{0} \end{bmatrix} \bullet \boldsymbol{Y} + \begin{bmatrix} \boldsymbol{0}_{n\times 1} \\ -\boldsymbol{y}_{m\times 1} \\ \boldsymbol{0}_{M\times 1} \end{bmatrix}^T \boldsymbol{p} \\
s.t. \quad & \boldsymbol{a}_i^T \boldsymbol{p} = b_i, \forall i \in [M] && (6)\\
& (\boldsymbol{a}_i \boldsymbol{a}_i^T) \bullet \boldsymbol{X} = b_i^2, \forall i \in [M] && (7)\\
& \boldsymbol{E}_{(j,j)} \bullet \boldsymbol{X} - \boldsymbol{e}_{(j)}^T \boldsymbol{p} = 0, \forall j \in [N] && (8)\\
& \begin{pmatrix} 1 & \boldsymbol{\mu}^T & \boldsymbol{p}^T \\ \boldsymbol{\mu} & \boldsymbol{\Sigma} & \boldsymbol{Y}^T \\ \boldsymbol{p} & \boldsymbol{Y} & \boldsymbol{X} \end{pmatrix} \succeq_{cp} \boldsymbol{0} && (9)
\end{aligned}
$$

where $\boldsymbol{p} \in \mathbb{R}_+^N$, $\boldsymbol{Y} \in \mathbb{R}_+^{N\times n}$, and $\boldsymbol{X} \in \mathbb{R}_+^{N\times N}$ are decision variables. We use superscript $^{[k]}$ to indicate a k-by-k matrix variable. $\boldsymbol{E}^{[n]}$ is an n-dimensional identity matrix. Additionally, we use $\boldsymbol{E}_{(j,j)}$ to represent a square matrix with the (j, j)-position element equal to 1. Furthermore, $\boldsymbol{e}_{(j)}$ is the unit vector with the j-th element equal to 1. $\succeq_{cp}$ means that the matrix belongs to a completely positive cone. Unless stated otherwise, $\mathbf{0}$ is a zero vector or zero matrix with the appropriate shape.

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

Taking the duality on $L_{CP}(\boldsymbol{y})$ yields the following results

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

$$s.t. \sum_{i\in[M]} \boldsymbol{a}_i\alpha_i - \sum_{j\in[N]} \boldsymbol{e}_{(j)}\theta_j - 2\boldsymbol{\varphi} = \begin{bmatrix} \mathbf{0}_{n\times 1} \\ -\boldsymbol{y}_{m\times 1} \\ \mathbf{0}_{M\times 1} \end{bmatrix} \tag{10}$$

$$\sum_{i\in[M]} \boldsymbol{a}_i\boldsymbol{a}_i^T\beta_i + \sum_{j\in[N]} \boldsymbol{E}_{(j,j)}\theta_j - \boldsymbol{W} = \mathbf{0} \tag{11}$$

$$-2\boldsymbol{\Psi} = \begin{bmatrix} \boldsymbol{E}^{[n]} \\ \mathbf{0} \end{bmatrix} \tag{12}$$

$$\begin{pmatrix} \tau & \boldsymbol{\xi}^T & \boldsymbol{\varphi}^T \\ \boldsymbol{\xi} & \boldsymbol{\Xi} & \boldsymbol{\Psi}^T \\ \boldsymbol{\varphi} & \boldsymbol{\Psi} & \boldsymbol{W} \end{pmatrix} \succeq_{co} \mathbf{0} \tag{13}$$

α_i, β_i, and θ_j are the dual variables for constraint (6), (7), and (8) respectively. $\begin{pmatrix} \tau & \boldsymbol{\xi}^T & \boldsymbol{\varphi}^T \\ \boldsymbol{\xi} & \boldsymbol{\Xi} & \boldsymbol{\Psi}^T \\ \boldsymbol{\varphi} & \boldsymbol{\Psi} & \boldsymbol{W} \end{pmatrix}$ is the dual variable for the completely positive cone constraint (9). According to the weak duality theorem, $L_{CO}(\boldsymbol{y}) \geq L_{CP}(\boldsymbol{y})$.

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

We first divide the solution to (5) into two parts. One is composed of decision variables $\boldsymbol{\theta}$, and the other consists of all slack variables $\boldsymbol{s}$, i.e., $\boldsymbol{x} = \binom{\boldsymbol{\theta}}{\boldsymbol{s}}$, where $\boldsymbol{\theta} = (\hat{\boldsymbol{u}}, \boldsymbol{v}) \in \mathbb{R}_+^{(n+m)}, \boldsymbol{s} = (\boldsymbol{s}^\dagger, \hat{\boldsymbol{u}}^\dagger, \boldsymbol{v}^\dagger) \in \mathbb{R}_+^{(r+n+m)}$. According to Theorem 2 in Yan et al. (2018), $L_{CO}(\boldsymbol{y}) = L_{CP}(\boldsymbol{y})$ holds as long as the following two conditions hold:

1. 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.

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

Since the first condition is given in the proposition, we only need to prove that the second condition is always satisfied. It is easy to check that $\boldsymbol{x}^{(0)} = \begin{pmatrix} \boldsymbol{\theta}^{(0)} \\ \boldsymbol{s}^{(0)} \end{pmatrix} = \begin{pmatrix} \mathbf{1} \\ \mathbf{0} \end{pmatrix}$ is a feasible solution, and $\boldsymbol{\theta}^{(0)}$ is strictly positive.

Another two sets of feasible solutions exist:

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

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

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

It is also easy to check that all vectors $\boldsymbol{\theta}$ in $\mathcal{S}_1 \cup \mathcal{S}_2$ span 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 that meets the second condition, which completes the current step of the proof.

