



**HAL**  
open science

## An alternating direction method of multipliers for optimizing $(s, S)$ policies in a distribution system with joint replenishment volume constraints

Bo Dai, Haoxun Chen, Yuan Li, Yidong Zhang, Xiaoqing Wang, Yuming Deng

### ► To cite this version:

Bo Dai, Haoxun Chen, Yuan Li, Yidong Zhang, Xiaoqing Wang, et al.. An alternating direction method of multipliers for optimizing  $(s, S)$  policies in a distribution system with joint replenishment volume constraints. *Omega*, 2023, 116, pp.102800. 10.1016/j.omega.2022.102800 . hal-04429492

**HAL Id: hal-04429492**

**<https://utt.hal.science/hal-04429492v1>**

Submitted on 20 Feb 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# An Alternating Direction Method of Multipliers for Optimizing $(s, S)$ Policies in a Distribution System with Joint Replenishment Volume Constraints

Bo Dai<sup>1,2</sup>, Haoxun Chen<sup>2\*</sup>, Yuan Li<sup>3</sup>, Yidong Zhang<sup>3</sup>, Xiaoqing Wang<sup>3</sup>, Yuming Deng<sup>3</sup>

<sup>1</sup>School of Management, Hunan University of Technology and Business, Changsha 410205, China,  
dr\_bo\_dai@163.com

<sup>2</sup>Logistics and Optimization of Industrial Systems, Laboratory of Computer Science and Digital Society (LIST3N), University of Technology of Troyes, 12 rue Marie Curie, CS 42060, 10004 Troyes, France,  
haoxun.chen@utt.fr

<sup>3</sup>Alibaba Supply Chain Platform (ASCP), Alibaba (China) Co., Ltd, Hangzhou 311121, China,  
yuan.lya@alibaba-inc.com, tanfu.zyd@alibaba-inc.com, robin.wxq@alibaba-inc.com,  
yuming.dym@alibaba-inc.com

## ABSTRACT

In this paper, we study a two-echelon distribution system in which multiple products are jointly replenished at each stocking location and the inventory of each product at each location is controlled by an  $(s, S)$  policy. The transportation capacity in volume of products for each joint replenishment is limited, and linear rationing policies are used for both on-hand inventory and transportation capacity allocation in the system in case of lack. We propose a novel scenario-based model for the optimization of the  $(s, S)$  policies in the system that considers the rationing policies. Because of its high complexity when the number of scenarios is large, an Alternating Direction Method of Multipliers is proposed to solve the model. Based on real data, forty instances were generated and tested to evaluate the model and the solution method. Our numerical experiments show that for these instances this method could find a better solution in a much shorter computation time compared with CPLEX 12.9, whereas the latter often runs out of memory for large-size instances on a personal computer. Moreover, the inventory policies found by this scenario-based optimization approach can reduce costs by 5.1% and improve fill rates by 9.7% on average compared with those currently used in Alibaba.

**Keywords:** Inventory management; distribution system; joint replenishment;  $(s, S)$  policy; scenario optimization; e-commerce

## 1 INTRODUCTION

In the era of e-commerce today, the competition between e-retailers becomes more and more fierce. To increase market shares and gain competitive advantages over their adversaries, e-commerce companies have been making efforts to improve the inventory management in their supply chains. Effective inventory management can help these companies reduce their logistics costs while assuring high service levels to their customers.

In recent years, e-commerce has been developing very fast in China, with the emergence of some e-commerce giants such as Alibaba and JD.com. Under a project of Alibaba Innovative Research (AIR), we investigate the optimization of inventory policies in a distribution system dedicated to on-line Tmall supermarket. This system consists of a central distribution center

---

\*Corresponding author: Haoxun Chen, Email: haoxun.chen@utt.fr

(CDC) and multiple regional distribution centers called front distribution centers (FDCs) in Alibaba. The CDC is the supplier of all FDCs, which serve final customers directly, while the CDC's suppliers are manufacturers. This system stocks and sells multiple products, which are jointly replenished at each stocking location. The customer demand of each product observed by each FDC in each period (day) is stochastic. An  $(s, S)$  policy is adopted for the inventory control of each product at each stocking location in the system. Because of limited transportation capacity in each distribution channel, the volume of all products in each joint replenishment cannot exceed a given maximum volume. Previously, we studied safety stock policies and inventory replenishment planning for a single period in such a system (Dai et al., 2019 and Dai et al., 2021). In this paper, we focus on the optimization of  $(s, S)$  policies in the system. The objective is to minimize the total cost of the distribution system composed of ordering costs, holding costs and lost sales costs, with the maximum joint replenishment volume constraints.

To our best knowledge, the problem considered in this paper was never studied in the literature except for a conference paper (Dai et al., 2018). However, the optimization model proposed in that paper does not consider stock allocation/rationing at the CDC and no algorithm was proposed for solving the model except for commercial solver CPLEX.

Because of its high complexity, this problem is unlikely to be solved analytically or by a commercial solver like CPLEX in an acceptable computation time. Because of this, we propose a novel scenario and decomposition-based approach to solve it. In contrast to the work of Noordhoek et al. (2018), we consider a distribution system with joint inventory replenishment subject to transportation volume constraints rather than with inventory replenishment of a single product. Moreover, our optimization approach is mathematical programming model-based rather than simulation-based. Although scenarios in our model correspond to realizations of stochastic demands in simulation, our approach has an advantage over theirs by allowing the evaluation of the quality of its solution based on the relative gap between an upper bound and a lower bound obtained by the corresponding mathematical programming model.

For this scenario optimization, the main challenge for establishing a deterministic equivalence model for our stochastic inventory policy optimization problem is to formulate the  $(s, S)$  policy used in each stocking location and the stock allocation policy/rationing rule used at the CDC. We first build a novel scenario optimization model for this system, which takes account of the rationing rule and all side constraints. Because solving the model optimally by a commercial solver like CPLEX is quite time consuming for instances with a large number of scenarios, we propose an Alternating Direction Method of Multipliers (ADMM), which is a decomposition approach, to solve the model near-optimally and quickly. Our numerical experiments on four sets of instances generated based on real data show that for these instances this method could find better solutions in a much shorter computation time compared with CPLEX 12.9 and could avoid the out-of-memory problem of CPLEX. Moreover, the inventory policies found by our method can reduce costs and improve fill rates compared with those currently used in Alibaba.

The rest of this paper is arranged as follows. The literature related to our study is reviewed in Section 2. Section 3 introduces the distribution system considered and the inventory optimization problem we study. A scenario-based optimization model is formulated for the problem in Section 4. Section 5 describes a decomposition-based ADMM approach for solving this model. The performance of this approach is evaluated by numerical experiments and its inventory policies found are compared with those currently used in Alibaba in Section 6. Section 7 summarizes the results of this paper and suggests directions for further study.

## 2 LITERATURE REVIEW

In this section, we review the literature related to our work. Inventory management gained many attentions in the past decades, but most of them dealt with a single stocking location (Veinott, 1966; Lin et al., 2020; Teunter and Kuipers, 2022; Qiu et al., 2022).

Previous studies of inventory management of distribution systems were focused on so called ‘one warehouse multi-retailer system’ with a single product (Federgruen and Zipkin, 1984; Axsater, 1990; Axsater, 2003a; Axsater, 2003b; Wang et al., 2022; Berling et al., 2023). Under the balance assumption, Diks and De Kok (1998) proved it is optimal for a distribution system to control each stocking location by a base-stock policy when ordering costs at each stocking location are negligible. However, the balance assumption may not hold and ordering costs may exist in the system. In either of the two cases, the distribution system’s optimal inventory policy is not known. Because of this, most studies assume a given type of inventory policy for each stocking location in a distribution system and try to optimize the parameters of these policies. One challenge for this optimization is that, except for determining the optimal inventory policy parameters, a stock allocation (rationing) decision must be made for each stocking location when its on-hand inventory cannot completely meet all replenishment orders of its immediate downstream locations. Various stock allocation policies/rationing rules were proposed for distribution systems (Eppen and Schrage, 1981; Van der Heijden et al., 1997; Bollapragada et al., 1998). All these rules belong to the class of linear rationing rules (Lagodimos and Koukoumialos, 2008). Each linear rationing rule allocates the shortage of an upstream stocking location to its immediate downstream stocking locations proportionally according to their rationing fractions. One application of rationing rules in a distribution system can be found in Jula and Leachman (2011). Chen et al. (2022) reviewed several stock allocation policies/rationing rules studied in the literature and proposed two nonlinear rationing rules for distribution systems controlled by  $(s, S)$  policies. They showed that the nonlinear rationing rules could reduce the total cost of a distribution system by 1.5% on average. Although a nonlinear rationing rule may outperform a linear one, the former is usually much more complicated than the latter. As a result, linear rationing rules are usually used in practice because of their simplicity and near-optimality.

Joint inventory replenishment is an industrial practice. It has been proved that joint replenishment can reduce ordering costs when multiple products are involved in an inventory system (Goyal and Satir, 1989; Abouee-Mehrzi, 2015). Readers can refer to a recent review paper for the studies on joint replenishment (Bastos et al., 2017). In most joint replenishment models, ordering costs of multiple products consist of product-independent major ordering costs and product-dependent minor ordering costs (Khouja and Goyal, 2008). Various joint replenishment policies were proposed in the literature, such as can-order policy, periodic  $(s, S)$  policy,  $(Q, S)$ , and  $(T, S)$  policy. The inventory policies considered in this paper can be considered periodic  $(s, S)$  policies (Viswanathan, 1997) where the review period (or review interval) of each stocking location is given. Most studies on joint replenishment only consider a single stocking location with only few exceptions. Axsäter and Zhang (1999) studied a continuous-review two-echelon distribution system composed of a central warehouse and multiple identical retailers, where the retailer who has the lowest inventory position orders a batch when the aggregate inventory position of all retailers in the system reaches a joint reorder point. Zhou et al. (2012) considered a multi-product multi-echelon inventory system composed of a supply network and a distribution network linked through a producer, where the inventory of each stocking location is controlled by a  $(T, S)$  policy. A mathematical programming model is established for optimizing the order cycles and the order up to levels of all products at all stocking locations. However, this model is only an approximate model, because the holding

cost and the shortage cost of each stocking location are only approximately formulated. Moreover, the model is only approximately solved by using a genetic algorithm, so our modeling and solution method are both different from theirs. Çapar (2013) studied joint shipment consolidation in a two echelon distribution system, where the distribution center uses a time-based shipment consolidation policy to dispatch to retailers their accumulated orders at the end of each consolidation cycle. An exact method was developed to determine the optimal replenishment quantity for the distribution center and the base-stock level for each retailer. However, only one product is involved in the system and the joint shipment consolidation of a single product is different from the joint replenishment of multiple products we study. Carvajal et al. (2020) studied joint inventory replenishment in a two-echelon distribution system under constant final demands, budgetary and storage capacity constraints. They proposed a hybrid meta-heuristic algorithm and a memetic algorithm for optimizing the system. However, they considered a distribution system with constant demands, whereas we consider a distribution system with stochastic demands.

Only few papers studied the optimization of an inventory system with multiple stocking locations and controlled by  $(s, S)$  policies (De Kok et al., 2018). Kukreja and Schmidt (2005) considered a single echelon continuous review inventory system with multiple stocking locations controlled by  $(s, S)$  policies, where lateral transshipments among stocking locations are allowed. Fattahi et al. (2015) investigated a serial system with a manufacturer and a retailer controlled by  $(s, S)$  policies. They considered maximum production capacity in the manufacturer and maximum ordering capacity in the retailer. The systems studied in these papers are simpler than distribution systems and neither of them considers joint inventory replenishment. For distribution systems, the optimization of  $(R, Q)$  policies or  $(R, nQ)$  policies was also studied in the literature (Axsäter, 2003a, Al-Rifai and Rossetti, 2007, Berling and Marklund, 2014). All these papers only consider one product without joint replenishment and assume identical review period (the time between two consecutive reviews) for all stocking locations. Moreover, they did not explicitly consider the stock allocation policy/rationing rule used in each upstream stocking location of a distribution system.

