Fulfillment-oriented Warehouse Network Design with Demand Information Uncertainty and Its Application in Upfront Warehouse Decision Making

Du Chen

Antai College of Economics and Management, Shanghai Jiao Tong University, China

Abstract

Disaster relief network, if properly designed, can save thousands of lives. However, disaster relief network design is hard due to the unavailability of demand information. In this paper, we study a network design problem for disaster relief purpose when only the support set, first moment and second moment informations are known to decision makers. A two-stage distributionally robust optimization problem is proposed. By utilizing properties of copositive cone (CO), which is widely adapted to study moment-based distributionally robust problems, the proposed model is equivalently reformulated to a CO programming model under modest assumptions. A case study shows that, in terms of total cost and second-type service level, our CO model with limited demand information is comparable to a two-stage stochastic model with full demand information. It also significantly outperforms a robust Ψ-expander model with only support set information.

Keywords: Distribution network design, Distributionally robust optimization, Copositive cone

1. Introduction

2. Literature Review

2.1. Network Design for Disaster Management

A network is *balanced* if there are an equal number of potential warehosues and demand areas, and the system is *asymmetric* if not all demand areas have i.i.d demand.

2.2. Moments-based Distributionally Robust Optimization

2.3. Copositive Programming

A n-dimensional Copositive Cone is defined as

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

Email address: chendu2017@sjtu.edu.cn (Du Chen)

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

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

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

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

During recent years, many operation management problem can be properly handled with copositive program and its approximation techniques. For example, Natarajan et al. (2011) applied copositive program on a project management problem to estimate the expected completion time and the persistency of each activity. Kong et al. (2013) handled a healthcare appointment-scheduling problem where only the mean and covariance of service durations are known. They formulated the problem as a convex copositive optimization

problem with a tractable semidefinite relaxation. Li et al. (2014) studied the sequencing problem with random costs. Utilizing copositve program technique, Yan et al. (2018) designed a roving team deployment plan for Singapore Changi Airport. Kong et al. (2020) further incorporated patients' no-show behavious into healthcare appointment scheduling problem, and reformulated the problem as a copositive program. We refer interested readers to Dickinson (2013) for general analytic properties of copositve cone, to Li et al. (2014) and Bomze (2012) for a comprehensive review over applications with copositive program.

3. Fulfillment-Oriented Network Design Problem

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

3.1. Model

The fulfillment-oriented network design (FOND) problem under demand distribuiton uncertainty, given the ambiguity set is characterized by first- and second-moment, can be formulated as:

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

where the distribution ambiguity set is:

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

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

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

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

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

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

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

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

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

where

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

Proof. See Appendix A.1 □

According to Proposition 1, instead of considering both $(\boldsymbol{I}, \boldsymbol{Z})$, we only need to take $\boldsymbol{I}$ into consideration. And getting rid of $\boldsymbol{Z}$ does not harm the optimality, however, at the cost of an increasing in the dimension of recourse decision variables $\boldsymbol{x}$. Surprisingly, the increased variables not only make further reformulation

easier, but also provides more insights on selection locations for warehouses, which actually accelarates the solving process. We will furthur explain this in Section 3.3.

From now on, we replace $g(\boldsymbol{I}, \boldsymbol{Z}, \tilde{\boldsymbol{d}})$ with $g(\boldsymbol{I}, \tilde{\boldsymbol{d}})$. With the more tractable recourse problem, we next equivalently reformulate Model (1) into a mixed integer copositive programming problem.

3.2. Reformulation

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

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

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

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

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

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

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

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

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

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

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

Moreover, the linear relaxation of the feasible region is

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

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

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

Unfortunately, though we have relaxed the inner allocation problem from an integer programming to a linear programming with a fixed feasible region $\hat{\mathcal{L}}_{lp}$, the difficuly of evaluating (4) does not reduced. Actually, Proposition 2 shows that problem (4), with feasible region $\hat{\mathcal{L}}_{lp}$, is still a NP-hard problem.

Proposition 2. *Given the feasible region to* (5) *is* $\hat{\mathcal{L}}_{lp}$*, calculating the value of problem* (4) *is still a NP-hard problem.*

Proof. See Appendix A.2 □

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

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

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

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

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

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

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

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

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

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

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

constraint with $\succeq_{co}$ is derived from $L(\boldsymbol{I})$. Formally, we propose this equivalent reformulation in Proposition 3.