Step 4: Combine with the first-stage problem

We further combine $L_{CO}(\boldsymbol{y})$ with the first-stage problem and replace some of the decision variables in (13) with equations (10), (11), and (12). Now, we obtain the rFOLI problem in mixed-integer copositive cone form, i.e., problem CO.

□

3.3. Acceleration of Branch&Bound

Although we have reformulated rFOLI as a convex conic problem based on a copositive cone, Model CO still cannot be efficiently solved because of the complexity of the copositive cone and the binary decision variable $\boldsymbol{z}$. The former issue can be addressed via semidefinite positive relaxation, while the latter issue is not trivial, and brute-force search is not suitable due to exponentially many solutions. In this subsection, we adopt the classic branch-and-bound framework with depth-first search to search for optimal solutions. To accelerate the branch-and-bound procedure, we exploit known information over moments to obtain an interpretable initial solution instead of randomly generating an initial solution. Additionally, we propose a dual-variable-based heuristic to identify a better branch in each iteration. The heuristic quickly determines a tighter bound such that fewer nodes are visited. As the only integer decision variable is $\boldsymbol{z}$, the acceleration technique proposed in this subsection is applied only to $\boldsymbol{z}$.

Since the main aim of building a new warehouse is to counter future uncertainty, an intuitive idea for determining an initial solution is to rank the weighted average demand upper bound and then select the top-$\lfloor \frac{m}{2} \rfloor$ locations as the initial solution. Following this idea, we can identify supply nodes that have the potential to relieve the greatest amount of uncertainty. 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 the moment information. Then, we 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 locations for warehouses according to the following criterion.

$$\boldsymbol{z}^{init} = \begin{cases} z_i = 1, & if\ (i) \leq \lfloor \frac{m}{2} \rfloor \\ z_i = 0, & o.w. \end{cases} \tag{14}$$

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

In terms of branching, note that the decision variable $\boldsymbol{\alpha}$ is the dual variable to the first-M constraints of equation system $\hat{\mathcal{L}}_{lp}^{\dagger}$. Specifically, $\alpha_k, \forall k \in [r]$ is the dual variable to the first-r constraints, which is directly imposed on the network structure. According to the duality theorem, economically, the optimal solution $\alpha_k^*, \forall k \in [r]$ can be interpreted as the value of the corresponding road in the current iteration. Similar arguments have been proposed in Yan et al. (2018), in which the authors' main purpose was to design a sparse network structure by iteratively deleting arcs from a fully flexible system. Since α_k^* is the value of the k-th arc, they greedily delete the arc with the smallest value in each iteration. In our branch-and-bound procedure, we exploit the value of α_k^* as well. Intuitively, in each iteration, we greedily select one supply node that has not been visited yet, having the highest aggregated value of all roads that originate from it. In other words, we always choose the most valuable supply node as the next branch. To describe our idea in a more clear mathematical manner, we first define an index mapping from road (i, j) to a scalar index, representing the road 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 visited before the t-th iteration. Mathematically, we choose i^* as the next branch according to (15).

$$i^* = \arg\max_{i \in \mathcal{W}^t} \sum_{j \in \Gamma(i)} \alpha_{(ij)} \tag{15}$$

One factor to note is that the fixed cost and holding cost are not considered. It is possible to include these costs as a denominator when calculating the return on investment (ROI) of each supply node; however, since $\boldsymbol{\alpha}$ reflects the values of roads in only the current iteration, in which there is no inventory stored at the chosen location, the 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 (15) as the criterion shortens the computational time substantially.

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

4. Numerical Study

In this section, we conduct numerous experiments to examine the advantages of the proposed two-stage distributionally robust model and the equivalent conic reformulation. First, we run our models to scrutinize the performance, including total cost, fulfillment rate, Value-at-Risk, etc.

Algorithm 1 Accelerated Branch and Bound Algorithm

Set $\boldsymbol{z}^{init}$ according to (14)
Solve CO with $\boldsymbol{z}^{init}$ to get L^{init}
Initialize $\boldsymbol{z}^* \leftarrow \boldsymbol{z}^{init}$, $L^* \leftarrow L^{init}$
Initialize constraint set $\mathcal{C}$ on $\boldsymbol{z}$, $\mathcal{C} := \{\boldsymbol{0} \leq \boldsymbol{z} \leq \boldsymbol{1}\}$
Run DFS($\mathcal{C}$)

function DFS($\mathcal{C}$)
    Solve linear relaxation CO with $\mathcal{C}$ to get current objective value L
    if $L < L^*$ **then**
        if $\boldsymbol{z}$ is integer **then**
            $L^* \leftarrow L$, $\boldsymbol{z}^* \leftarrow \boldsymbol{z}$
        else
            Select branch i according to (15)
            $\mathcal{C} \leftarrow \mathcal{C} \cup \{z_i = 1\}$
            Run DFS($\mathcal{C}$)
            $\mathcal{C} \leftarrow \mathcal{C} \cup \{z_i = 0\}$
            Run DFS($\mathcal{C}$)
        end if
    end if
end function

Second, we conduct sensitivity analyses on three dimensions, namely, the correlation parameter ρ, coefficient of variation c_v, and risk attitude parameter κ. Finally, we extend our experiments to more general network structures and repeat all the experiments in the first two steps to obtain a comprehensive understanding of the proposed models in a more general setting (see Appendix B). A stochastic model and a mean-variance model, which will be discussed in detail in the next subsection, are introduced as benchmarks throughout this section. Experiments are conducted on a computer with 16 GB memory and an AMD 3700X CPU. Throughout this section, we use positive semidefinite relaxation to solve copositive programming problems. Specifically, we use $\{\boldsymbol{A}\,|\,\boldsymbol{A} = \boldsymbol{A}_1 + \boldsymbol{A}_2, \boldsymbol{A}_1 \succeq \boldsymbol{0}, \boldsymbol{A}_2 \geq \boldsymbol{0}\}$ as an inner approximation to the copositive cone. We use Mosek 9.2 to solve conic problems and Gurobi 9.0.3 to solve mixed-integer problems. All codes are open-source[1], and interested readers can easily reproduce the results of our experiments.