Three kinds of methods have been used to optimize inventory policies of a stochastic inventory system in the literature, i.e., analytical, simulation-based, and scenario-based methods. Among them, most analytical approaches were used for the optimization of base-stock policies (Clark and Scarf, 1960), or  $(R, Q)$  policies in one warehouse multi-retailer systems (Forsberg, 1996, Axsäter, 2000). However, using analytical methods to optimize  $(s, S)$  policies for a multi-echelon system was rare because of its high complexity. Simulation-based optimization approaches use a metaheuristic as optimization engine and simulation for cost evaluation of an inventory system. Jalali and Nieuwenhuyse (2015) provided an overview of simulation-based approaches for inventory optimization. Tsai and Chen (2017) considered multiple objectives and used a simulation-based approach to optimize the reorder points and the order quantities in a two-echelon serial system. The three objectives considered are the expected inventory cost, inventory level, and frequency of shortage. Noordhoek et al. (2018) developed an approach that combines scatter search and simulation for the optimization of  $(s, S)$  policies in a multi-echelon distribution system of Dutch food retail industry. Numerical results show this approach outperforms a nested bisection search method. Avci and Selim (2018) proposed a simulation-based approach for the optimization of an inventory system in an automotive supply chain. They considered multiple objectives and designed a heuristic algorithm to optimize the objectives simultaneously. Scenario-based methods were also used in inventory optimization. Fattahi et al. (2015) formulated the problem of optimizing inventory policies in a single product serial inventory system with a retailer and a manufacturer as a mixed-integer linear program

using scenario approximation. The model was solved by applying either an evolutionary or imperialist competitive algorithm. Nguyen and Chen (2018) formulated a stochastic programming model for supplier selection in a biomass supply chain with uncertain supply. The deterministic equivalence model of the stochastic program was solved by applying a regularized L-shaped algorithm.

Because only a meta-heuristic is used as its optimization engine, the quality of the solution found by a simulation-based approach for the optimization of a stochastic inventory system cannot be well evaluated. This is one major weakness of such approach. Although each scenario corresponds to a realization of random variables in a stochastic inventory system as in simulation, the quality of the solution found by a scenario-based approach can be well evaluated, because for a given number of scenarios, the stochastic optimization problem can be equivalently transformed into a deterministic mathematical program. This is one advantage of scenario-based approach versus simulation-based approach.

In the literature, both inventory systems with backorders and lost-sales were widely studied. A recent literature review on inventory systems with lost-sales was given by Bijvank and Vis (2011). They concluded that the optimization of an inventory system with lost-sales is usually much more difficult than the optimization of its counterpart with backorders. For this reason, most studies on inventory systems with lost-sales only considered one stocking location with one product. Only few papers studied distribution systems with lost sales (Andersson and Melchior, 2001; Seifbarghy and Jokar, 2006; Hill et al., 2007). Because lost sales occur in our considered distribution system, we adopt an inventory model with lost sales for this system. In addition, in this system, the inventory review period of the CDC is different from that of each FDC.

With respect to the previous works reviewed above, our work makes the following contributions to the literature.

1. A two-echelon joint replenishment distribution system with lost sales and managed by  $(s, S)$  policies with transportation volume constraints and different review periods at the CDC and the FDCs is considered. Most of previous works on distribution systems consider backorders and no previous work studied a distribution system with all the features we consider.

2. By applying scenario approximation, a novel mixed-integer linear programming model is formulated for the optimization of the inventory policies. This model well considers stock allocation/rationing in the system and all side constraints.

3. An alternating direction method of multipliers is proposed to solve the model. This method can find a near-optimal solution for the instances generated based on real data in a reasonable computation time.

### **3 PROBLEM DESCRIPTION**

A two-echelon distribution system dedicated to on-line Tmall supermarket of Alibaba is studied, which is illustrated in Figure 1. This distribution system consists of a Central Distribution Center (CDC) and multiple regional distribution centers called Front Distribution Centers (FDCs), where the CDC is the supplier of each FDC which serves final customers. Each stocking location in the system stores multiple fast moving products. It is assumed that each product's demand observed by each FDC in each period (day) is normally distributed, and the demands of different products at different stocking locations in different periods are independent. Note that our scenario-based optimization model and solution method are also

applicable in case when the demand of each product at each stocking location in each period is subject to any other distribution such as Poisson or compound Poisson distribution.

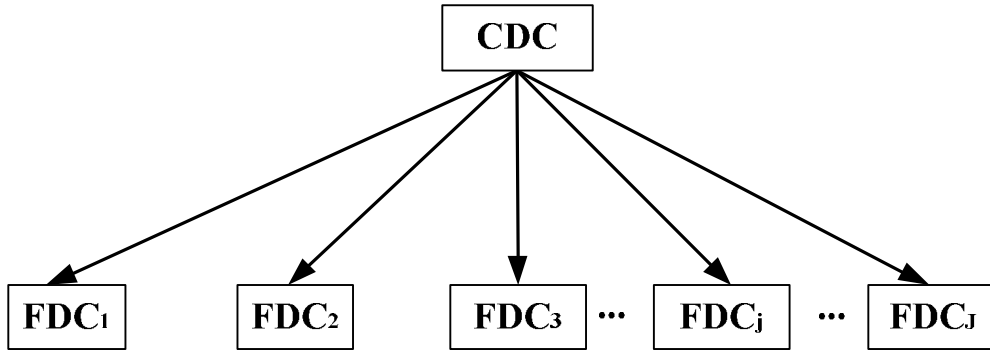


Figure 1: A two-echelon distribution system for Tmall supermarket

The inventory of each product at each stocking location is periodically reviewed, where each period corresponds to one day. All FDCs have the same review period (the time between two consecutive reviews), whereas the CDC's review period is a multiple of each FDC's. Suppose that the review period of the CDC and each FDC are  $R$  and  $1$ , respectively. To distinguish between the two review periods, the CDC's review period is also called review cycle hereafter. Each stocking location's replenishment lead time is given, and the CDC may have a lead time different from that of the FDCs. Each stocking location jointly replenishes multiple products involved. Because of limited transportation capacity for each distribution channel, the volume of each joint replenishment of a stocking location cannot exceed a given maximum volume. The inventory of each product at each stocking location in this distribution system is controlled by an  $(s, S)$  policy, where  $s$  and  $S$  are reorder point and order-up-to level, respectively. In addition, when the on-hand inventory of the CDC cannot satisfy all replenishment orders of the FDCs, a stock allocation/rationing rule is used to allocate the on-hand inventory among the orders.

We want to optimize the  $(s, S)$  policies of the distribution system in a planning horizon starting from period  $r$  and ending with period  $T$ , where  $1 \leq r \leq R$  and  $T = r + N \times R - 1$  for some positive integer  $N$ . The number of periods considered in the planning horizon is  $N \times R$ . In this planning horizon, each FDC reviews its inventory in each period, and the periods in which the CDC reviews its inventory can be written as  $n \times R + 1$ , for integer  $n$  with  $0 \leq n \leq N - 1$  and  $n \times R + 1 \geq r$ . Note that considering a finite horizon in the optimization of a dynamic system is also adopted by lot-sizing models for production planning (Balkhi, 2001; Zhou et al., 2004).

Holding costs, lost sales costs, and two types of ordering costs are considered in the system. Minor ordering cost of each stocking location is determined by transportation costs of its inventory replenishments. Major ordering cost of the CDC is charged every time when it places a joint replenishment order to external suppliers, whereas that cost is negligible for each FDC. The negligibility of major ordering cost for each FDC is because this cost is much smaller than transportation costs for its order delivery. Our objective of optimizing the  $(s, S)$  policies of the distribution system is to minimize its expected total cost in the planning horizon.

The problem studied is a very complex stochastic optimization problem since it deals with a distribution system controlled by  $(s, S)$  policies with stochastic demands of multiple products jointly replenished at each stocking location with maximum joint replenishment volume constraints. Because of the complexity, it is unlikely to find an analytical approach for this

optimization. For this reason, we propose a scenario-based approach for this optimization, where the random demands are approximated by a finite number of their possible realizations (scenarios) with respective occurrence probabilities. For a given number of scenarios, this stochastic optimization problem can be transformed into its deterministic equivalence model. In this equivalence model, the expected total cost of the system is replaced by its total cost averaged over all demand scenarios.

#### 4 MATHEMATICAL MODELLING OF THE PROBLEM

In this section, a mixed integer linear program model is established for optimizing  $(s, S)$  policies in the distribution system considered. Since the demands of the distribution system are relatively stationary, we assume  $s$  and  $S$  for each product at each stocking does not change over time, but the optimization model and the solution method proposed in this paper can be easily extended to ones with time-dependent  $s$  and  $S$ .

Since most stock allocation/rationing rules proposed in the literature and that used in the distribution system of Alibaba are linear ones, we restrict rationing rules used in this system to linear ones. That is, a linear rationing rule is applied to both the allocation of on-hand inventory of the CDC among orders of the FDCs for each product and the allocation of transportation capacity of each distribution channel among multiple products for each stocking location in the system. This restriction will not have a significant impact on the cost of the system (Chen et al. 2022) while making its stock allocation decisions simpler. Like the on-hand inventory allocation, when a linear rationing rule is applied to the transportation capacity allocation, the shortage of transportation capacity is allocated proportionally among the products involved in each joint replenishment according to their rationing fractions pre-specified. We assume all rationing fractions (coefficient) in each linear rationing rule used are given.

Before presenting the model, we first introduce the following notations used in it.

##### *Indices*

$i$ : index of a product,  $i \in I = \{1, 2, \dots, I\}$ , where  $I$  is the number of products considered.

$j$ : index of a stocking location,  $j \in \vartheta = \{0, 1, 2, \dots, J\}$ , where  $j = 0$  represents the CDC and  $j > 0$  represents a FDC, and  $J$  is the number of FDCs in the distribution system considered.

$t, \tau$ : index of a period,  $t, \tau \in T = \{r, r+1, \dots, r+N \times R-1\}$ , where  $r$  with  $1 \leq r \leq R$  is the starting period of the planning horizon,  $R$  is the inventory review period of the CDC,  $N$  is a positive integer, and  $N \times R$  is the number of periods considered in the planning horizon.

$k$ : index of a scenario index,  $k \in K = \{1, 2, \dots, K\}$ , where  $K$  is the number of demand scenarios considered.