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

Proof. We concisely describe the roadmap to prove this proposition here. Detailed proof is provided in Appendix A.3. We first transform the second-stage problem $L(\boldsymbol{I})$ into a maximizing completely positive conic problem according to Natarajan et al. (2011) Theorem 3.3. Then taking dual to obtain a minimizing problem on copositive cone. We further show that strong duality holds between the maximizing and minimizing problem. Finally, combining the second-stage minimizing problem with first-stage problem gives (CO). □

3.3. Speed Up Branch&Bound

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

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

$$\boldsymbol{Z}^{init} = \begin{cases} Z_i = 1, & if\ (i) \leq \lfloor \frac{m}{2} \rfloor \\ Z_i = 0, & o.w. \end{cases} \tag{6}$$

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

Another issus is about branching. Please note that the decision variable $\boldsymbol{\alpha}$ is the dual variable to first-M constraints of equation system $\hat{\mathcal{L}}_{lp}^{\dagger}$. And more specifically, $\alpha_k, \forall k \in [r]$ is the dual variable to first-r

constraints, which is directly imposed on network structure. According to duality theorem, economically, the optimal solution $\alpha_k^*, \forall k \in [r]$ can be interpreted as the value of corresponding road in current iteration. Similar arguments have been proposed in Yan et al. (2018), in which the authors' main purpose is designing a sparse network structure by iteratively deleting arcs from a full-flexibile system. Since α_k^* is the value of i-th arc, they greedily delete the arc with smallest value in each iteration. As to our branch and bound procedure, we can exploite the value of α_k^* as well. Intuitively, in each iteration, we greedily select one supply node, having the highest aggregated value of all roads originate from it, from remaining supply nodes as next branch. In other words, we always choose the most valuable supply node as the next branch. To describe our idea mathematically more clear, we first define an index mapping from road (i,j) to a scalar index, representing the roads index in [r]. For example we use $(ij), \forall (i,j) \in \mathcal{E}$ to represent road (i,j)'s index in set [r]. Let $\mathcal{W}^t$ be the set of supply nodes that have not been explored before t-th iteration. Mathematically, we choose i^* as the next branch according to (7).

$$i^* = \arg\max_{i \in \mathcal{W}^t} \sum_{j \in \Gamma(i)} \alpha_{(ij)} \tag{7}$$

One thing worthy of attention is that the fixed cost and the holding cost are not taken into consideration. It is possible to involve them as a denominator for calculating return of investment (ROI) of each supply node. However, since $\boldsymbol{\alpha}$ only reflect values of roads in current iteration, in which there is no inventory stored at the chosen location, ROI does not accurately reflect the importance of supply node. In some extreme cases, regarding ROI as select criterion leads to unnecessary search in inexpensive but less connected supply node. From a practical perspective, adopting (7) as criterion has already shortened computational time significantly.

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

4. Numerical Study

In this section, we conduct numerous experiments to examine the advantages of the proposed two-stage distributionally robust model, as well as the equivalent conic reformulation. First, We run our models in hundreads of randomly generated balanced but asymmetric networks with predetermined parameters. Second, we conduct sensitivity analysis on three dimensions, including correlation parameter ρ, coefficient of variantion c_v, and risk attitude parameter κ. Finally, we extend our experiments to more general network structures, and re-run all experiments in the first two steps, trying to obtain a comprehensive understanding of proposed models in a more general setting. Throughout this section, a stochastic model and a mean-variance model are introduced as benchmarks, which will be clearly illustrated in the next subsection.

Algorithm 1 Accelerated Branch and Bound Algorithm

Set $\boldsymbol{Z}^{init}$ according to (6)

Solve linear relaxation (CO) with $\boldsymbol{Z}^{init}$ to get L^{init}

Initialize $\boldsymbol{Z}^* \leftarrow \boldsymbol{Z}^{init}$, $L^* \leftarrow L^{init}$

Initialize $\boldsymbol{Z} \leftarrow \{0\}_{i\in[m]}$

Run DFS($\boldsymbol{Z}$)

function DFS(Z)

Solve linear relaxation (CO) with $\boldsymbol{Z}$ 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 (7)

Update $Z_i = 1$

Run DFS($\boldsymbol{Z}$)

Update $Z_i = 0$

Run DFS($\boldsymbol{Z}$)

end if

end if

end function

4.1. Experiment Setting

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

4.1.1. benchmark

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

SAA with Empirical Distribution

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

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

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

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

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

Mean-Variance

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

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

where the ambiguity set is

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

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

4.1.2. Data Generation

Suppose we consider a network with m potential warehouses and n demand nodes. We randomly generate dozens of or hundreds of networks, and each network is defined by a six-elements tuple $(\mathcal{W}, \mathcal{R}, \mathcal{A}, \boldsymbol{f}, \boldsymbol{h}, \boldsymbol{\mu})$. $\mathcal{W}$ and $\mathcal{R}$ are the set of potential warehouse nodes and demand nodes. Temporarily, we assume $|\mathcal{W}| = |\mathcal{R}| = 8$. $\mathcal{A}$ is set of links. According to Ni et al. (2018), the average node degree of a typical road network is about 2.4. This is also align with industrial practices in manufacturing, disaster management, and school bus routing design (Jordan and Graves (1995), Mete and Zabinsky (2010), and Bertsimas et al. (2019)). Therefore, we require $|\mathcal{A}| = 1.2(m+n)$. Also, the following two constraints on the generated network should be satisfied: (1) at least one road linked to each supply node (2) at least one road linked to each demand node. These conditions help avoid the appearance of isolated nodes.