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

4.1. Experimental 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 provided to describe how we generate synthetic data for the 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 considers only two statistics, mean and variance, and does not consider correlations between demand nodes. Since our model exploits only the first two moments, the amount of available information is slightly more than that of the mean-variance model but less than that of the stochastic model.

SAA with Empirical Distribution

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

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

where $g(\boldsymbol{y}, \tilde{\boldsymbol{d}})$ is a recourse allocation problem for any given $\boldsymbol{y}$ and realized $\tilde{\boldsymbol{d}}$, exactly the same as Model (2). $(\boldsymbol{y}, \boldsymbol{z})$ are the inventory and location selection decision variables. The feature 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 the SAA model involves training samples, and by tradition, we regard this benchmark as the sample average approximation.

Mean-Variance

The other benchmark is a distributionally robust model with only mean $\boldsymbol{\mu}$ and variance $\boldsymbol{\sigma}^2$ information, i.e., the covariance is ignored.

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

where the ambiguity set is

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

$\sigma_j^2, \forall j \in [n]$ is the variance of $\tilde{d}_i$. Compared to rFOLI, the most notable distinction is the construction of the distribution ambiguity set. Although MV is still intractable in most cases, we can transform $\mathscr{F}_{mv}(\boldsymbol{\mu},\boldsymbol{\sigma})$ into $\mathscr{F}(\boldsymbol{\mu},\boldsymbol{\Sigma}_{mv})$, where $\boldsymbol{\Sigma}_{mv} = \boldsymbol{\mu}\boldsymbol{\mu}^T + \boldsymbol{Diag}(\boldsymbol{\sigma}^2)$. Then, every analysis holds, as shown in Proposition 3. Replacing $\boldsymbol{\Sigma}$ in CO with $\boldsymbol{\Sigma}_{mv}$ yields the resulting conic model. For some special cases, such as the 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

Consider a network with m potential warehouses and n demand nodes. We randomly generate dozens or hundreds of networks, and each network is defined by a six-element tuple $(\mathcal{W}, \mathcal{R}, \mathcal{E}, \boldsymbol{f}, \boldsymbol{h}, \boldsymbol{\mu})$. $\mathcal{W}$ and $\mathcal{R}$ are the sets of potential warehouse nodes and demand nodes, and $\mathcal{E}$ is a set of links. According to Ni et al. (2018), the average node degree of a typical real-world road network is approximately 2.4; therefore, we require $|\mathcal{E}| = 1.2(m+n)$. The following two constraints on the generated network must satisfied: (1) at least one road is linked to each supply node and (2) at least one road is linked to each demand node. These conditions help to avoid the appearance of isolated nodes. $\boldsymbol{f}$ is the fixed setup cost, which is randomly drawn 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, 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 $\kappa = 1.0$. 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 nodes, 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 that the underlying demand distribution follows a truncated multivariate Gaussian distribution, with domain $\mathbb{R}_+^n$, mean $\boldsymbol{\mu}$, and covariance matrix $\boldsymbol{\Sigma}_{\rho,c_v}$, 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 the mean and variance of demand are known. Moreover, for model SAA, we assume 30 historical samples are available. The sample size is statistically large enough to obtain unbiased estimates of the first two moments.

After obtaining optimal solutions by solving the three models, we randomly generate 2000 demand realizations to conduct the second-stage simulations. A total of 1000 demand realizations are drawn from the underlying true multivariate Gaussian distribution, and another 1000 demand realizations are drawn from a mixture distribution with four independent and equally weighted components, including a 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 deviation $\boldsymbol{\sigma}$. The above four distributions are truncated to $\mathbb{R}_+^n$. It is easy to check that the mixture distribution ignores correlations but has a mean and standard deviation near the true $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$. We separately analyze the results for the truncated Multivariate Gaussian Distribution (MGD) and Mixture Distribution (MixD). And for each network, we regard the average (or worst-case) result for 1000 simulations as indicative of 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 in the above subsection, we first restrict ourselves to the $(|\mathcal{W}|, |\mathcal{R}|) = (8, 8)$ structure for a closer look at the model's performance. We randomly generate 200 networks with $(\rho, c_v, \kappa) = (0.3, 0.3, 1.0)$. Unless stated otherwise, all analyses in this subsection rely on the

200 networks: for each network, we construct a measurement based on the simulation results, for example, the average unmet demand, to represent the network's performance. Then, we calculate the average for the 200 networks.

First-stage Deployment

First, we compare the first-stage deployments and characteristics of the designed networks. Table 1 summarizes the first-stage decisions and resulting networks. The first column, *total inv*,

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)

represents the average total inventory of the 200 networks. The numbers in parentheses show the standard deviations. Clearly, Model CO results in the highest inventory level, while Model MV has the lowest inventory level. The difference results 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 the potential locations to build warehouses. The third column is the number of roads in the designed networks: Model CO designs the sparsest network. The last two columns exhibit the warehouse node and demand node degrees.