##### *Parameters*

$d_{ij}^{t,k}$ : product  $i$ 's demand observed by stocking location  $j$  in period  $t$  under scenario  $k$

$L_{ij}$ : product  $i$ 's replenishment lead time at stocking location  $j$

$R_j$ : inventory review period of stocking location  $j$  with  $R_j = 1$  if  $j \geq 1$  and  $R_0 = R$ . That is, each FDC is reviewed once every period, and the CDC is reviewed once every  $R$  periods,  $R$  is a positive integer



$v_i$  : volume of product  $i$

$C_j^{\max}$  : maximum joint replenishment volume of stocking location  $j$

$h_{ij}$  : unit holding cost of product  $i$  in each period at stocking location  $j$

$l_{ij}$  : unit lost sales cost of product  $i$  at stocking location  $j$

$t_{ij}$  : unit transportation cost of product  $i$  in any joint replenishment of stocking location  $j$

$O_0$  : major ordering cost for each joint replenishment of the CDC

$f_{ij}$  : rationing fraction of FDC  $j$  for on-hand inventory allocation of the CDC among orders of the FDCs for each product  $i$ ,  $\sum_{j=1}^J f_{ij} = 1$ ,  $f_{ij} \geq 0$

$f'_{ij}$  : rationing fraction of product  $i$  for transportation capacity allocation of each joint replenishment of stocking location  $j$  among all products,  $\sum_{i=1}^I f'_{ij} = 1$ ,  $f'_{ij} \geq 0$

$p_k$  : occurrence probability of scenario  $k$

### **Decision Variables**

$IP_{ij}^{t,k}$  : product  $i$ 's inventory position at stocking location  $j$  in period  $t$  prior to receiving shipments from its supplier in this period under scenario  $k$

$I_{ij}^{t,k}$  : product  $i$ 's on-hand inventory at stocking location  $j$  in period  $t$  prior to receiving shipments from its supplier in this period under scenario  $k$

$LS_{ij}^{t,k}$  : lost sales of product  $i$  at stocking location  $j$  in period  $t$  under scenario  $k$

$s_{ij}$  : the reorder point of product  $i$  at stocking location  $j$

$S_{ij}$  : the order-up-to level of product  $i$  at stocking location  $j$

$x_{ij}^{t,k}$  : product  $i$ 's real replenishment quantity for stocking location  $j$  in period  $t$  under scenario  $k$  after both on-hand inventory allocation/rationing of the CDC for this product and the transportation capacity allocation/rationing among all products involved in the corresponding joint replenishment

$q_{ij}^{t,k}$  : product  $i$ 's replenishment quantity for FDC  $j$  in period  $t$  under scenario  $k$ , determined by its  $(s, S)$  policy

$q_{ij}^{u,k}$  : product  $i$ 's replenishment quantity for FDC  $j$  in period  $t$  under scenario  $k$  after on-hand inventory allocation/rationing of the CDC for this product but before the transportation capacity allocation/rationing among all products involved in the corresponding joint replenishment

$y_{ij}^{t,k}$  :  $y_{ij}^{t,k} = 1$  if stocking location  $j$  replenishes its inventory of product  $i$  in period  $t$  under scenario  $k$ , and  $y_{ij}^{t,k} = 0$  otherwise

$y_0^{t,k}$  :  $y_0^{t,k} = 1$  if the CDC makes a joint replenishment in period  $t$  under scenario  $k$ , and  $y_0^{t,k} = 0$  otherwise

With these notations, we can establish the following mixed-integer linear programming model for scenario-based optimization of the inventory policies of the distribution system with lost sales considered.

Model *SBO*:

$$Z^{SBO} = \text{Min} \left( \sum_{k \in K} p_k \cdot \left( \sum_{t \in T} \sum_{i \in I} \sum_{j \in J} (h_{ij} \cdot I_{ij}^{t,k} + t_{ij} \cdot v_i \cdot x_{ij}^{t,k}) + \sum_{i \in T} O_0 \cdot y_0^{t,k} + \sum_{i \in T} \sum_{i \in I} \sum_{j \in J \setminus \{0\}} l_{ij} \cdot LS_{ij}^{t,k} \right) \right) \quad (1)$$

subject to:

$$I_{ij}^{(t+1),k} = I_{ij}^{t,k} + x_{ij}^{(t-L_{ij}),k} - d_{ij}^{t,k} + LS_{ij}^{t,k}, i \in I, j \in J \setminus \{0\}, t \in T, k \in K \quad (2)$$

$$I_{i0}^{(t+1),k} = I_{i0}^{t,k} + x_{i0}^{(t-L_{i0}),k} - \sum_{j=1}^J x_{ij}^{t,k}, i \in I, t \in T, k \in K \quad (3)$$

$$IP_{ij}^{t,k} = I_{ij}^{t,k} + \sum_{\tau=t-L_{ij}}^{t-1} x_{ij}^{\tau,k}, i \in I, j \in J \setminus \{0\}, t \in T, k \in K \quad (4)$$

$$IP_{i0}^{t,k} = I_{i0}^{t,k} + \sum_{\tau=t-L_{i0}}^{t-1} x_{i0}^{\tau,k}, i \in I, t \in T, k \in K \quad (5)$$

$$\text{If } IP_{ij}^{t,k} \leq s_{ij}, y_{ij}^{t,k} = 1, i \in I, j \in J \setminus \{0\}, t \in T, k \in K \quad (6)$$

$$\text{If } IP_{ij}^{t,k} > s_{ij}, y_{ij}^{t,k} = 0, i \in I, j \in J \setminus \{0\}, t \in T, k \in K \quad (7)$$

$$q_{ij}^{t,k} \leq s_{ij} - IP_{ij}^{t,k}, i \in I, j \in J \setminus \{0\}, t \in T, k \in K \quad (8)$$

$$\text{If } IP_{i0}^{t,k} \leq s_{i0}, y_{i0}^{t,k} = 1, i \in I, t = 1 + n \times R, t \geq r, 0 \leq n \leq N-1, k \in K \quad (9)$$

$$\text{If } IP_{i0}^{t,k} > s_{i0}, y_{i0}^{t,k} = 0, i \in I, t = 1 + n \times R, t \geq r, 0 \leq n \leq N-1, k \in K \quad (10)$$

$$q_{i0}^{t,k} \leq s_{i0} - IP_{i0}^{t,k}, i \in I, t = 1 + n \times R, t \geq r, 0 \leq n \leq N-1, k \in K \quad (11)$$

$$y_{i0}^{t,k} \leq y_0^{t,k}, i \in I, t \in T, k \in K \quad (12)$$

$$v_i \cdot x_{ij}^{t,k} \leq y_{ij}^{t,k} \cdot C_j^{\max}, i \in I, j \in J, t \in T, k \in K \quad (13)$$

$$\text{If } \sum_{j=1}^J q_{ij}^{t,k} > I_{i0}^{t,k}, q_{ij}^{t,k} = q_{ij}^{t,k} - f_{ij} \left( \sum_{j=1}^J q_{ij}^{t,k} - I_{i0}^{t,k} \right), i \in I, j \in J \setminus \{0\}, t \in T, k \in K \quad (14)$$

$$\text{If } \sum_{j=1}^J q_{ij}^{t,k} \leq I_{i0}^{t,k}, q_{ij}^{t,k} = q_{ij}^{t,k}, i \in I, j \in J \setminus \{0\}, t \in T, k \in K \quad (15)$$

$$\text{If } \sum_{i=1}^I v_i \cdot q_{ij}^{u,k} > C_j^{\max}, \quad v_i \cdot x_{ij}^{t,k} = v_i \cdot q_{ij}^{u,k} - f'_{ij} \left( \sum_{i=1}^I v_i \cdot q_{ij}^{u,k} - C_j^{\max} \right),$$

$$i \in I, j \in J \setminus \{0\}, t \in T, k \in K \quad (16)$$

$$\text{If } \sum_{i=1}^I v_i \cdot q_{ij}^{u,k} \leq C_j^{\max}, \quad x_{ij}^{t,k} = q_{ij}^{u,k}, i \in I, j \in J \setminus \{0\}, t \in T, k \in K \quad (17)$$

$$\text{If } \sum_{i=1}^I v_i \cdot q_{i0}^{t,k} > C_0^{\max}, \quad v_i \cdot x_{i0}^{t,k} = v_i \cdot q_{i0}^{t,k} - f'_{i0} \left( \sum_{i=1}^I v_i \cdot q_{i0}^{t,k} - C_0^{\max} \right),$$

$$i \in I, t = 1+n \times R, t \geq r, 0 \leq n \leq N-1, k \in K \quad (18)$$

$$\text{If } \sum_{i=1}^I v_i \cdot q_{i0}^{t,k} \leq C_0^{\max}, \quad x_{i0}^{t,k} = q_{i0}^{t,k}, i \in I, t = 1+n \times R, t \geq r, 0 \leq n \leq N-1, k \in K \quad (19)$$

$$S_{ij} \geq s_{ij}, i \in I, j \in J \quad (20)$$

$$s_{ij} \geq 0, i \in I, j \in J \quad (21)$$

$$x_{i0}^{t,k} = 0, i \in I, t \in T, t \neq 1+n \times R, 0 \leq n \leq N-1, k \in K \quad (22)$$

$$IP_{ij}^{t,k}, I_{ij}^{t,k}, LS_{ij}^{t,k}, x_{ij}^{t,k}, q_{ij}^{t,k}, q_{ij}^{u,k} \geq 0; y_{ij}^{t,k}, y_{ij}^{u,k} \in \{0,1\}, i \in I, j \in J, t \in T, k \in K \quad (23)$$

The objective function (1) represents the total cost composed of inventory holding costs at all stocking locations, transportation costs (minor ordering costs) for inventory replenishments of all stocking locations, major ordering costs of the CDC, and lost sales costs at all FDCs for the distribution system considered. Equations (2) and (3) are inventory balance equations of the FDCs and the CDC, respectively. Equations (4) and (5) establish the relationship among inventory position, inventory level, and outstanding orders at each stocking location.

Constraints (6) to (11) ensure that an  $(s, S)$  inventory policy is used at each stocking location. Among them, logical constraints (6) and (7) imply that inventory replenishment of each product at each FDC only occurs when its inventory position drops below its reorder point, constraints (8) imply that each product's replenishment quantity at each FDC does not exceed its order-up-to level minus its inventory position. Similarly, logical constraints (9) and (10) imply that the inventory replenishment of each product at the CDC only occurs when its inventory position drops below its order-up-to level, constraints (11) ensure that the inventory replenishment quantity of each product at the CDC is no more than its order-up-to level minus its inventory position, where constraints (9) to (11) are imposed only for periods  $t = 1+n \times R, t \geq r$  with  $0 \leq n \leq N-1$ , this is because the CDC replenishes its inventory only in its review periods.

Note that, in this model, the real replenishment quantity of a product at some stocking location in some period may be smaller than the order quantity determined by its  $(s, S)$  policy because of on-hand inventory rationing at the CDC and transportation capacity rationing among products jointly replenished.

Constraints (12) imply that any product is replenished at the CDC only if it is one of the products jointly replenished. Constraints (13) link real variable  $x_{ij}^t$  and binary variable  $y_{ij}^t$ .

Logical constraints (14)-(19) describe the inventory rationing at the CDC in each period and the transportation capacity rationing at each distribution channel for each joint

replenishment. For each product  $i$ , the inventory rationing constraints (14) and (15) indicate the replenishment quantity  $q_{ij}^{n,k}$  allocated to each FDC  $j$  determined by a linear rationing rule and according to on-hand inventory of the CDC without considering the maximum joint replenishment volume constraints. This quantity  $q_{ij}^{n,k}$  is a function of the replenishment quantity  $q_{ij}^{t,k}$  determined by an  $(s, S)$  policy, the shortage  $\sum_{j=1}^J q_{ij}^{t,k} - I_{i0}^{t,k}$  of the CDC, and the rationing fraction  $f_{ij}$  for each FDC  $j$ , where  $I_{i0}^{t,k}$  is product  $i$ 's on-hand inventory at the CDC at the moment of rationing.

Note that only the linear inventory rationing equations (14) themselves may generate negative allocation  $q_{ij}^{n,k}$ , when the balance assumption does not hold for the distribution system considered (Diks and De Kok, 1998). This problem is overcome by imposing the non-negativity of  $q_{ij}^{n,k}$  in (23) and allowing  $q_{ij}^{t,k} \leq S_{ij} - IP_{ij}^{t,k}$  even if  $IP_{ij}^{t,k} \leq S_{ij}$ . Only imposing the non-negativity of  $q_{ij}^{n,k}$  may lead to the infeasibility of the model, but this infeasibility can be avoided by relaxing  $q_{ij}^{t,k} = S_{ij} - IP_{ij}^{t,k}$  to  $q_{ij}^{t,k} \leq S_{ij} - IP_{ij}^{t,k}$  in case of  $IP_{ij}^{t,k} \leq S_{ij}$ . This relaxation is reasonable since the real replenishment quantity of a product at some FDC in some period may be smaller than the order quantity determined by its  $(s, S)$  policy as explained above.

The transportation capacity rationing constraints (16) and (17) indicate the real replenishment quantity  $x_{ij}^{t,k}$  of product  $i$  for each FDC  $j$  determined by a linear rationing rule of transportation capacity that considers the maximum joint replenishment volume  $C_j^{\max}$  for each replenishment of the FDC. This quantity is a function of  $q_{ij}^{n,k}$ , the shortage of transportation capacity  $\sum_{i=1}^I v_i \cdot q_{ij}^{n,k} - C_j^{\max}$ , and the rationing fraction  $f'_{ij}$  for each product  $i$ . Similarly, the rationing constraints (18) and (19) indicate the real replenishment quantity  $x_{i0}^{t,k}$  of product  $i$  for the CDC determined by a linear rationing rule of transportation capacity that considers the maximum joint replenishment volume  $C_0^{\max}$  for each replenishment of the CDC, where  $q_{i0}^{t,k}$  is the quantity of product  $i$  replenished by the CDC determined by an  $(s, S)$  policy.