$\boldsymbol{f}$ is the fixed setup cost randomly draw from a uniform distribution U[10, 25]. $\boldsymbol{h}$ is the holding cost drawn from U[0.1, 0.2]. And $\boldsymbol{\mu}$ is the first moment, i.e. the mean value, of demand, drawn from U[400, 600]. Initially, we set the correlation coefficient $\rho = 0.3$, coefficient of variation $c_v = 0.3$, and risk attitude coefficient $\kappa = 1.0$, all of which will be further modified to conduct sensitivity analysis. Therefore, the standard deviation of each demand node is $\sigma_j = \mu_j c_v, \forall j \in [n]$. For simplicity, we assume the correlation parameter ρ and coefficient of variation c_v are applied to all demand node pairs or demand node, which means the covariance of any two demand nodes (i, j) is $\mu_i \mu_j c_v^2 \rho$, $\forall i, j \in \mathcal{R}$. We further assume the underlying demand distirbution follows a multinormial gaussian distribution, whose domain is defined as $\mathbb{R}^n_+$, the mean value is $\boldsymbol{\mu}$, and the covariance matrix is $\boldsymbol{\Sigma}_{\rho,c_v}$ as defined above. We use the subscripts $_{\rho,c_v}$ to emphasize that the covariance matrix depends on ρ and c_v.

For model (FOND), we suppose the first-moment $\boldsymbol{\mu}$ and second-moment matrix $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_{\rho,c_v} + \boldsymbol{\mu}\boldsymbol{\mu}^T$ are known to decision makers. For model (MV), only mean and variance of demand are known. And for model

(SAA), we assume there are 30 historical samples available. The sample size is statistically large enough to obtain an unbiased estimation of first-two moments for (CO) and (MV).

After obtaining optimal solutions through solving three models, we randomly generate 2000 demand realizations to conduct the second-stage simulations. 1000 in-sample demand realizations are drawn from underlying true distribution, while another 1000 out-sample realizations are drawn from a mixture distribution with four independent and equally weighted components, including two-point distribution $\mathbb{P}(\tilde{\boldsymbol{d}} = \boldsymbol{\mu} - \boldsymbol{\sigma}) = \mathbb{P}(\tilde{\boldsymbol{d}} = \boldsymbol{\mu} + \boldsymbol{\sigma}) = 0.5$, element-wise independent uniform distribution $U\left[\boldsymbol{\mu} - \sqrt{3}\boldsymbol{\sigma}, \boldsymbol{\mu} + \sqrt{3}\boldsymbol{\sigma}\right]$, independent normal distribution $d_j \sim N(\mu_j, \sigma_j), \forall j \in [n]$, and log-normal distribution with shape parameter 0.3 and scale parameter $\boldsymbol{\mu}$. It is easy to check that the mixture distribution has first-two moments approximately around true $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. We seperately analyze results on in-sample and out-sample realizations, and for each network, we regard the average (or worst-case) result on 1000 second-stage simulations as the corresponding network's performance.

4.2. (8, 8) Network

4.2.1. (ρ, c_v, κ)=(0.3, 0.3, 1.0)

Following the initial setting as depticted in the above subsection, we first restrict ourself to the (8, 8) structure for a closer look at the model's performance. We randomly generate 200 networks with $(\rho, c_v, \kappa) = (0.3, 0.3, 1.0)$. Except furthur explanations, all analysis in this subsection rely on the 200 networks: for each network, we construct a measurement on results of simulation, for example the average unmet demand, and then take average on 200 networks.

First-stage Deployment

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

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)

the average total inventory of 200 networks. The number in the parenthesis shows the standard deviation.

It is clear to see that Model (CO) result in the highest inventory level while Model (MV) has the lowest inventory level. The difference comes from whether the strong correlation of demand is taken into account. The second column, *# of w.h.*, reveals the number of established warehouses on average. Model (SAA) selects almost half of potential locations to build warehouses. The third columns is the amount of roads in designed networks, which implies that Model (CO) designs the most sparse network. The last two columns exhibit supply nodes' and demand nodes' degree repsectively.