Cost

We next investigate the cost performance, including the first-stage deployment cost and second-stage penalty due to unmet demand, which are the costs of greatest concern. Table 2 shows the average costs for the 200 networks under different models with MGD and MixD realizations. The numbers in parentheses represent the standard deviation. The first two columns represent the setup cost $\boldsymbol{f}$ and the holding cost $\boldsymbol{h}$. $\boldsymbol{p}$ is the penalty cost, or equivalently, the unsatisfied demand. *total cost* is the aggregated cost. Clearly, Model CO always achieves the lowest total cost. One remarkable decrease occurs for $\boldsymbol{p}$, which decreases from approximately 64.5 (for SAA) to 53.1 (for CO), i.e., a 17.7% decrease in unmet demand under the in-sample simulation. For the MixD simulation, the decrease is also considerable, from 70.0 (for SAA) to 57.3, approximately 18.4%,

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)

or from 108.7 (for MV) to 57.3, approximately 47.3%. Although the absolute decrease is marginal, the relative decrease is remarkable, which highlights the advantages of our proposed conic model: we can further substantially reduce the unmet demand even though it is already small. This managerial insight is quite attractive, especially to disaster management experts or fulfillment-oriented warehouse managers, since their priority is to reduce unmet demand without increasing expenditure.

We next scrutinize the unmet demand more deeply. Figure 2 shows the distribution of the average unmet demand estimated by kernel density estimation (KDE), where the horizontal axis shows the average unmet demand obtained through simulations, and the vertical axis represents the probability.

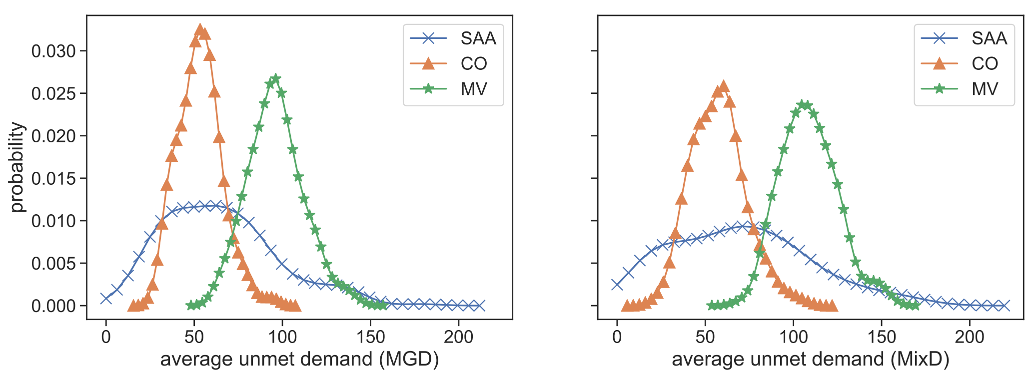


Figure 2: Average unmet demand distribution, $(\rho, c_v, \kappa) = (0.3, 0.3, 1.0)$

In the left graph, when the true demand distribution aligns with the distribution that decision makers know, Model CO dominates Model MV, except in some instances of overlap around 75. Although Model CO achieves a higher level of concentration around 55, the lowest 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 the MixD simulations. Similarly, Model CO dominates Model MV most of the time. However, the most interesting change is the increased dispersion of Model SAA. Clearly, the performance of Model SAA becomes unstable in different networks, while CO and MV remain concentrated near their former mean values. The changes in the standard deviations of $\boldsymbol{p}$ in Table 2 further support our findings.

Service Level

Another critical metric is the service level. Table 3 summarizes two kinds of service levels. The type 1 service level, or α service level, is an event-oriented performance criterion that measures the probability that the entire demand will be completely satisfied by stock on hand. The type 2 service level, or β service level, is a quantity-oriented performance measure describing the proportion of satisfied demand to total demand. Model CO performs slightly bit better than Model SAA in terms

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

of the type 2 service level. Moreover, in terms of the type 1 service level, Model CO outperforms the two other models by approximately 1%.

Value-at-Risk

Furthermore, we calculate the 99.9th, 95th, 90th, 85th, and 80th quantiles of unmet demand to demonstrate the ability of the models to hedge long-tail risk. When the true distribution is known to decision makers, these quantiles can 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 indicates that its ability to hedge long-tail risk comes from information about the true underlying distribution. Surprisingly, columns *MixD VaR* in Table 4 further reveal that Model CO is sufficiently robust to the misspecification in the distribution. In terms of the 95th quantile, Model CO's unmet demand under MixD is 20% (from 335.4 to 267.9) lower than that of

Table 4: Unmet Demand with Various Quantiles

	MGD VaR					MixD VaR				
unmet demand	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

Model SAA, while the gap is only 13.6% (from 406.4 to 350.9) under the MGD situation. This result verifies the advantages of Model CO and the distributionally robust technique, especially when historical data are limited and misspecification in the distribution is likely.