Similarly, only the linear transportation capacity rationing equations (16) or (18) themselves may generate negative allocation  $x_{ij}^{t,k}$  or  $x_{i0}^{t,k}$ . This problem is overcome in the same way as  $q_{ij}^{n,k}$  by imposing the non-negativity of  $x_{ij}^{t,k}$  and  $x_{i0}^{t,k}$  in (23) and allowing  $q_{ij}^{t,k} \leq S_{ij} - IP_{ij}^{t,k}$  generally.

Constraints (20) imply that the order-up-to level of product  $i$  at stocking location  $j$  is always larger or equal than its corresponding reorder point. Since in this inventory model with lost sales, the inventory position  $IP_{ij}^{t,k}$  is never negative, constraints (21) are added for all products  $i$  and all stocking locations  $j$  without loss of generality.

Finally, (22) and (23) indicate the ranges of all decision variables. As a replenishment from its supplier to the CDC in each scenario happens only once every  $N$  periods, and the distribution system starts in the  $r$ -th period in the review cycle of  $N$  periods of the CDC, the replenishment

quantity  $x_{i0}^{t,k} \geq 0$  in the periods  $t = n \times R + 1, t \geq r$  with  $0 \leq n \leq N - 1$  and  $x_{i0}^{t,k} = 0$  in the other periods.

All logical constraints in the above model *SBO* can be equivalently transformed into linear constraints by introducing auxiliary variables and big numbers. After the transformation, this model becomes a MILP model. In the remainder of this section, we only linearize logical constraints (6), (7), (9), and (10) which will be cited and reformulated in the next section when describing a solution approach for the model. For the linearization of other logical constraints and the setting of big numbers introduced in the linearization of all logical constraints in model *SBO*, please see the appendix at the end of this paper.

By introducing big numbers  $M_{ij}^{11}$  and  $M_{ij}^{12}$  for each  $i$  and  $j$ , logical constraints (6) and (7) can be transformed into the following linear constraints (24) and (25).

$$M_{ij}^{11} \cdot (y_{ij}^{t,k} - 1) \leq s_{ij} - IP_{ij}^{t,k}, i \in I, j \in J \setminus \{0\}, t \in T, k \in K \quad (24)$$

$$s_{ij} - IP_{ij}^{t,k} \leq M_{ij}^{12} \cdot y_{ij}^{t,k}, i \in I, j \in J \setminus \{0\}, t \in T, k \in K \quad (25)$$

Similarly, logical constraints (9) and (10) can be transformed into the following linear constraints (26) and (27) by introducing big numbers  $M_i^{21}$  and  $M_i^{22}$  for each  $i$ .

$$M_i^{21} \cdot (y_{i0}^{t,k} - 1) \leq s_{i0} - IP_{i0}^{t,k}, i \in I, t = 1 + n \times R, t \geq r, 0 \leq n \leq N - 1, k \in K \quad (26)$$

$$s_{i0} - IP_{i0}^{t,k} \leq M_i^{22} \cdot y_{i0}^{t,k}, i \in I, t = 1 + n \times R, t \geq r, 0 \leq n \leq N - 1, k \in K \quad (27)$$

## 5 SOLUTION APPROACH

After the linearization of all logical constraints, model *SBO* becomes a mixed integer linear program (MILP). This MILP can be solved by using a commercial MILP solver such as CPLEX. However, the time required by CPLEX to solve the model increases exponentially as the number of its integer variables increases, so when the number of scenarios is large, a more efficient method is required to solve this model. Because when  $s_{ij}$  and  $S_{ij}$  are given for all products and all stocking locations, this model can be decomposed into  $K$  independent sub-models, where  $K$  is the number of scenarios, we try to design a scenario-based decomposition approach to solve it. In order to do so, we first reformulate *SBO* by introducing scenario-dependant variables  $s_{ij}^k$  and  $S_{ij}^k$  to replace  $s_{ij}$  and  $S_{ij}$  and adding constraints

$$s_{ij}^k = \sum_{k \in K} s_{ij}^{k'} / K \text{ and } S_{ij}^k = \sum_{k \in K} S_{ij}^{k'} / K, \text{ for any } k \in K, \text{ where } K \text{ is the number of scenarios in } K.$$

The reformulated model is given as follows:

Model *SBO2*:

$$Z^{SBO2} = \text{Min} \left( \sum_{k \in K} p_k \cdot \left( \sum_{t \in T} \sum_{i \in I} \sum_{j \in J} (h_{ij} \cdot I_{ij}^{t,k} + t_j \cdot v_i \cdot x_{ij}^{t,k}) + \sum_{t \in T} O_0 \cdot y_0^{t,k} + \sum_{t \in T} \sum_{i \in I} \sum_{j \in J \setminus \{0\}} l_{ij} \cdot LS_{ij}^{t,k} \right) \right) \quad (28)$$

subject to constraints (2)-(5), (12)-(13), (22)-(23),

$$M_{ij}^{11} \cdot (y_{ij}^{t,k} - 1) \leq s_{ij}^k - IP_{ij}^{t,k}, i \in I, j \in J \setminus \{0\}, t \in T, k \in K \quad (29)$$

$$s_{ij}^k - IP_{ij}^{t,k} \leq M_{ij}^{12} \cdot y_{ij}^{t,k}, i \in I, j \in J \setminus \{0\}, t \in T, k \in K \quad (30)$$

$$q_{ij}^{t,k} \leq S_{ij}^k - IP_{ij}^{t,k}, i \in I, j \in J \setminus \{0\}, t \in T, k \in K \quad (31)$$

$$M_i^{21} \cdot (y_{i0}^{t,k} - 1) \leq s_{i0}^k - IP_{i0}^{t,k}, i \in I, t = 1 + n \times R, t \geq r, 0 \leq n \leq N - 1, k \in K \quad (32)$$

$$s_{i0}^k - IP_{i0}^{t,k} \leq M_i^{22} \cdot y_{i0}^{t,k}, i \in I, t = 1 + n \times R, t \geq r, 0 \leq n \leq N - 1, k \in K \quad (33)$$

$$q_{i0}^{t,k} \leq S_{i0}^k - IP_{i0}^{t,k}, i \in I, t = 1 + n \times R, t \geq r, 0 \leq n \leq N - 1, k \in K \quad (34)$$

$$S_{ij}^k \geq s_{ij}^k, i \in I, j \in J, k \in K \quad (35)$$

$$s_{ij}^k = \sum_{k' \in K} s_{ij}^{k'} / K, i \in I, j \in J, k \in K \quad (36)$$

$$S_{ij}^k = \sum_{k' \in K} S_{ij}^{k'} / K, i \in I, j \in J, k \in K \quad (37)$$

$$s_{ij}^k \geq 0, i \in I, j \in J, k \in K \quad (38)$$

and constraints (46)-(69) in the Appendix.

In model *SBO2*, the objective function (28) is the same as that of model *SBO*; constraints (29) to (34) are the reformulation of constraints (6) to (11), or equivalently the reformulation of constraints (24), (25), (8), (26), (27), and (11); constraints (35) to (37) are the reformulation of constraints (20); constraints (38) are the reformulation of constraints (21). Since constraints (36) and (37) imply  $s_{ij}^k = s_{ij}$  and  $S_{ij}^k = S_{ij}$  for any  $k$  for some  $s_{ij}$  and  $S_{ij}$ , model *SBO2* is equivalent to model *SBO*.

We first attempted to solve model *SBO2* by Lagrangian relaxation that relaxes constraints (36) and (37) by introducing Lagrange multipliers. However, although such approach can obtain a good lower bound, we cannot construct a high quality feasible solution of the model from the solution of its Lagrangian relaxed problem/model, because for the relaxed model, its optimal values of some variables  $s_{ij}^k$  and  $S_{ij}^k$  may be infinite during the Lagrangian dual optimization process as the coefficients associated with those variables in its objective function may become negative. Of course, we can use the augmented Lagrangian relaxation method (Boyd et al., 2011) to solve the infinite problem, but it requires solving all subproblems optimally in each iteration, which are quite time consuming, especially for large-size instances. Because of this, we finally chose Alternating Direction Method of Multipliers (ADMM, Boyd et al., 2011) to solve the reformulated model. This method, which can avoid the above mentioned infinite problem, is computationally more efficient and can generate more stable and better solutions than the other two methods mentioned. ADMM has been successfully used in distributed convex optimization and machine learning (Boyd et al., 2011).

## 5.1 ADMM Procedure

By introducing Lagrange multipliers  $\lambda_{ij}^k$  and  $\mu_{ij}^k$  and relaxing constraints (36) and (37) respectively, the ADMM relaxed model of *SBO2* can be formulated as *RSBO2* in the following, where the objective function (28) in model *SBO2* is replaced by an augmented Lagrangian function (39) with penalty terms. Here,  $\rho$  is a small positive real number.

Model *RSBO2*:

$$Z^{RSBO2} = \text{Min} \left( \begin{array}{l} \sum_{k \in K} p_k \cdot \left( \sum_{t \in T} \sum_{i \in I} \sum_{j \in J} (h_{ij} \cdot I_{ij}^{t,k} + t_j \cdot v_i \cdot x_{ij}^{t,k}) + \sum_{t \in T} O_0 \cdot y_0^{t,k} + \sum_{t \in T} \sum_{i \in I} \sum_{j \in J \setminus \{0\}} l_{ij} \cdot LS_{ij}^{t,k} \right) \\ + \sum_{k \in K} \sum_{i \in I} \sum_{j \in J} \left( \mu_{ij}^k \cdot \left( s_{ij}^k - \sum_{k' \in K} s_{ij}^{k'} / K \right) + \frac{\rho}{2} \cdot \left( s_{ij}^k - \sum_{k' \in K} s_{ij}^{k'} / K \right)^2 \right) \\ + \sum_{k \in K} \sum_{i \in I} \sum_{j \in J} \left( \lambda_{ij}^k \cdot \left( S_{ij}^k - \sum_{k' \in K} S_{ij}^{k'} / K \right) + \frac{\rho}{2} \cdot \left( S_{ij}^k - \sum_{k' \in K} S_{ij}^{k'} / K \right)^2 \right) \end{array} \right) \quad (39)$$

subject to constraints (2)-(5), (12)-(13), (22)-(23), (29)-(35), (38), and constraints (46)-(69) in the Appendix.

ADMM is an iterative method. In each iteration, the relaxed model is approximately solved by applying a Gauss-Seidel like approach. That is, when ADMM optimizes the decision variables related to each scenario  $k$  in model *RSBO2*, all other decision variables' values are fixed, and all decision variables of the relaxed model are optimized scenario by scenario and sequentially. In this way, the relaxed model *RSBO2* can be decomposed into  $K$  sub-models, denoted by *RSBO2<sub>k</sub>*,  $k \in K$ , where  $\bar{s}_{ij}^{k'}$  and  $\bar{S}_{ij}^{k'}$  are the values of  $(s_{ij}^{k'}, S_{ij}^{k'})$  for all other scenarios obtained in the last iteration of ADMM.