Cost

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

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

			In-sample		Out-sample	
	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)

under different models with in-sample and out-sample realizations respectively. The number in parenthesis represents the standard deviation. First two columns represent the setup cost $\boldsymbol{f}$ and the holding cost $\boldsymbol{h}$. *In-sample* means that demand realizations in the simulation are drawn from the genuine gaussian distribuiton while *Out-sample* implies demand realizations are generated according to the mixture distribution described above. $\boldsymbol{p}$ is the penalty cost, or equivalently, the number of unsatisfied demand. *total cost* is the aggregated cost. It is clear that Model (CO) always not only achieves the lowest total cost but also has the smallest standard deviation. One remarkable decreasing appears in $\boldsymbol{p}$. It decreases from about 64.5 (by SAA) to 53.1 (by CO), i.e. 17.7% decreasing in unmet demand under the in-sample simulation. For out-sample simulation, the decreasing is also astounding, from 70.0 to 57.3, about 18.4%. Although the absolutely decreasing value is marginal, the relative decreasing is remarkable, which underlines the advatanges of our proposed conic model and reformulation technique: we can further significantly reduced the amount of unmet demand even it is already small. This managerial insight is quiet attractive, especially to disaster management experts, or fulfillment-oriented warehouse managers, since their first priority is reducing unmet demand.

We next scrutinize the unmet demand in a deeper way. Figure 1 shows the histograms about average

unmet demand, where the horizontal axis shows the average unmet demand obtaining through simulations, and the vertical axis represents the number of networks falling in the corresponding bins. In the left graph,

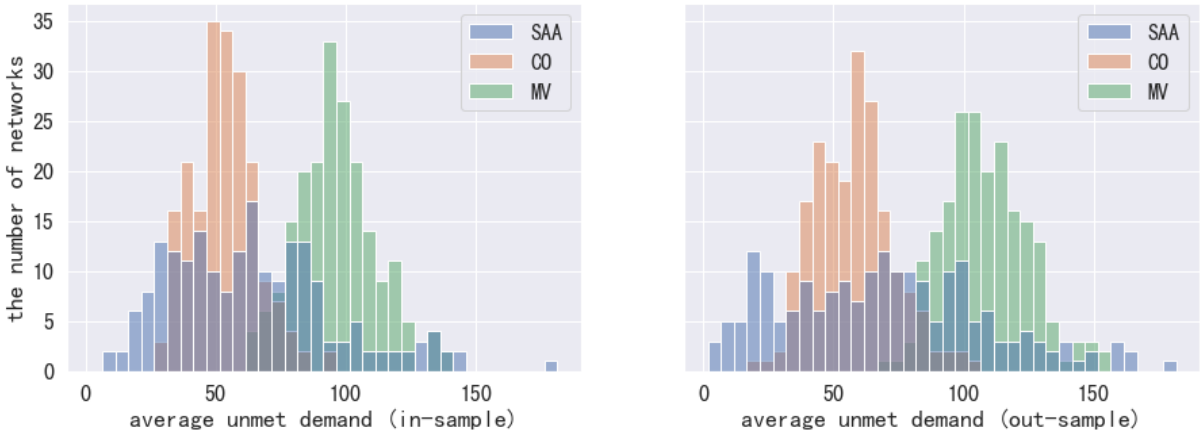

Figure 1: Histogram on average unmet demand, $(\rho, c_v, \kappa) = (0.3, 0.3, 1.0)$

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

Service Level

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

Value-at-Risk

Furthur more, we calculate the 95%, 90%, 85% and 80% quantile of unmet demand to show the ability of Model (CO) to hedge long-tail risk. When the true distribuiton is known to decision makers, these

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

	In-sample		Out-sample	
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

quantiles could be regarded as estimators of the value-at-risk (VaR) of unmet demand after implementing the corresponding first-stage solutions. Table 4 shows the VaR values under different quantiles. *In-sample VaR* shows that, Model (CO) dominates other models under all circumstances, which underlines its ability to hedge long-tail risk comes from the underlying true distribuiton. Surprisingly, the columns *Out-sample VaR*

Table 4: Unmet Demand with Various Quantiles

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

in Table 4 further reveals that Model (CO) is robust enough to the misspecification in distribuiton. In terms of 95% quantile, Model (CO)'s amount of unmet demand of *out-sample* simulation is 20% lower than that of Model (SAA), while the gap is only 13.6% under the *in-sample* situation. This phnomenon complementarily verify the advantages of Model (CO) and distributionally robust techinique, especially when historical data is limited, and misspecification in distribuiton is likely to happen.