CPU Time

Computational time is also essential in practice. Although the problem we consider is a strategy-level problem, reasonable computational time is necessary. Therefore, we record the CPU time for solving each optimization problem, thereby highlighting the superiority of the proposed branch-and-bound acceleration algorithm. Notably, since we run experiments with four parallel tasks synchronously on one computer, the CPU time is slightly longer. Table 5 summarizes the time

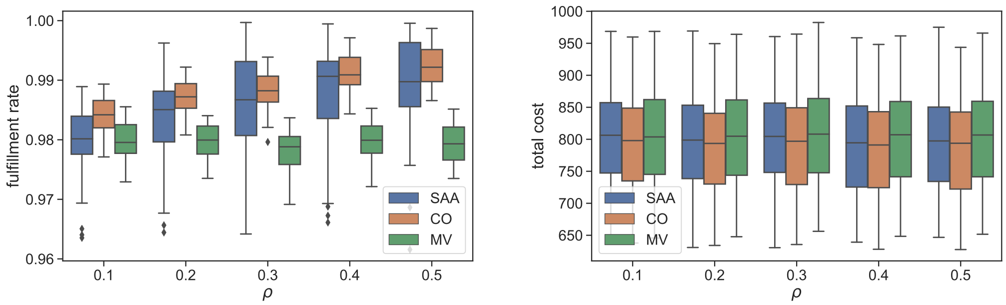
Table 5: CPU Time to Solve Models

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

needed to solve each model. We use *(acce)* to indicate that the model is solved with the accelerated branch-and-bound technique proposed in Section 3.3. The first column, *CPU Time (s)*, reports the computational time, in seconds, required to obtain the optimal solution, and the second column, *Node*, exhibits the number of visited branch-and-bound nodes before terminating the algorithm. Clearly, for Model CO, the accelerated algorithm reduces the time by more than half by visiting fewer nodes. The decreased computational time also demonstrates that including more variables in the reformulated model in Proposition 1 is worthwhile.

4.2.2. Sensitivity Analysis

We modify some of the factors (correlation parameter ρ, coefficient of variation c_v, and risk attitude κ) to conduct sensitivity analyses. Since we have explored the measurement of interest in



(a) Fulfillment Rate (b) Total Cost

Figure 3: Sensitivity Analysis on ρ

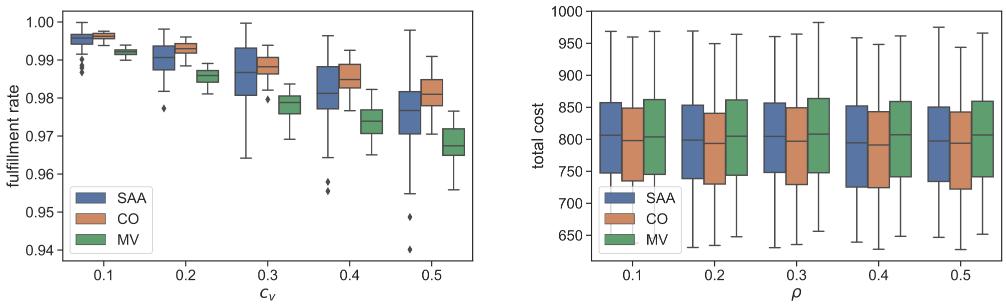
detail, throughout this part, we consider only 50 graphs for each parameter combination. Additionally, we focus on MixD simulations, where the true underlying distribution deviates from the distribution used in the training sample generation process. This situation is likely to occur in reality; hence, a sufficiently robust warehouse network is needed. The results show that all the conclusions proposed earlier hold.

Correlation Parameter ρ

We first present a sensitivity analysis of the correlation parameter ρ, which we change from 0.05 to 0.5. Figure 3a shows the fulfillment rates of the three models under different values of parameter ρ. The fulfillment rate, or type 2 service level, measures the proportion of demand that is satisfied. Model CO always has a higher fulfillment rate, on average, than the other models. Moreover, the perturbation of Model CO is much less severe than that of the other models, indicating that Model CO is robust to the network structure. An interesting trend is that the fulfillment rate of Model MV is stable over a range of ρ values, while that of the other models increases. The difference is due to the models' abilities to utilize correlation information. While MV does not consider any correlation between demand nodes, CO and SAA take the correlation into consideration explicitly and implicitly, respectively. Another noteworthy phenomenon is that outliers appear only in Model SAA, reflecting its weakness in terms of hedging extreme risks.

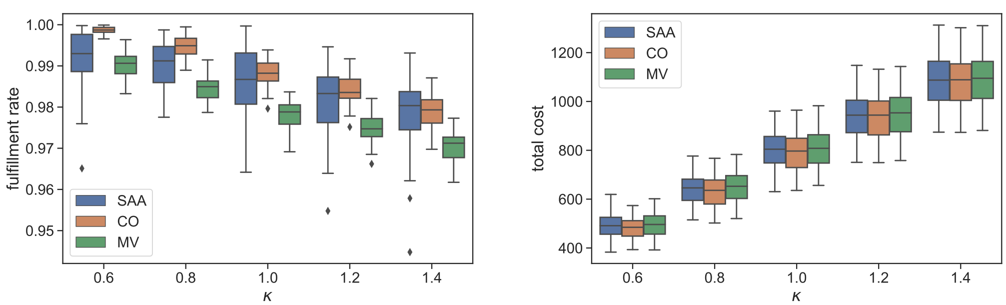
Figure 3b exhibits the total cost of the three models under different values of ρ. Model CO can always achieve the lowest total cost.

Coefficient of Variation c_v



(a) Fulfillment Rate (b) Total Cost

Figure 4: Sensitivity Analysis on c_v



(a) Fulfillment Rate (b) Total Cost

Figure 5: Sensitivity Analysis on κ

As in the analysis of ρ, we first consider the fulfillment rate. Figure 4a depicts fulfillment rates under a range of c_v values. As c_v increases, the mean fulfillment rates decrease in general, and the variance of the fulfillment rate increases. However, among the three models, CO has the lowest rate of decrease and the smallest variance, which further emphasizes the robustness of Model CO. Figure 4b shows the change in total cost in terms of c_v. The total cost increases as c_v increases since more warehouses and products are needed to counter the increasing uncertainty.