Model *RSBO2<sub>k</sub>*:

$$Z_k^{RSBO2} = \text{Min} \left( \begin{array}{l} \sum_{k \in K} p_k \cdot \left( \sum_{t \in T} \sum_{i \in I} \sum_{j \in J} (h_{ij} \cdot I_{ij}^{t,k} + t_j \cdot v_i \cdot x_{ij}^{t,k}) + \sum_{t \in T} O_0 \cdot y_0^{t,k} + \sum_{t \in T} \sum_{i \in I} \sum_{j \in J \setminus \{0\}} l_{ij} \cdot LS_{ij}^{t,k} \right) \\ + \sum_{k \in K} \sum_{i \in I} \sum_{j \in J} \left( \mu_{ij}^k \cdot \left( s_{ij}^k - \left( s_{ij}^k + \sum_{k' \in K, k' \neq k} \bar{s}_{ij}^{k'} \right) / K \right) + \frac{\rho}{2} \cdot \left( s_{ij}^k - \left( s_{ij}^k + \sum_{k' \in K, k' \neq k} \bar{s}_{ij}^{k'} \right) / K \right)^2 \right) \\ + \sum_{k \in K} \sum_{i \in I} \sum_{j \in J} \left( \lambda_{ij}^k \cdot \left( S_{ij}^k - \left( S_{ij}^k + \sum_{k' \in K, k' \neq k} \bar{S}_{ij}^{k'} \right) / K \right) + \frac{\rho}{2} \cdot \left( S_{ij}^k - \left( S_{ij}^k + \sum_{k' \in K, k' \neq k} \bar{S}_{ij}^{k'} \right) / K \right)^2 \right) \end{array} \right) \quad (40)$$

subject to constraints (2)-(5), (12)-(13), (22)-(23), (29)-(35), (38), and constraints (46)-(69) in the Appendix.

Let us denote the values of the Lagrange multipliers in iteration  $m$  by  $\mu_{ij}^{k,m}$  and  $\lambda_{ij}^{k,m}$ . Given initial values  $\mu_{ij}^{k,0}$  and  $\lambda_{ij}^{k,0}$ , the values of the Lagrange multipliers in each iteration of ADMM are updated according to the following equations (41) and (42).

$$\mu_{ij}^{k,m+1} = \mu_{ij}^{k,m} + \rho \cdot \left( s_{ij}^{k,m} - \sum_{k' \in K} s_{ij}^{k',m} / K \right), i \in I, j \in J, k \in K \quad (41)$$

$$\lambda_{ij}^{k,m+1} = \lambda_{ij}^{k,m} + \rho \cdot \left( S_{ij}^{k,m} - \sum_{k' \in K} S_{ij}^{k',m} / K \right), i \in I, j \in J, k \in K \quad (42)$$

where  $s_{ij}^{k,m}$  and  $S_{ij}^{k,m}$  denote the values of  $(s_{ij}^k, S_{ij}^k)$  obtained by solving sub-model  $RSBO2_k$  in  $m$ -th iteration.

In summary, the iterative procedure of ADMM can be described as the following.

#### ADMM:

**Step 0.** Set  $\mu_{ij}^{k,0} = 0$  and  $\lambda_{ij}^{k,0} = 0$  for all  $i, j$  and  $k$ . Solve all sub-models  $RSBO2_k$  with the objective function (35) ignoring the penalty terms (i.e., the two terms with coefficient  $\rho/2$ ) to obtain the values of  $s_{ij}^{k,0}$  and  $S_{ij}^{k,0}$  for all scenarios. Set  $m = 0$ .

**Step 1.** Update the values of Lagrangian multipliers  $\mu_{ij}^k$  and  $\lambda_{ij}^k$  for all  $i, j$  and  $k$  according to equations (41) and (42), and set  $m + 1 \rightarrow m$ .

**Step 2.** Solve all sub-models one by one with Lagrange multipliers  $\mu_{ij}^{k,m}$  and  $\lambda_{ij}^{k,m}$ . When solving sub-model  $RSBO2_k$ , the values of  $(\bar{s}_{ij}^{k'}, \bar{S}_{ij}^{k'})$  for any  $k' \neq k$  are taken as the latest values of  $s_{ij}^{k'}$  and  $S_{ij}^{k'}$  obtained in the last or current iteration of ADMM.

**Step 3.** If a given stop stopping criterion is achieved, output  $s_{ij}^* = \sum_{k \in K} s_{ij}^{k,m} / K$  and  $S_{ij}^* = \sum_{k \in K} S_{ij}^{k,m} / K$  for all  $i, j$ , calculate an upper bound of the optimal objective value of model  $SBO$  by solving it with  $s_{ij}$  and  $S_{ij}$  replaced by  $s_{ij}^*$  and  $S_{ij}^*$  respectively for all  $i, j$ , and stop. Otherwise, go to Step 1.

The stopping criterion of ADMM can be defined in different ways, such as a given computation time is achieved, a predefined gap between an upper bound and a lower bound is achieved, or a given number of iterations is reached. Note that in each iteration of ADMM, an upper bound of the model  $SBO$  can be obtained by solving it with  $s_{ij}$  and  $S_{ij}$  replaced by  $s_{ij}^* =$

$\sum_{k \in K} s_{ij}^{k,m} / K$  and  $S_{ij}^* = \sum_{k \in K} S_{ij}^{k,m} / K$  respectively for all  $i, j$ , and a lower bound of the model can be obtained by solving the ADMM relaxed model  $RSBO2$  with the quadratic penalty terms in its objective function (39) ignored.

## 5.2 Solving ADMM Submodels by Linear Approximation

Each sub-model  $RSBO2_k$  to be solved in each iteration of ADMM is a mixed-integer quadratic program. Although all the sub-models can be solved by a solver of quadratic programs, it is quite time consuming if the number of stocking locations, the number of products, and/or the number of periods considered are large. For this reason, we propose a linear approximation approach to solve the sub-models more quickly.



$$\text{Let } u_{ij}^k = G_s(s_{ij}^k) = \left( s_{ij}^k - \left( s_{ij}^k + \sum_{k' \in K, k' \neq k} \bar{s}_{ij}^{k'} \right) / K \right)^2$$

$$\text{and } w_{ij}^k = G_S(S_{ij}^k) = \left( S_{ij}^k - \left( S_{ij}^k + \sum_{k' \in K, k' \neq k} \bar{S}_{ij}^{k'} \right) / K \right)^2.$$

The quadratic terms in the objective function of  $RSBO2_k$  can be written as  $\frac{\rho}{2} \sum_{k \in K} \sum_{i \in I} \sum_{j \in J} G_s(s_{ij}^k) + \frac{\rho}{2} \sum_{k \in K} \sum_{i \in I} \sum_{j \in J} G_S(S_{ij}^k)$ . Since all  $G_s(s_{ij}^k)$  and  $G_S(S_{ij}^k)$  are convex functions with a single variable and  $RSBO2_k$  is a minimization problem, these functions can be well approximated from below by a set of linear inequalities.

Let  $G'_s(s_{ij}^k)$  and  $G'_S(S_{ij}^k)$  be the first derivative of function  $G_s(s_{ij}^k)$  and  $G_S(S_{ij}^k)$ , respectively, we have  $G'_s(s_{ij}^k) = 2 \left( \frac{K-1}{K} \right)^2 \cdot s_{ij}^k - 2 \frac{K-1}{K^2} \cdot \sum_{k' \in K, k' \neq k} \bar{s}_{ij}^{k'}$ , and  $G'_S(S_{ij}^k) = 2 \left( \frac{K-1}{K} \right)^2 \cdot S_{ij}^k - 2 \frac{K-1}{K^2} \cdot \sum_{k' \in K, k' \neq k} \bar{S}_{ij}^{k'}$ . Function  $G_s(s_{ij}^k)$  and  $G_S(S_{ij}^k)$  can be approximated from below by the following two sets of linear inequalities (43) and (44), respectively.

$$u_{ij}^k \geq G_s(\bar{s}_{ij}) + G'_s(\bar{s}_{ij})(s_{ij}^k - \bar{s}_{ij}), \quad \bar{s}_{ij} \in \Pi_{ij}^s, \quad i \in I, j \in J \quad (43)$$

$$w_{ij}^k \geq G_S(\bar{S}_{ij}) + G'_S(\bar{S}_{ij})(S_{ij}^k - \bar{S}_{ij}), \quad \bar{S}_{ij} \in \Pi_{ij}^S, \quad i \in I, j \in J \quad (44)$$

where  $\Pi_{ij}^s$  and  $\Pi_{ij}^S$  are a set of integer values of  $s_{ij}^k$  and  $S_{ij}^k$ , respectively.

With this linear approximation, sub-model  $RSBO2_k$  can be reformulated as a mixed integer linear program by transforming its quadratic objective function into the following linear objective function (45) and adding two sets of linear constraints (43) and (44) to the set of original (existing) constraints of  $RSBO2_k$ .

$$Z_k^{RSBO2} = \text{Min} \left( \begin{aligned} & \sum_{k \in K} p_k \cdot \left( \sum_{t \in T} \sum_{i \in I} \sum_{j \in J} (h_{ij} \cdot I_{ij}^{t,k} + t_j \cdot v_i \cdot x_{ij}^{t,k}) + \sum_{t \in T} O_0 \cdot y_0^{t,k} + \sum_{t \in T} \sum_{i \in I} \sum_{j \in J \setminus \{0\}} l_{ij} \cdot LS_{ij}^{t,k} \right) \\ & + \sum_{k \in K} \sum_{i \in I} \sum_{j \in J} \left( \mu_{ij}^k \cdot \left( s_{ij}^k - \left( s_{ij}^k + \sum_{k' \in K, k' \neq k} \bar{s}_{ij}^{k'} \right) / K \right) + \frac{\rho}{2} \cdot u_{ij}^k \right) \\ & + \sum_{k \in K} \sum_{i \in I} \sum_{j \in J} \left( \lambda_{ij}^k \cdot \left( S_{ij}^k - \left( S_{ij}^k + \sum_{k' \in K, k' \neq k} \bar{S}_{ij}^{k'} \right) / K \right) + \frac{\rho}{2} \cdot v_{ij}^k \right) \end{aligned} \right) \quad (45)$$

Let  $s_{ij}^{\min}$  and  $s_{ij}^{\max}$  be the minimum and maximum possible value of  $s_{ij}^k$ , respectively,  $S_{ij}^{\min}$  and  $S_{ij}^{\max}$  be the minimum and maximum possible value of  $S_{ij}^k$ , respectively. There are two ways to generate sets  $\Pi_{ij}^s$  and  $\Pi_{ij}^S$  and their corresponding linear inequalities (43) and (44).

One is to generate the two sets dynamically. That is, we first take an integer value between  $s_{ij}^{\min}$  and  $s_{ij}^{\max}$  for  $\bar{s}_{ij}$  and an integer value between  $S_{ij}^{\min}$  and  $S_{ij}^{\max}$  for  $\bar{S}_{ij}$ , add the corresponding constraints (43) and (44) to model  $RSBO2_k$ , and replace its original quadratic

objective function by linear objective function (45). The obtained MILP model is then solved by a MILP solver. If the optimal solution of the MILP model, denoted by  $(\{\bar{s}_{ij}^*\}, \{\bar{S}_{ij}^*\})$ , coincides with a solution already considered, i.e.,  $\bar{s}_{ij}^* \in \Pi_{ij}^s$  and  $\bar{S}_{ij}^* \in \Pi_{ij}^S$  for all  $i \in I$  and  $j \in \mathcal{V}$ , then  $\Pi_{ij}^s$  and  $\Pi_{ij}^S$  have been found, and this solution is the optimal solution of  $RSBO2_k$ . Otherwise,  $\bar{s}_{ij}^*$  and  $\bar{S}_{ij}^*$  are added to  $\Pi_{ij}^s$  and  $\Pi_{ij}^S$ , respectively, and the above procedure is repeated until an optimal solution of  $RSBO2_k$  is found.

The other way is to generate  $\Pi_{ij}^s$  and  $\Pi_{ij}^S$  once. That is, we set  $\Pi_{ij}^s = \{s_{ij}^{\min}, s_{ij}^{\min}+1, \dots, s_{ij}^{\max}\}$ ,  $\Pi_{ij}^S = \{S_{ij}^{\min}, S_{ij}^{\min}+1, \dots, S_{ij}^{\max}\}$ , replace the quadratic objective function of model  $RSBO2_k$  by linear objective function (45), and add all linear constraints (43) and (44) to this model at the beginning. This MILP model is solved once to obtain the optimal solution of  $RSBO2_k$ . In our numerical experiments in the next section, the second way is adopted.