CPU Time

Computational time is also essential in practice. Although the problem we consider is strategy-level, a reasonable computational time is indeed necessary. Therefore, we examine the cpu time for solving each individual optimization problem, underlining the superiority of proposed branch and bound acceleration algorithm. Please note that, since we run experiments with four parallel tasks synchronously on one computer, the cpu time will be slightly extended. Table 5 summarizes the time needed to solve each model. We use *(acce)* to represent the model is solved with accelerated branch and bound technique proposed in Section 3.3. The first columns named as *CPU Time (s)* reports computational time in seconds to obtain the optimal solution, and the second columns, *Node*, exhibits the number of visted branch and bound node before terminating the algorithm. Obviously, for Model (CO), the accelerated algorithm reduces more than

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

half time by selecting the better branch, finding a tigher bound, and visting less nodes. The decreasing computational time also demonstrates that involving more variables into reformulated model in Proposition 1 is worthy.

4.2.2. Sensitivity Analysis

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

Correlation Parameter ρ

We first show sensitivity analysis on correlation parameter ρ. We modify it from 0.05 to 0.5. Figure 2 shows

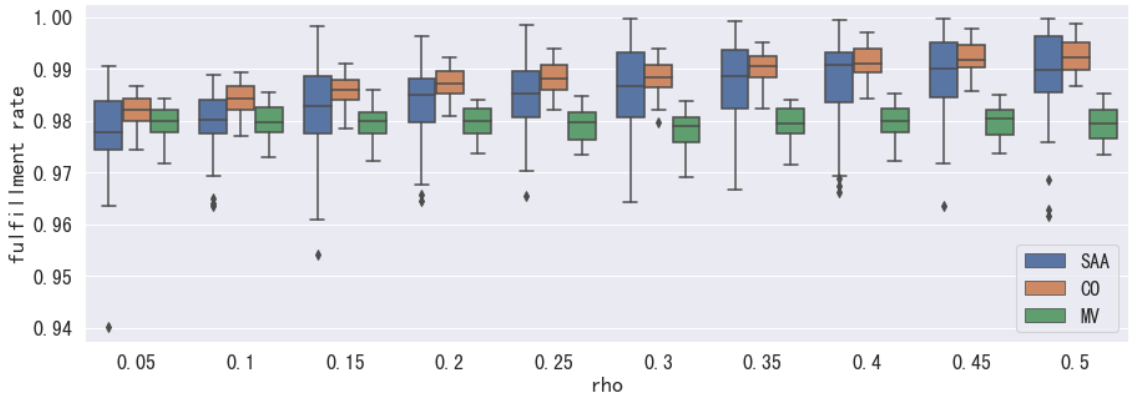

Figure 2: Sensitivity Analysis on ρ

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

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

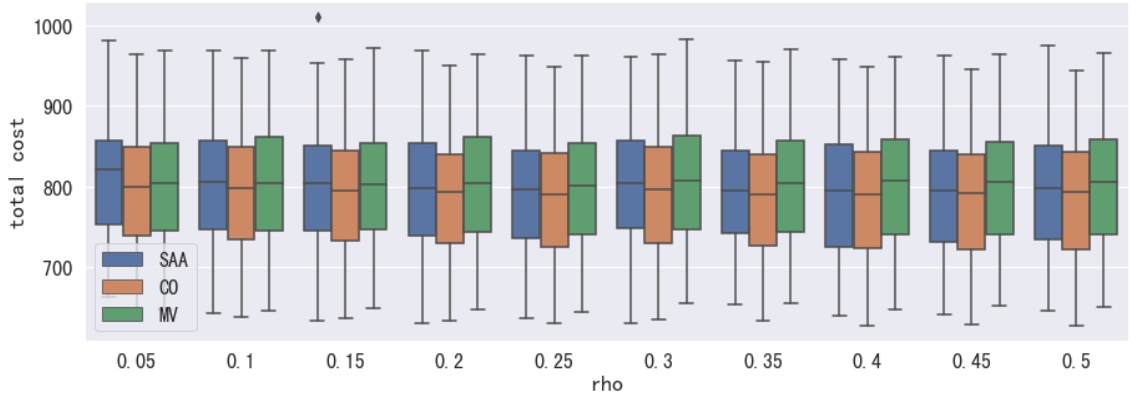

Figure 3: Sensitivity Analysis on ρ

lowest total cost.