Risk Attitude Parameter κ

Figure 5a represents the relationship between the fulfillment rate and risk attitude κ. As κ increases, i.e., the decision maker focuses more on the first-stage cost, the fulfillment rate decreases significantly. This is a direct result of a change in risk attitude since a larger κ reflects a more aggressive attitude to future risk. Therefore, the corresponding prepositioning decisions become less conservative. Another noteworthy phenomenon is the diminishing performance gap between CO and SAA as κ increases. Since more attention is shifted to the deployment cost, the advantage of CO in addressing demand uncertainty is gradually weakened. Therefore, CO's superiority in terms of fulfillment rate diminishes. Figure 5b expresses the total costs for the models as κ increases. CO has a slightly lower total cost, regardless of the value of κ.

To summarize this subsection, we conduct numerous experiments on given networks to evaluate the performance of the proposed copositive cone-based conic model. The conic model is robust to demand uncertainty in most cases and outperforms the SAA-based model when historical data are limited but statistically sufficient to obtain moment statistics. Additionally, it is essential to incorporate correlation information into modeling for situations such as disaster management where demand correlation indeed exists and plays an important role. As the variance or correlation increases, these values become more influential in the subsequent second-stage fulfillment performance. Furthermore, we demonstrated that the acceleration algorithm indeed reduces the computation time.

5. Extensions

5.1. Multiple Items and Multiple Periods

In this subsection, we extend our fulfillment-oriented location-inventory model to a multiple-items and multiple-periods model. Suppose there are K different kinds of items, and the total planning period is S. For each item $k \in [k]$ in each period $s \in [S]$, suppose the first moment $\boldsymbol{\mu}_{ks}$ and second moment $\boldsymbol{\Sigma}_{ks}$ are known to decision makers. Further, assume that the demand variables are uncorrelated over items and periods. Under these assumption, the second-stage problem is separable and can be modified as:

$$
\begin{aligned}
\min_{\boldsymbol{y}_{ks},\boldsymbol{z}} \quad & \kappa(\boldsymbol{f}^T\boldsymbol{z} + \sum_{k\in[K]}\sum_{s\in[S]} \boldsymbol{h}_{ks}^T\boldsymbol{y}_{ks}) + \sum_{k\in[K]}\sum_{s\in[S]} \sup_{\mathbb{P}\in\mathscr{F}(\boldsymbol{\mu}_{ks},\boldsymbol{\Sigma}_{ks})} \mathbb{E}_{\mathbb{P}}\left[g_{ks}(\boldsymbol{y}_{ks},\tilde{\boldsymbol{d}})\right] \\
\text{s.t.} \quad & \boldsymbol{y}_{ks} \leq U\boldsymbol{z},\ \forall k\in[K], s\in[S] \\
& \boldsymbol{y}_{ks} \in \mathbb{R}_+^m, \boldsymbol{z}\in\{0,1\}^m
\end{aligned}
\tag{16}
$$

Let $L_{ks}(\boldsymbol{y}) = \sup_{\mathbb{P}\in\mathscr{F}(\boldsymbol{\mu}_{ks},\boldsymbol{\Sigma}_{ks})} \mathbb{E}_{\mathbb{P}}\left[g_{ks}(\boldsymbol{y},\tilde{\boldsymbol{d}})\right]$. Following the analysis in Section 3.2, we can reformulate $L_{ks}(\boldsymbol{y})$ as a conic programming problem. The resulting model contains $|K|\times|S|$ copositive cone constraints, which slows the computational process but, theoretically, with positive semidefinite relaxation, does not make the model intractable.

5.2. Capacitated Warehouses

Capacitated warehouses are common in reality, and the setup cost of warehouses $\boldsymbol{f}$ can be capacity-specific. Since these constraints are imposed in the first stage, they are easy to incorporate into our model. Suppose K types of warehouses are available, each of which has a limited capacity $C_k, \forall k \in [K]$ with a fixed setup cost $\boldsymbol{f}_k, \forall k \in [K]$. Now, we can revise the reduced model (rFOLI) as:

$$
\begin{aligned}
\min_{\boldsymbol{y},\boldsymbol{z}_k} \quad & \kappa\Big(\sum_{k\in[K]} \boldsymbol{f}_k^T \boldsymbol{z}_k + \boldsymbol{h}^T\boldsymbol{y}\Big) + \sup_{\mathbb{P}\in\mathscr{F}(\boldsymbol{\mu},\boldsymbol{\Sigma})} \mathbb{E}_{\mathbb{P}}\left[g(\boldsymbol{y},\tilde{\boldsymbol{d}})\right] \\
\text{s.t.} \quad & \boldsymbol{y} \le \sum_{k\in[K]} C_k \boldsymbol{z}_k,\ \forall k \in [K] \\
& \sum_{k\in[K]} \boldsymbol{z}_k \le \mathbf{1}, \\
& \boldsymbol{y} \in \mathbb{R}_+^m, \boldsymbol{z}_k \in \{0,1\}^m
\end{aligned}
\tag{17}
$$

The objective function is self-evident. The first constraint requires that the number of stored products does not exceed warehouse capacity limits, while the second constraint restricts the number of warehouses at each location to one. We can still follow the same analysis as in Section 3.2 to reformulate the *sup* problem, and the resulting conic model is still tractable under relaxation, although it involves additional binary variables. One notable result is that imposing additional tractable constraints on the first-stage problem does not theoretically increase the difficulty of obtaining the solution. Therefore, our proposed model is compatible to additional constraints in the first-stage problem, as long as they are polynomial-time-solvable constraints, such as linear constraints, second-order cone constraints, and positive semidefinite constraints.