## 6 NUMERICAL RESULTS

In this section, the performances of model  $SBO$  and its ADMM algorithm are evaluated on four sets of instances generated according to real data from Alibaba, and the  $(s, S)$  policies found by the algorithm are compared with those used in Alibaba for these instances.

The parameters used to generate the instances are set in the following ways: the number of products  $I$  is set to 5 or 10, the number of FDCs  $J$  is set to 9, and the number of periods in the planning horizon is set to 30 periods (days). In addition, the review period of each FDC and that of the CDC are 1 period and 3 periods respectively. The replenishment lead time of each stocking location is taken as an integer in  $[0, 2]$ . This lead time may be 0 because the locations of some suppliers and FDCs are close to the CDC, so that goods can be delivered from those suppliers to the CDC or from the CDC to those FDCs on the same day. The parameter  $r$ , i.e., the starting period of the planning horizon, is an integer randomly taken from  $[0, 3]$ , and the ending period of the planning horizon is taken as  $r+30-1$ .

Since only fast moving goods are involved in the distribution system, we assume that each product's demand observed by each FDC in each period (day) is stationary and normally distributed, with mean and standard deviation generated from historical demand data of Alibaba. For the demand of each product, its mean is in the interval  $[18, 239]$ , and its coefficient of variation is in the interval  $[0.1, 0.4]$ . Each product's volume is ranged from 0.001 to 0.01  $\text{m}^3$ . For the case with  $I = 5$ , the joint replenishment volume of each FDC and that of the CDC are limited to 5  $\text{m}^3$  and 50  $\text{m}^3$ , respectively, whereas for the case with  $I = 10$ , this volume is limited to 10  $\text{m}^3$  and 100  $\text{m}^3$  for each FDC and the CDC respectively. In addition, the unit holding cost of each product in each day at each stocking location is set based on the price of the product, and the unit lost sales cost of each product at each stocking location is set as the unit holding cost  $\times 0.95 / (1-0.95) = 19 \times$  the unit holding cost, where  $0.95 = 95\%$  is the expected service level. For each stocking location, its transportation cost for each replenishment is set based on its volume in  $\text{m}^3$  and its transportation distance in km. For this distribution system, all FDCs and the CDC are located in a region with the distance between any two stocking locations ranged from 80 to 150 km, and the transportation cost per  $\text{m}^3$  per km is estimated as 1 RMB Yuan. Accordingly, the transportation cost for the replenishment of each  $\text{m}^3$  by each stocking location is in the interval  $[80, 150]$ . The major ordering cost for each joint replenishment of the CDC is set between 100 and 2000 RMB Yuan. For the reason of confidentiality, price data of the products are not provided in detail here.

Let  $\mu_{ij}$  denote the mean and  $\sigma_{ij}$  the standard deviation of product  $i$ 's demand observed by stocking location  $j$  ( $j = 0$  for the CDC and  $j = 1, 2, \dots, J$  for the FDCs), where  $\mu_{i0} = \sum_{j=1}^J \mu_{ij}$  and  $\sigma_{i0} = \sqrt{\sum_{j=1}^J \sigma_{ij}^2}$ . The fair share rationing rule introduced in Eppen and Schrage (1981) is applied for both the allocation of on-hand inventory of the CDC among the FDCs and the allocation of transportation capacity among multiple products for each joint replenishment, with the rationing fractions  $f_{ij}$  and  $f'_{ij}$  given as  $f_{ij} = \sigma_{ij} \sqrt{L_{ij} + R_j} / \sum_{j=1}^J \sigma_{ij} \sqrt{L_{ij} + R_j}$  for each product  $i$  and  $f'_{ij} = v_i \sigma_{ij} \sqrt{L_{ij} + R_j} / \sum_{i=1}^I v_i \sigma_{ij} \sqrt{L_{ij} + R_j}$  at each stocking location  $j$ .

The minimum possible value of  $s_{ij}$  and  $S_{ij}$  are both set to zero, i.e.,  $s_{ij}^{\min} = S_{ij}^{\min} = 0$ . The maximum possible value of  $s_{ij}$  and  $S_{ij}$  are set to  $s_{ij}^{\max} = (L_{ij} + R_j) \mu_{ij} + z_{ij} \sigma_{ij} \sqrt{(L_{ij} + R_j)}$  and  $S_{ij}^{\max} = (L_{ij} + R_j) \mu_{ij} + z_{ij} \sigma_{ij} \sqrt{(L_{ij} + R_j)} + C_j^{\max} / v_i$ , respectively, where  $z_{ij}$  is the safety stock factor (z-value) determined by the expected service level 95%, i.e.,  $z_{ij} = 1.65$ . Note that  $s_{ij}^{\max}$  is set according to an upper bound of  $s$  for the optimal  $(s, S)$  policy of a single stocking location found by Veinott (1966),  $S_{ij}^{\max}$  is set to  $s_{ij}^{\max} + C_j^{\max} / v_i$  since the quantity of product  $i$  replenished by stocking location  $j$  never exceeds  $C_j^{\max} / v_i$  due to the corresponding maximum joint replenishment volume constraint. These  $s_{ij}^{\max}$  and  $S_{ij}^{\max}$  are also used in the setting of some big numbers in model *SBO* as explained in subsection 3.3 and in the linear approximation approach used to solve all sub-models *RSBO2<sub>k</sub>* in ADMM.

In the planning horizon, each product's initial inventory position and final inventory position at each stocking location is set to its order-up-to level. For each scenario, each product  $i$ 's demand observed by each FDC  $j$  in each period  $t$  was randomly generated according to its normal distribution by using the Monte Carlo Sampling method. That is, for each scenario  $k$ , to generate  $d_{ij}^{t,k}$ , a real number  $u_{ij}^{t,k}$  was first generated from the uniform distribution on  $[0, 1]$ ,  $d_{ij}^{t,k}$  is then generated as  $(F_{ij}^t)^{-1}(u_{ij}^{t,k})$ , where  $F_{ij}^t(\cdot)$  is the probability distribution function of  $d_{ij}^t$  and  $(F_{ij}^t)^{-1}$  is the inverse function of  $F_{ij}^t$ . With this scenario generation method, all scenarios generated are assumed to have the same probability of occurrence, i.e., the probability of each scenario is  $1/K$ , that is  $p_k = 1/K$  for any  $k = 1, 2, \dots, K$ , where  $K$  is the number of scenarios generated.

The number of scenarios  $K$  is set by applying the statistical analysis method introduced in Shapiro (2000) and Nguyen and Chen (2018). This method calculates the number of scenarios  $K$  required according to a given level of confidence  $\alpha$  and a given length of confidence interval  $H$ , i.e.,  $K = \max((z_{\alpha/2} \cdot \sigma(K_0) / H)^2, K_0)$ , where  $\sigma(K_0) = \sqrt{\sum_{k=1}^{K_0} (\bar{Z}^{SBO} - Z^{SBO}(k))^2} / (K_0 - 1)$ , which can be estimated based on  $K_0$  scenarios,  $Z^{SBO}(k)$  is the distribution system's total cost under scenario  $k$ , and  $\bar{Z}^{SBO}$  is the average of all  $Z^{SBO}(k)$ ,  $z_{\alpha/2}$  is a z-value defined by  $P(x \leq z_{\alpha/2}) = 1 - \alpha/2$ , where  $x \sim N(0, 1)$ . We set  $K_0 = 100$  and tested 10 randomly generated instances with 5

and 10 products, respectively. After conducting the statistical analysis of their results, we observed that our scenario-based MILP model requires 55 and 339 scenarios on average for the instances with 5 products and 35 and 211 scenarios on average for the instances with 10 products to ensure their solution within the deviation 5% and 2% respectively from their true optimal solution in terms of cost at a probability no less than 99.9%. Based on this observation, we set the number of scenarios to 100 and 500 respectively for each instance in our numerical experiments.

Four sets of 10 instances were generated. Each set is indicated by the number of products and the number of scenarios. For example, the set 5x100 contains 10 instances with 5 products and 100 scenarios, and the set 10x500 contains 10 instances with 10 products and 500 scenarios.

In the numerical experiments, all MILP models were solved by calling the solver of CPLEX 12.9 on a personal PC with a CPU of i7-8565U CPU and a RAM of 16GB. The parameter  $\rho$  in ADMM is set to  $10^{-6}$  for all instances tested. Moreover, ADMM is terminated when a predefined relative gap 10% between an upper bound and a lower bound is achieved. The two bounds are obtained by ADMM and the Lagrangian relaxation method described at the end of section 5, respectively. Table 1 to Table 8 show the computational results of the tested instances.

To evaluate the efficiency and effectiveness of our ADMM, we compare it with CPLEX MIP solver for the instances sets 5x100, 10x100 and 5x500. For the instances set 10x500, this comparison is impossible because CPLEX MIP solver run out of memory when it solved these instances. The reason is that for each instance in set 10x500 with 10 products, 10 stocking locations, 30 periods and 500 scenarios, its model *SBO* contains 1500000 integer variables  $y_{ij}^{t,k}$ . However, our ADMM could be used to solve these instances because it requires much less memory due to its decomposition nature. Therefore, ADMM is more adapted to solve large-size instances. For each large-size instance tested, the quality of its solution found by ADMM is evaluated by the relative gap between its lower bound and upper bound obtained by ADMM (see section 5). These results are presented in Table 1 to Table 4.

In these tables, row ‘Cost<sub>cpplx</sub>’ gives the best total cost of model *SBO*, i.e., the smallest objective value (1), found by CPLEX MIP solver with the computation time limited to 2 hours (7200s) for each instance in set 5x100 and 4 hours (14400s) for each instance in set 10x100 and 5x500. Row ‘Cost<sub>ADMM</sub>’ provides the best total cost of model *SBO* found by ADMM for each instance, Cost<sub>ADMM</sub> represents an upper bound of the optimal total cost of model *SBO*. ‘Cost<sub>LB</sub>’ gives the lower bound of the total cost found by the Lagrangian relaxation method mentioned at the end of section 5. Row ‘Reduce<sub>Cost</sub>’ gives the percentage cost reduction of the solution found by ADMM with respect to the solution found by CPLEX, i.e.,  $(\text{Cost}_{\text{cpplx}} - \text{Cost}_{\text{ADMM}}) / \text{Cost}_{\text{cpplx}}$ . ‘Fillrate<sub>cpplx</sub>’ indicates the average fill rate of all products at all FDCs of the solution found by CPLEX, and ‘Fillrate<sub>ADMM</sub>’ gives the average fill rate of all products at all FDCs of the solution found by ADMM. Row ‘Increase<sub>FRI</sub>’ represents the percentage increase of the average fill rate of the solution found by ADMM compared with the solution found by CPLEX, i.e.,  $(\text{Fillrate}_{\text{ADMM}} - \text{Fillrate}_{\text{cpplx}}) / \text{Fillrate}_{\text{cpplx}}$ . ‘Time<sub>ADMM</sub>’ provides the computation time (in seconds) of ADMM. Row ‘Reduce<sub>Time</sub>’ indicates the percentage computation time reduction of ADMM with respect to CPLEX, i.e.,  $(\text{Time}_{\text{cpplx}} - \text{Time}_{\text{ADMM}}) / \text{Time}_{\text{cpplx}}$ . Row ‘Gap<sub>ADMM</sub>’ represents the relative gap between the upper bound Cost<sub>ADMM</sub> and the lower bound Cost<sub>LB</sub>.

Note that Table 4 only provides the results of ADMM since CPLEX run out of memory for all instances in set 10x500. In addition, in our numerical experiments, the fill rate of product  $i$  at

FDC  $j$  indicated by  $fillrate_{ij}$  is calculated as the average of  $fillrate_{ij}^{t,k}$ 's for all periods  $t$  under all scenarios  $k$  considered, where  $fillrate_{ij}^{t,k}$  is defined as the percentage of product  $i$ 's demand (in units) fulfilled (satisfied) by FDC  $j$  in period  $t$  under scenario  $k$ . Because of the space limitation, we only report the average fill rate of all products at all FDCs for each instance, this average fill rate is obtained by averaging  $fillrate_{ij}$  over all products  $i$  and all FDC  $j$ ; individual fill rate for each of the products at each FDC is not reported in the following tables.