Coefficient of Variation c_v

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

Risk Attitude Parameter κ

Figure 6 represents the relationship between fulfillment rate and risk attitude κ. As κ grows, which implies the decision maker focuses more one first-stage cost, the fulfillment rate decreases significantly. This is a direct result of changing risk attitude, since larger κ reflects a more aggressive attitude to future risk. And therefore, corresponding prepositioning decision would become less conservative. Another noteworthy phnomenon is the diminishing gap between (CO) and (SAA) as κ increases. Since the most concerned cost

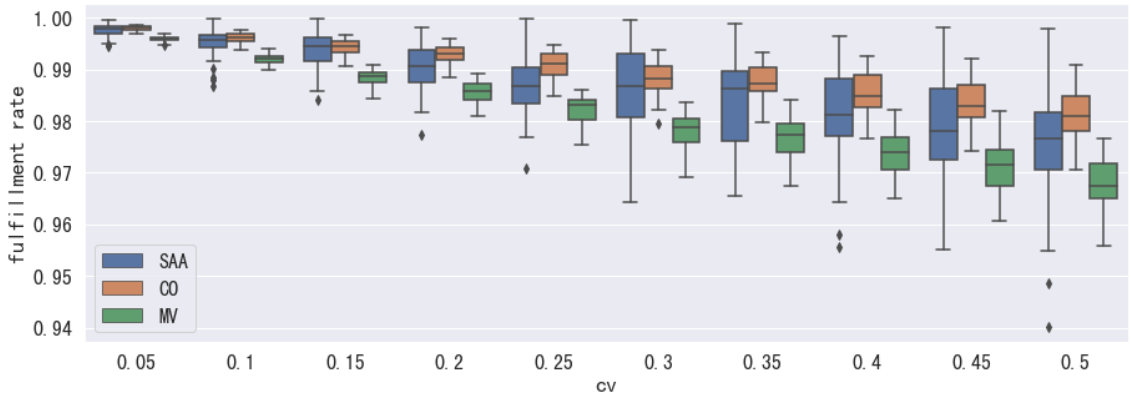

Figure 4: Sensitivity Analysis on c_v

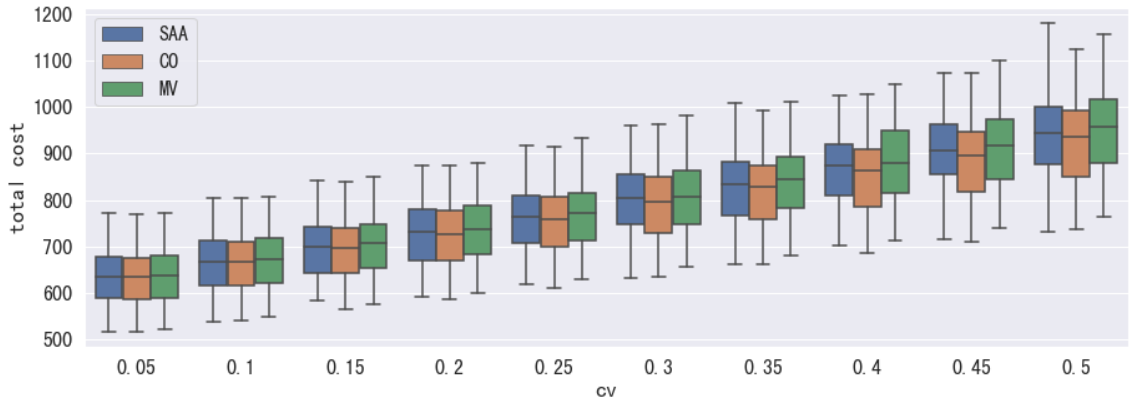

Figure 5: Sensitivity Analysis on c_v

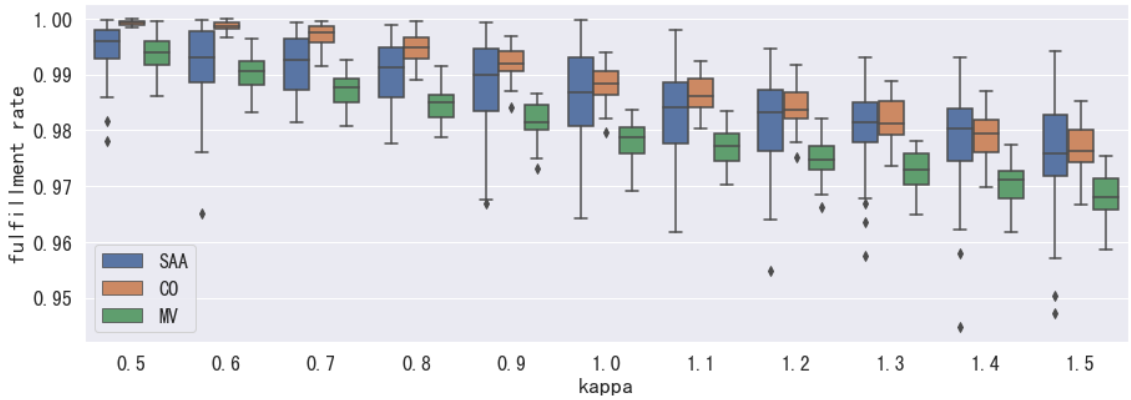

Figure 6: Sensitivity Analysis on κ

gradually shift to deployment cost, the advantage of (CO) in dealing with demand uncertainty is gradually cannibalized. Therefore, (CO)'s superority in fulfillment rate diminishes. Figure 7 expresses total costs of