5.3. Uncertainty in the First and Second Moments

We further introduce uncertainty in the first two moments into our model by utilizing the technique proposed in Delage and Ye (2010). Suppose that the first moment $\boldsymbol{\mu}$ and second cross-moment $\boldsymbol{\Sigma}$ are uncertain and that the uncertainty set can be defined 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 a given vector and matrix describing an ellipsoid. Parameters $\gamma_1 \geq 0$ and $\gamma_2 \geq 1$ naturally quantify one's confidence in $\boldsymbol{\mu}_0$ and $\boldsymbol{\Sigma}_0$. Therefore, incorporating new decision variables $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ into Model (rFOLI) yields the following model:

$$\begin{aligned} \min_{\boldsymbol{y}, \boldsymbol{z}} \quad & \kappa(\boldsymbol{f}^T \boldsymbol{z} + \boldsymbol{h}^T \boldsymbol{y}) + \max_{(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \in \mathscr{D}} \sup_{\mathbb{P} \in \mathscr{F}(\boldsymbol{\mu}, \boldsymbol{\Sigma})} \mathbb{E}_{\mathbb{P}} \left[g(\boldsymbol{y}, \tilde{\boldsymbol{d}}) \right] \\ \text{s.t.} \quad & \boldsymbol{y} \leq U \boldsymbol{z}, \\ & \boldsymbol{y} \in \mathbb{R}_+^m, \boldsymbol{z} \in \{0,1\}^m \end{aligned} \tag{18}$$

Recall that $\sup_{\mathbb{P} \in \mathscr{F}(\boldsymbol{\mu}, \boldsymbol{\Sigma})} \mathbb{E}_{\mathbb{P}} \left[g(\boldsymbol{y}, \tilde{\boldsymbol{d}}) \right]$ can be reformulated as a completely positive programming maximization problem, which can be combined with the decision $(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \in \mathscr{D}$. Since $\mathscr{D}$ involves only positive semidefinite constraints, the strong duality still holds between the copositive programming minimization problem and the completely positive programming maximization problem. 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 problem with an ambiguity set characterized by the first two moments. Utilizing min-cut max-flow theory, we proposed an equivalent reformulation to transform the model into a mixed-integer conic problem based on copositive cones. We also proposed a dual-variable-based selection criterion to accelerate the branch-and-bound process. Extensive numerical studies demonstrate the necessity of incorporating demand correlation when it exists, and a larger coefficient of variation increases the impact of correlation. Indeed, our conic model includes demand correlation information. Therefore it is robust to the network structure and long-tail risks, and achieve the lowest total cost in most cases.

Our study also demonstrated the superiority of copositive programming and its advantages in solving operation management problems. We hope that our study will stimulate further research on copositive programming and its applications in the field of operations management. One potential research direction is to consider a more realistic recourse problem that includes shipping costs, ordering costs, location-specified penalty costs, etc. Interested readers are also encouraged to explore the analytic properties of copositive cones mathematically.

References

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

Berman, O., Krass, D., Tajbakhsh, M. M., 2012. A coordinated location-inventory model. European Journal of Operational Research 217 (3), 500–508.

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., 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.

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

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.

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.

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.

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.

Lim, M. K., Bassamboo, A., Chopra, S., Daskin, M. S., 2013. Facility location decisions with random disruptions and imperfect estimation. Manufacturing & Service Operations Management 15 (2), 239–249.

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.

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.

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.

Rebennack, S., Reinelt, G., Pardalos, P. M., 2012. A tutorial on branch and cut algorithms for the maximum stable set problem. International Transactions in Operational Research 19 (1-2), 161–199.

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

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.

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 $g(\boldsymbol{y}, \boldsymbol{z}, \tilde{\boldsymbol{d}}) \geq g(\boldsymbol{y}, \tilde{\boldsymbol{d}})$ and $g(\boldsymbol{y}, \boldsymbol{z}, \tilde{\boldsymbol{d}}) \leq g(\boldsymbol{y}, \tilde{\boldsymbol{d}})$ both hold. For clarity, we rewrite both problems as follows:

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

Define an index set $\boldsymbol{z}^+$ containing all location indices for building a warehouse, i.e., $\boldsymbol{z}^+ := \{i \mid z_i = 1,\ \forall i \in [m]\}$.

$$\begin{aligned} g(\boldsymbol{y}, \tilde{\boldsymbol{d}}) = \quad \min_{\boldsymbol{t}} \quad & \sum_{j\in[n]} \tilde{d}_j - \sum_{(i,j)\in\mathcal{E}(\boldsymbol{z})} t_{ij} - \sum_{(i,j)\notin\mathcal{E}(\boldsymbol{z})} t_{ij} && \text{(A.2)} \\ s.t. \quad & \sum_{i:(i,j)\in\mathcal{E}(\boldsymbol{z})} t_{ij} + \sum_{i:(i,j)\notin\mathcal{E}(\boldsymbol{z})} t_{ij} \leq \tilde{d}_j, \forall j \in [n] \\ & \sum_{j:(i,j)\in\mathcal{E}} t_{ij} \leq y_i, \forall i \in \boldsymbol{z}^+ \\ & \sum_{j:(i,j)\in\mathcal{E}} t_{ij} \leq 0, \forall i \notin \boldsymbol{z}^+ && \text{(A.3)} \\ & t_{ij} \geq 0, \forall (i,j) \in \mathcal{E} \end{aligned}$$

In $g(\boldsymbol{y}, \tilde{\boldsymbol{d}})$, the right-hand-side coefficient of the third constraint is zero since establishing a warehouse is a prerequisite for storing products.