**Table 1: Comparison of ADMM and CPLEX on instances 5x100**

Instance	1	2	3	4	5	6	7	8	9	10
Cost <sub>plex</sub>	54148.5	45927.3	72308.4	71311.6	45073.5	61723.4	78200.9	64957.3	49526.9	77990.7
Cost <sub>ADMM</sub>	53123.4	44698.2	69741.9	70140.6	43832.6	59505.2	73924.8	63218.9	48125	75927.4
Reduce <sub>Cost</sub>	1.9%	2.7%	3.5%	1.6%	2.8%	3.6%	5.5%	2.7%	2.8%	2.6%
Fillrate <sub>plex</sub>	88.7%	88.6%	87.4%	85.4%	92.1%	87.6%	80.7%	91.8%	89.9%	86.2%
Fillrate <sub>ADMM</sub>	90.9%	90.3%	89.0%	87.9%	93.3%	88.8%	85.2%	92.6%	90.9%	87.4%
Increase <sub>FR1</sub>	2.4%	1.9%	1.9%	2.9%	1.3%	1.4%	5.6%	0.9%	1.1%	1.5%
Time <sub>ADMM</sub>	415	343	306	1361	350	435	427	365	396	342
Reduce <sub>Time</sub>	94.2%	95.2%	95.8%	81.1%	95.1%	94.0%	94.1%	94.9%	94.5%	95.3%
Cost <sub>LB</sub>	50818.1	43690.6	67825.7	67988.1	41760.4	57779.3	72893.6	61693.2	46525.3	74875
Gap <sub>ADMM</sub>	4.5%	2.3%	2.8%	3.2%	5%	3%	1.4%	2.5%	3.4%	1.4%

**Table 2: Comparison of ADMM and CPLEX on instances 10x100**

Instance	1	2	3	4	5	6	7	8	9	10
Cost <sub>plex</sub>	216714	235497	265637	253580	261575	99569.7	94146.3	96343.9	239737	91116.7
Cost <sub>ADMM</sub>	110263	101235	130895	105792	142650	98101.7	92929.9	94704.2	109825	90126
Reduce <sub>Cost</sub>	49.1%	57.0%	50.7%	58.3%	45.5%	1.5%	1.3%	1.7%	54.2%	1.1%
Fillrate <sub>plex</sub>	47.9%	50.6%	45.5%	50.5%	44.2%	94.9%	95.0%	94.4%	48.4%	92.2%
Fillrate <sub>ADMM</sub>	90.9%	94.2%	91.6%	95.1%	86.8%	96.2%	96.2%	95.8%	93.5%	93.1%
Increase <sub>FR1</sub>	89.9%	86.2%	101.2%	88.1%	96.6%	1.3%	1.3%	1.5%	93.3%	0.9%
Time <sub>ADMM</sub>	1472	1121	1041	1240	1218	955	1043	1386	1463	1391

Reduce <sub>Time</sub>	89.8%	92.2%	92.8%	91.4%	91.5%	93.4%	92.8%	90.4%	89.8%	90.3%
Cost <sub>LB</sub>	107165	97622.6	128878	103825	139959	95433.3	89246	90444.1	106652	85832.6
Gap <sub>ADMM</sub>	2.9%	3.7%	1.6%	1.9%	1.9%	2.8%	4.1%	4.7%	3%	5%

**Table 3: Comparison of ADMM and CPLEX on instances 5x500**

Instance	1	2	3	4	5	6	7	8	9	10
Cost <sub>plex</sub>	142471	128374	108164	113174	135716	120875	138493	137840	126383	138000
Cost <sub>ADMM</sub>	68712.9	49656.2	39304.3	45540.8	72423	48179.9	46437.8	67227.4	59191	63151.8
Reduce <sub>Cost</sub>	51.8%	61.3%	63.7%	59.8%	46.6%	60.1%	66.5%	51.2%	53.2%	54.2%
Fillrate <sub>plex</sub>	29.5%	40.2%	39.0%	36.7%	28.2%	39.9%	43.1%	30.6%	31.6%	30.3%
Fillrate <sub>ADMM</sub>	96.2%	92.8%	96.4%	95.3%	89.5%	95.9%	94.8%	90.3%	90.0%	94.4%
Increase <sub>FR1</sub>	226.3%	130.9%	147%	159.7%	217.3%	140%	120.1%	195%	185.1%	211.6%
Time <sub>ADMM</sub>	2257	3767	1545	3851	4746	5719	2380	3451	4682	2353
Reduce <sub>Time</sub>	84.3%	73.8%	89.3%	73.3%	67%	60.3%	83.5%	76%	67.5%	83.7%
Cost <sub>LB</sub>	65256.1	46492.2	36514.9	41974.6	67488.4	44649.1	43292.5	64465.4	56298.8	57611.4
Gap <sub>ADMM</sub>	5.3%	6.8%	7.6%	8.5%	7.3%	7.9%	7.3%	4.3%	5.1%	9.6%

**Table 4: Computational results of ADMM on instances 10x500**

Instance	31	32	33	34	35	36	37	38	39	40
Cost <sub>LB</sub>	90567.5	100740	125097	100226	112858	97113.2	134968	118209	112987	117087
Cost <sub>ADMM</sub>	95520.9	106416	133027	105902	122107	104413	141952	124308	117320	124081
Gap <sub>ADMM</sub>	5.5%	5.6%	6.3%	5.7%	8.2%	7.5%	5.2%	5.2%	3.8%	6%
Time <sub>ADMM</sub>	11173	15097	18893	23851	13580	23922	16413	17749	16458	14856

From the results of instance sets 5x100, 10x100 and 5x500 in Table 1 to 3, we can observe that ADMM could find a better solution in a much shorter time compared with the computation time of CPLEX MIP solver with relative cost reduction between 1.6% and 5.5% for set 5x100, between 1.1% and 58.3% for set 10x100, and between 46.6% to 66.5% for set 5x500. At the same time, ADMM could find a solution with a higher average customer fill rate than that of the solution found by CPLEX for all instances, with relative fill rate increase more than 1% for 38 of the 40 instances and maximum relative fill rate increase 226.3%. Note that 226.3% is not an absolute fill rate increase for an instance but the relative increase in percentage of the average fill

rate of this instance obtained by ADMM compared with that obtained by CPLEX, because for this instance, the solution found by CPLEX is very poor. In addition, the computation time of ADMM is significantly less than that of CPLEX with relative time reduction between 60.3% and 95.8% for all instances. Moreover, CPLEX could not solve any instance in set 5x500 because of out of memory, but for all instances in this set ADMM could find a solution with the relative gap ranged from 3.8% to 8.2% as shown in Table 4.

To prove the superiority of the  $(s, S)$  policies found by our scenario-based optimization (SBO) approach for the distribution system considered, we compare them with those currently used in Alibaba. For ease of exposition, our approach and the approach used by Alibaba to define  $(s, S)$  policies in its distribution system are referred to as SBO and IPA respectively hereafter.

In IPA, for each product at each stocking location, its inventory policy parameters  $s$  and  $S$  are determined based on the mean  $\mu$  and standard deviation  $\sigma$  of its demand, its replenishment lead time  $L$ , and its review period  $R$ . More precisely, the two parameters are defined as  $s = \mu \cdot (L + R) + z \cdot \sigma \cdot \sqrt{L + R}$ , and  $S = s + Q$ , respectively, where  $z$  is the z-value (safety stock factor) determined by the expected service level of the product at this stocking location and  $Q$  is determined by the mean  $\mu$  and the maximum between the replenishment lead time  $L$  and the review period  $R$ , i.e.,  $\max(L, R)$ . For the confidentiality reason, the formula for calculating  $Q$  is not detailed here.

We compare the two approaches by two criteria: the expected total cost and the average fill rate of the distribution system considered, where the expected total cost is defined as the objective value of model *SBO* and the average fill rate of the system is defined as the average of the fill rates of all products at all FDCs. Since it is very hard to derive analytical formulas for calculating the two criteria for the considered distribution system, we evaluate/estimate the two criteria by simulation of the system with its  $(s, S)$  policies set by either SBO or IPA. To keep consistence, we use the same demand scenarios (realizations of random demands) in the simulation of the system in both cases of SBO and IPA. These demand scenarios are the same as those used in the optimization model *SBO*.

In the following Table 5 to 8, row ‘Fillrates<sub>SBO</sub>’ and ‘Costs<sub>SBO</sub>’ provide respectively the average fill rate of all products at all FDCs of the distribution system and its expected total cost if it is controlled by the  $(s, S)$  policies found by SBO. Row ‘Fillrate<sub>IPA</sub>’ and ‘Cost<sub>IPA</sub>’ provide respectively the average fill rate of all products at all FDCs of the distribution system and its expected total cost if it is controlled by the  $(s, S)$  policies found by IPA. Row ‘Increase<sub>FR2</sub>’ indicates the percentage increase of the average fill rate obtained by SBO compared with that obtained by IPA, i.e.,  $(\text{Fillrates}_{\text{SBO}} - \text{Fillrate}_{\text{IPA}}) / \text{Fillrate}_{\text{IPA}}$ . Row ‘Reduce<sub>Cost</sub>’ indicates the percentage reduction of the expected total cost obtained by SBO with respect to that obtained by IPA, i.e.,  $(\text{Cost}_{\text{IPA}} - \text{Costs}_{\text{SBO}}) / \text{Cost}_{\text{IPA}}$ .

**Table 5: Comparison of SBO and IPA on instances 5x100**

Instance	1	2	3	4	5	6	7	8	9	10
Fillrate <sub>IPA</sub>	84.2%	85.8%	79.3%	80.0%	89.3%	82.7%	66.7%	79.7%	86.5%	73.8%
Fillrates <sub>SBO</sub>	90.9%	90.3%	89.0%	87.9%	93.3%	88.8%	85.2%	92.6%	90.9%	87.4%
Increase <sub>FR2</sub>	7.9%	5.3%	12.3%	9.8%	4.5%	7.4%	27.7%	16.1%	5.1%	18.4%

Cost <sub>IPA</sub>	55887.3	46354.1	76335.9	72839.8	47328.2	64893	81523.7	69909.4	51790.8	81960.1
Cost <sub>SBO</sub>	53123.4	44698.2	69741.9	70140.6	43832.6	59505.2	73924.8	63218.9	48125	75927.4
Reduce <sub>Cost</sub>	4.9%	3.6%	8.6%	3.7%	7.4%	8.3%	9.3%	9.6%	7.1%	7.4%

**Table 6: Comparison of SBO and IPA on instances 10x100**

Instance	11	12	13	14	15	16	17	18	19	20
Fillrate <sub>IPA</sub>	86.6%	90.9%	81.3%	91.3%	73.9%	93.1%	93.3%	92.8%	88.6%	89.4%
Fillrate <sub>SBO</sub>	90.9%	94.2%	91.6%	95.1%	86.8%	96.2%	96.2%	95.8%	93.5%	93.1%
Increase <sub>FR2</sub>	5%	3.6%	12.7%	4.1%	17.6%	3.3%	3%	3.3%	5.5%	4.1%
Cost <sub>IPA</sub>	117286	107270	138506	114752	154252	101247	98783.8	100029	118629	95810.3
Cost <sub>SBO</sub>	110263	101235	130895	105792	142650	98101.7	92929.9	94704.2	109825	90126
Reduce <sub>Cost</sub>	6%	5.6%	5.5%	7.8%	7.5%	3.1%	5.9%	5.3%	7.4%	5.9%