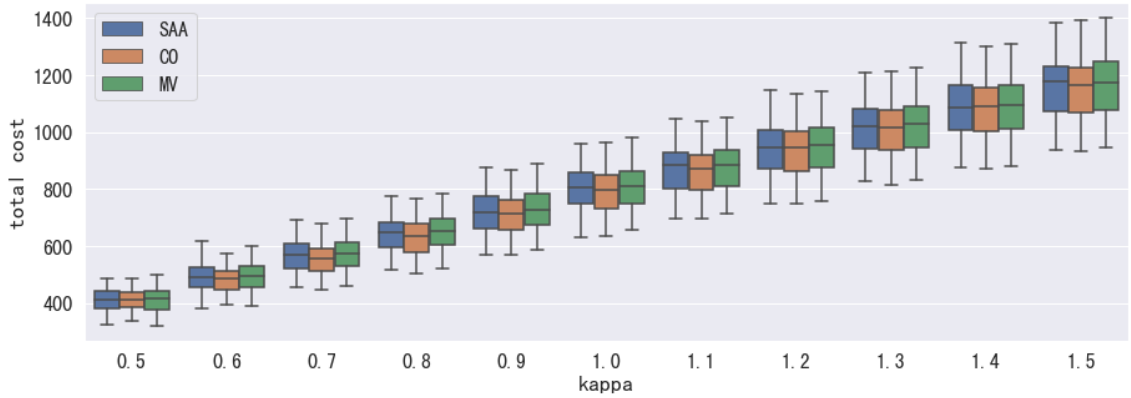

Figure 7: Sensitivity Analysis on κ

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

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

4.3. General Network

In this subsection, we conduct more simulation experiments on general unbalacend networks, and compare these results with $(8, 8)$ networks. More specifically, we consider $(4, 8)$ and $(8, 12)$ respectively. Except the number of potential supply nodes and that of demand nodes, other parameters remain the same. We also modify three parameters, including correlation parameter ρ, coefficient of variation c_v, and risk attitude parameter κ to conduct sensitivity analysis. Generally speaking, we re-run all the above mentioned experiment in this section but on different networks. Table 6 summarizes fulfillment rate on different networks with different parameters.

4.4. Specific Network Structure

1. JG 1995 2. k-chain

Table 6: Total cost comparison on general networks

fulfillment rate	In-sample Simulation									Out-sample Simulation								
	(4, 8)			(8, 8)			(8, 12)			(4, 8)			(8, 8)			(8, 12)		
(rho, cv, kappa)	SAA	CO	MV	SAA	CO	MV	SAA	CO	MV	SAA	CO	MV	SAA	CO	MV	SAA	CO	MV
0.30 0.05 1.0																		
0.30 0.10 1.0																		
0.30 0.15 1.0																		
0.30 0.20 1.0																		
0.30 0.25 1.0																		
0.30 0.30 1.0																		
0.30 0.35 1.0																		
0.30 0.40 1.0																		
0.30 0.45 1.0																		
0.30 0.50 1.0																		
0.05 0.30 1.0																		
0.10 0.30 1.0																		
0.15 0.30 1.0																		
0.20 0.30 1.0																		
0.25 0.30 1.0																		
0.30 0.30 1.0																		
0.35 0.30 1.0																		
0.40 0.30 1.0																		
0.45 0.30 1.0																		
0.50 0.30 1.0																		
0.30 0.30 0.5																		
0.30 0.30 0.6																		
0.30 0.30 0.7																		
0.30 0.30 0.8																		
0.30 0.30 0.9																		
0.30 0.30 1.0																		
0.30 0.30 1.1																		
0.30 0.30 1.2																		
0.30 0.30 1.3																		
0.30 0.30 1.4																		
0.30 0.30 1.5																		

References

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

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

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

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

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

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

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.

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

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

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

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

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

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

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

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

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

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

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.

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.