1. $g(\boldsymbol{y}, \boldsymbol{z}, \tilde{\boldsymbol{d}}) \geq g(\boldsymbol{y}, \tilde{\boldsymbol{d}})$

   Clearly, the optimal solution $\boldsymbol{t}^*$ to (A.1) is always a feasible solution to (A.2) by setting other $t_{ij} = 0, (i,j) \notin \mathcal{E}(\boldsymbol{z})$. Therefore, $g(\boldsymbol{y}, \boldsymbol{z}, \tilde{\boldsymbol{d}}) \geq g(\boldsymbol{y}, \tilde{\boldsymbol{d}})$ holds.

2. $g(\boldsymbol{y}, \boldsymbol{z}, \tilde{\boldsymbol{d}}) \leq g(\boldsymbol{y}, \tilde{\boldsymbol{d}})$

Now, suppose $\boldsymbol{t}^*$ is the optimal solution to (A.2). We split the solution into two parts according to whether $i \in \boldsymbol{z}^+$, i.e., $\boldsymbol{t}^* = (\boldsymbol{t}^+, \boldsymbol{t}^0)$, where $\boldsymbol{t}^+$ contains the allocation decision from warehouses and $\boldsymbol{t}^0$ contains the allocation decisions from locations where no warehouse is built. From constraint (A.3), we have $\boldsymbol{t}^0 = \mathbf{0}$. It is easy to check that $\boldsymbol{t}^+$ is a feasible solution to (A.1). Therefore, $g(\boldsymbol{y}, \boldsymbol{z}, \tilde{\boldsymbol{d}}) \leq g(\boldsymbol{y}, \tilde{\boldsymbol{d}})$ always holds.

Combining the above two statements completes the proof. □

Appendix A.2. Proof for Proposition 2

Proof. Instead of considering $g(\boldsymbol{y}, \tilde{\boldsymbol{d}}) = \max\limits_{(\hat{\boldsymbol{u}},\boldsymbol{v}) \in \hat{\mathcal{L}}_{lp} \cap \{0,1\}^{n+m}} \tilde{\boldsymbol{d}}^T \hat{\boldsymbol{u}} - \boldsymbol{y}^T \boldsymbol{v}$, we consider its linear relaxation problem $\max\limits_{(\hat{\boldsymbol{u}},\boldsymbol{v}) \in \hat{\mathcal{L}}_{lp}} \tilde{\boldsymbol{d}}^T \hat{\boldsymbol{u}} - \boldsymbol{y}^T \boldsymbol{v}$. We prove this proposition by demonstrating that problem (4) with the linear relaxation problem is NP-hard according to Bertsimas et al. (2010). As discretizing the feasible region does not reduce the computation complexity, (4) is at least NP-hard.

Firstly, we take the dual on (4) as follows:

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

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

$$
\begin{aligned}
& \inf_{s, \boldsymbol{t}, \boldsymbol{\Xi}} s + \boldsymbol{\mu}^T \boldsymbol{t} + \boldsymbol{\Sigma} \bullet \boldsymbol{\Xi} \\
& s.t. \quad \min_{(\hat{\boldsymbol{u}},\boldsymbol{v}) \in \hat{\mathcal{L}}_{lp}, \tilde{\boldsymbol{d}} \in \mathbb{R}^n_+} \tilde{\boldsymbol{d}} \boldsymbol{\Xi} \tilde{\boldsymbol{d}} + \tilde{\boldsymbol{d}}^T (\boldsymbol{t} - \hat{\boldsymbol{u}}) + \boldsymbol{y}^T \boldsymbol{v} + s \geq 0
\end{aligned} \tag{A.4}
$$

Clearly, strong duality holds between the maximization problem and the dual minimization problem. According to Theorem 3.1 in Bertsimas et al. (2010), the separation problem, i.e., for a given $\boldsymbol{t}$, $\boldsymbol{y}$ and s, check if the constraint (A.4) 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.4), is NP-complete. Because of the equivalence of separation and optimization, the minimization problem is NP-hard. Therefore, evaluating the value of $L(\boldsymbol{y})$ is at least NP-hard. $\square$

Appendix B. Simulations on General Networks

We conduct more simulation experiments on general networks and compare the results with those for $(8, 8)$ networks. Specifically, we consider $(|\mathcal{W}|, |\mathcal{R}|) = (4, 8)$ and $(|\mathcal{W}|, |\mathcal{R}|) = (8, 12)$. Except for changes in the number of potential supply nodes and demand nodes, the other parameters remain the same. We also modify three parameters, namely, the correlation parameter ρ, coefficient of variation c_v, and risk attitude parameter κ, to conduct sensitivity analyses. Generally, we repeat all the experiments in this section but on different networks. Table B.6 summarizes the performance of different networks in terms of fulfillment rate (FR). We regard the performance of SAA as the benchmark, and the ratios in Table B.6 are calculated according to $\frac{FR_x - FR_{SAA}}{FR_{SAA}} \times 100\%, x \in \{CO, MV\}$.

In general, Model CO achieves higher fulfillment rates than Model SAA does, except in a few situations, while Model MV achieves lower fulfillment rates than Model SAA does. When we gradually increase ρ from 0.05 to 0.5, as shown in the first part of Table B.6, the performance of MV gradually degrades, which implies the importance of considering correlations between nodes when correlations indeed exist. As c_v increases, as shown in the second part of Table B.6, the performance gap between CO and MV also increases. This phenomenon indicates that when the variance of demand is sufficiently large, explicitly considering cross-moment information, instead of only marginal moment information, can further improve fulfillment rates.

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

Table B.6: Fulfillment rate comparison on general networks

Fulfillment Rate			MGD									MixD								
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	-	0.30%	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%

Table B.7: Total cost comparison on general networks

Total Cost Relative Changes			MGD									MixD								
			(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%