Appendix A. Proofs

Appendix A.1. Proof for Proposition 1

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

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

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

$$
\begin{aligned}
g(\boldsymbol{I}, \tilde{\boldsymbol{d}}) = \quad \min_{\boldsymbol{x}} \quad & \sum_{j \in [n]} \tilde{d}_j - \sum_{(i,j) \in \mathcal{E}(\boldsymbol{Z})} x_{ij} - \sum_{(i,j) \notin \mathcal{E}(\boldsymbol{Z})} x_{ij} && \text{(A.2)} \\
s.t. \quad & \sum_{i:(i,j) \in \mathcal{E}(\boldsymbol{Z})} x_{ij} + \sum_{i:(i,j) \notin \mathcal{E}(\boldsymbol{Z})} x_{ij} \leq \tilde{d}_j, \forall j \in [n] && \text{(A.3)} \\
& \sum_{j:(i,j) \in \mathcal{E}} x_{ij} \leq I_i, \forall i \in \boldsymbol{Z}^+ \\
& \sum_{j:(i,j) \in \mathcal{E}} x_{ij} \leq 0, \forall i \notin \boldsymbol{Z}^+ && \text{(A.4)} \\
& x_{ij} \geq 0, \forall (i,j) \in \mathcal{E}
\end{aligned}
$$

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

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

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

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

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

Combining above two statements, we finished the proof. □

Appendix A.2. Proof for Proposition 2

Proof. We prove Proposition 2 by equivalently reformulating (4) into a minimizing problem with only one constraint. And conducting optimization on the unique constraint has been proved to be NP-hard in Bertsimas et al. (2010).

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

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

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

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

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

Appendix A.3. Proof for Proposition 3

Proof.

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

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

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

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

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

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

Similar transformation can be found in Kong et al. (2013), Yan et al. (2018), and Kong et al. (2020).

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

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

$$
\begin{aligned}
L_{CO}(\boldsymbol{I}) = \quad & \min_{\alpha_i,\beta_i,\theta_j,\tau,\boldsymbol{\xi},\boldsymbol{\varphi},\boldsymbol{\psi},\boldsymbol{\eta},\boldsymbol{w}} \sum_{i\in[M]} b_i\alpha_i + b_i^2\beta_i + \tau + 2\boldsymbol{\mu}^T\boldsymbol{\xi} + \boldsymbol{\Sigma} \bullet \boldsymbol{\eta} \\
s.t. \quad & \sum_{i\in[M]} \boldsymbol{a}_i\alpha_i - \sum_{j\in[N]} \boldsymbol{e}_{(j)}\theta_j - 2\boldsymbol{\varphi} = \begin{bmatrix} \boldsymbol{0}_{n\times 1} \\ -\boldsymbol{I}_{m\times 1} \\ \boldsymbol{0}_{M\times 1} \end{bmatrix} && \text{(A.10)} \\
& \sum_{i\in[M]} \boldsymbol{a}_i\boldsymbol{a}_i^T\beta_i + \sum_{j\in[N]} \boldsymbol{e}_{(j,j)}\theta_j - \boldsymbol{w} = \boldsymbol{0} && \text{(A.11)} \\
& -2\boldsymbol{\psi} = \begin{bmatrix} \boldsymbol{e}^{[n]} \\ \boldsymbol{0} \end{bmatrix} && \text{(A.12)} \\
& \begin{pmatrix} \tau & \boldsymbol{\xi}^T & \boldsymbol{\varphi}^T \\ \boldsymbol{\xi} & \boldsymbol{\eta} & \boldsymbol{\psi}^T \\ \boldsymbol{\varphi} & \boldsymbol{\psi} & \boldsymbol{w} \end{pmatrix} \succeq_{co} \boldsymbol{0} && \text{(A.13)}
\end{aligned}
$$

α_i, β_i, and θ_j is the dual variable for constraint (A.6), (A.7), and (A.8) respectively. Variable $\begin{pmatrix} \tau & \boldsymbol{\xi}^T & \boldsymbol{\varphi}^T \\ \boldsymbol{\xi} & \boldsymbol{\eta} & \boldsymbol{\psi}^T \\ \boldsymbol{\varphi} & \boldsymbol{\psi} & \boldsymbol{w} \end{pmatrix}$ is the dual variables for completely positive cone constraint (A.9). By weak duality theorem, $L_{CO}(\boldsymbol{I}) \geq L_{CP}(\boldsymbol{I})$. Surprisingly, $L_{CO}(\boldsymbol{I})$ is equivalent to $L_{CP}(\boldsymbol{I})$.

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

according to Yan et al. (2018), if the non-slackness variables (we say *reduced solutions*) of the inner allocation problem has strictly feasible solution in $\mathcal{L}_{lp}^{\dagger}$, and a set of reduced feasible solution can span to $\mathbb{R}^{m+n}$, and has solution $(\hat{\boldsymbol{u}}, \boldsymbol{v}) > 0$, as well as the moment info matrix is in CP cone; then COCone problem = CPCone problem.

Step 4: Combine with first-stage problem

We further combine $L_{CO}(\boldsymbol{I})$ with first-stage problem and replace a part of decision variables in (A.13) with equations (A.10), (A.11), and (A.12). Now, we obtain the (FOND) problem in a Mixed-Integer Copositive Cone form, i.e. problem (CO), which we are fond of.

□