**Table 7: Comparison of SBO and IPA on instances 5x500**

Instance	21	22	23	24	25	26	27	28	29	30
Fillrate <sub>IPA</sub>	79.6%	86.5%	92.0%	89.0%	80.0%	92.0%	91.4%	81.6%	75.1%	80.3%
Fillrate <sub>SBO</sub>	96.2%	92.8%	96.4%	95.3%	89.5%	95.9%	94.8%	90.3%	90.0%	94.4%
Increase <sub>FR2</sub>	20.9%	7.3%	4.8%	7.1%	11.8%	4.2%	3.8%	10.6%	19.9%	17.5%
Cost <sub>IPA</sub>	73510.6	51754.9	40153	47250.5	75942.9	49512.2	48662.8	68932.9	62010.6	64316.6
Cost <sub>SBO</sub>	68712.9	49656.2	39304.3	45540.8	72423	48179.9	46437.8	67227.4	59191	63151.8
Reduce <sub>Cost</sub>	6.5%	4.1%	2.1%	3.6%	4.6%	2.7%	4.6%	2.5%	4.5%	1.8%

**Table 8: Comparison of SBO and IPA on instances 10x500**

Instance	31	32	33	34	35	36	37	38	39	40
Fillrate <sub>IPA</sub>	93.0%	84.8%	83.9%	84.7%	88.5%	91.1%	75.6%	0.822	81.4%	84.8%
Fillrate <sub>SBO</sub>	95.4%	93.5%	94.6%	93.1%	95.5%	95.1%	94.0%	0.921	90.0%	95.4%
Increase <sub>FR2</sub>	2.6%	10.2%	12.7%	9.9%	7.9%	4.4%	24.3%	12.1%	10.6%	12.4%
Cost <sub>IPA</sub>	100098	111053	137970	110116	125884	109469	148445	128599	121266	128619



Cost <sub>SBO</sub>	95520.9	106416	133027	105902	122107	104413	141952	124308	117320	124081
ReduceCost	4.6%	4.2%	3.6%	3.8%	3%	4.6%	4.4%	4.9%	3.3%	3.5%

From Table 5 to 8, we can observe that SBO can find the  $(s, S)$  policies with a higher average customer fill rate than IPA for all instances, with relative fill rate increase ranged from 2.6% to 27.7% and 9.7% on average. At the same time, the inventory polices found by SBO can result in a lower total cost with respect to IPA for all instances, with relative cost reduction ranged from 1.8% to 9.6% and 5.1% on average. That is, the  $(s, S)$  policies found by SBO can achieve a larger fill rate at a smaller total cost with respect to those currently used in Alibaba. Therefore, SBO is more effective than IPA.

It should be noted that all numerical experiments were conducted on a personal PC, and the results in Table 1 to 8 are only used to verify the feasibility of SBO and the effectiveness of ADMM. Since ADMM is a decomposition algorithm, when it is applied to solve real instances in Alibaba, we can use parallel computing techniques and run it on multiple distributed high-performance cloud servers of Alibaba, since all sub-models  $RSBO_k$  in each iteration of ADMM can be solved in parallel. This can dramatically reduce the running time of ADMM. By contrast, CPLEX MIP solver cannot realize this and has the out-of-memory problem for large-size instances.

Furthermore, we analyse in more detail the fill rates of each product at FDCs obtained by ADMM and IPA respectively on instance set 5x100. Table 9 and 10 provide the maximum and minimum fill rate of each product at all FDCs obtained by the two methods respectively. For example, the rows 'P1<sub>Max</sub>' and 'P1<sub>Min</sub>' give the maximal and minimal fill rate of product 1 at all FDCs obtained by ADMM or IPA. The row 'Fillrate<sub>ave</sub>' gives the average fill rate of each product at all FDCs. From the two tables, we can observe that the fill rates of some products at all FDCs are very close, such as product 2 for instance 1, 4, 5, 6 and 9, and product 3 for instance 1, 2, 3, 5 and 8 (see Table 9). However, there are products whose fill rates at different FDCs are different, which may be because of the joint replenishment constraints. Compared with the method IPA currently used in Alibaba, SBO solved by ADMM can achieve a maximal fill rate very close to the best one and a much better minimal fill rate (in 35 of all 50 fill rates) for each product at each FDC. The minimal fill rates of some products obtained by IPA are very small, such as 29.5% for one product in instance 6, 16.31%, 27.66% and 28.1% for three products in instance 7, 15.99% for one product in instance 8, and 16.55% and 29.22% for two products in instance 10. Thus, there is a much larger variation of the fill rates of each product at all FDCs obtained by IPA. In practice, e-commerce companies as Alibaba usually seek for similar service levels for each product at all FDCs to achieve the best customer satisfaction. Therefore, SBO is more effective than IPA in achieving this goal.

**Table 9: Fill rate of each product at FDCs obtained by ADMM on instances 5x100**

Instance	1	2	3	4	5	6	7	8	9	10
P1 <sub>Max</sub>	99.95%	99.94%	99.77%	99.88%	99.01%	96.27%	100%	99.91%	99.93%	97.04%
P1 <sub>Min</sub>	61.71%	70.42%	70.25%	67.15%	59.79%	65.58%	82.55%	77.43%	99.70%	61.89%
P2 <sub>Max</sub>	100%	99.93%	99.91%	100%	99.97%	100%	100%	100%	99.93%	99.57%

P2 <sub>Min</sub>	99.77%	70%	66.03%	99.66%	98.76%	99.66%	66%	69.33%	99.46%	59.60%
P3 <sub>Max</sub>	100%	99.91%	99.95%	99.8%	100%	99.88%	99.99%	99.94%	100%	100%
P3 <sub>Min</sub>	99.75%	99.62%	99.79%	67.93%	99.67%	68.28%	62.55%	99.73%	59.89%	65.12%
P4 <sub>Max</sub>	98.51%	72.81%	100%	93.14%	99.96%	99.69%	98.79%	100%	100%	99.89%
P4 <sub>Min</sub>	62.13%	60.11%	64.86%	62.06%	63.71%	61.68%	68.20%	68.78%	99.62%	76.64%
P5 <sub>Max</sub>	99.93%	99.86%	100%	99.99%	99.96%	100%	99.74%	100%	99.83%	99.89%
P5 <sub>Min</sub>	78.16%	89.68%	99.6%	67.09%	98.28%	98.91%	64.25%	60.71%	64.53%	99.69%
Fillrate <sub>ave</sub>	90.89%	90.31%	89%	87.85%	93.29%	88.83%	85.2%	92.58%	90.89%	87.45%

**Table 10: Fill rates of each product at FDCs obtained by IPA on instances 5x100**

Instance	1	2	3	4	5	6	7	8	9	10
P1 <sub>Max</sub>	100%	100%	99.77%	100%	100%	92.07%	100%	100%	100%	83.99%
P1 <sub>Min</sub>	34.06%	57.32%	36.44%	47.8%	39.36%	34.25%	57.25%	55.90%	99.97%	29.22%
P2 <sub>Max</sub>	100%	100%	100%	100%	100%	100%	100%	100%	100%	87.21%
P2 <sub>Min</sub>	99.998%	58.134%	35.133%	99.993%	99.996%	99.99%	37.48%	46.47%	99.98%	16.55%
P3 <sub>Max</sub>	100%	100%	100%	100%	100%	100%	100%	100%	100%	99.99%
P3 <sub>Min</sub>	97.39%	99.64%	99.99%	53.97%	99.99%	56.84%	27.66%	99.996%	33.74%	33.39%
P4 <sub>Max</sub>	97.23%	60.34%	100%	80.30%	100%	100%	99.21%	100%	100%	100%
P4 <sub>Min</sub>	39.18%	30.94%	37.10%	34.41%	44.14%	29.5%	28.1%	46.13%	99.99%	56.69%
P5 <sub>Max</sub>	100%	100%	100%	100%	100%	100%	99.99%	57.69%	100%	100%
P5 <sub>Min</sub>	60.93%	87.85%	99.9995%	44.48%	99.998%	99.996%	16.31%	15.99%	42.73%	99.99%
Fillrate <sub>ave</sub>	84.24%	85.77%	79.28%	80.04%	89.27%	82.74%	66.70%	79.72%	86.50%	73.84%

To further evaluate the performance of our ADMM, we compare the solution obtained by it with the optimal solution of model *SBO* for 10 instances of a distribution system with one warehouse (CDC), one retailer (FDC), and one product. The ten instances were randomly generated with the parameters set in the same ways as described above. The comparison results are given in Table 11, where the rows  $Cost_{cplex}$ ,  $Cost_{ADMM}$ ,  $Time_{cplex}$ ,  $Time_{ADMM}$ ,  $Fillrate_{cplex}$  and  $Fillrate_{ADMM}$  have the same meanings as the corresponding rows in Table 1-3 except that the optimal solution of model *SBO* for each instance is found by CPLEX.  $Gap_{Cost}$ ,  $Gap_{Time}$  and  $Gap_{FR}$  denote the cost gap, the computation time gap and the fillrate gap between the solution of

ADMM and the optimal solution. From Table 11, we can see that our ADMM can find a solution very close to the optimal solution for most instances. The computation time of ADMM is much shorter than that of CPLEX for most instances.

**Table 11: Comparison of the solution of ADMM and the optimal solution**

Instance	1	2	3	4	5	6	7	8	9	10
Cost <sub>cp</sub>	741.732	1280.22	1751.77	1996.48	902.055	449.7	2308.28	3282.67	803.649	1059.98
Cost <sub>ADMM</sub>	743.693	1286.67	1802.9	2093.92	924.718	495.551	2379.96	3339.95	810.718	1066.72
Gap <sub>Cost</sub>	0.3%	0.5%	2.9%	4.9%	2.5%	10.2%	3.1%	1.7%	0.9%	0.6%
Time <sub>cp</sub>	309	7204	7210	38	136	101	204	43	815	207
Time <sub>ADMM</sub>	43	54	43	33	42	43	52	44	47	45
Gap <sub>Time</sub>	86.1%	99.3%	99.4%	13.2%	69.1%	57.4%	74.5%	-2.3%	94.2%	78.3%
Fillrate <sub>cp</sub>	80.66%	76.48%	70.45%	63.06%	82.28%	98.71%	74.01%	59.89%	97.46%	71.32%
Fillrate <sub>ADMM</sub>	82.86%	78.53%	75.7%	66.18%	86.93%	83.86%	79.58%	66.46%	87.8%	76.4%
Gap <sub>FR</sub>	2.7%	2.7%	7.5%	4.9%	5.6%	-15.0%	7.5%	11.0%	-9.9%	7.1%

In the above numerical experiments, the planning horizon is set to 30 days based on the practice of Alibaba. To evaluate the effect of the planning horizon on the solution of model *SBO*, we compare the total cost of the distribution system obtained with the planning horizon of 30 days (case PH1) and that obtained with the planning horizon of 60 days (case PH2) respectively. More precisely, we consider 60 days of operation of the distribution system. In case PH1, the inventory policy optimization of the system is done twice with the planning horizon of 30 days for each *SBO* model, and the first and the second 30-day *SBO* model are solved independently, under the condition that the final state of the first 30-day model is turned into the initial state of the second 30-day model. The sum of the total costs of the two models are regarded as the total cost of the system in the 60 days of operation in case PH1, which is compared with the total cost of model *SBO* (the system) with the planning horizon of 60 days in case PH2. Table 12 shows the comparison results, from which we can see that the relative gap between the two total costs is no larger than 3.28% for all instances. Thus, it is reasonable to set the planning horizon to 30 days by considering both the effect of the planning horizon and the increase of computational complexity of model *SBO* if we take a longer planning horizon.

**Table 12: Comparison of the results with different planning horizons**

Instance	1	2	3	4	5	6	7	8	9	10
PH1	2423.42	4187.6	6458.03	7794.42	3032.5	1064.453	7985.84	12890.57	1759.431	3621.15
PH2	2505.53	4278.3	6480.1	7889.53	3000.37	1043.67	8178.63	12949.3	1703.52	3702.76
Gap	3.28%	2.12%	0.34%	1.21%	1.07%	1.99%	2.36%	0.45%	3.28%	2.20%

Finally, we evaluate the effect of the end-of-horizon conditions on the solution of model *SBO*. We tested 10 instances of the distribution system with 5 products and 100 demand scenarios and compared its total cost obtained by ADMM with the end-of-horizon conditions for model *SBO* mentioned above (EC1) and its total cost obtained by the same solution method but with no end-of-horizon conditions for the model (EC2: the inventory position of each product at each stocking location at the end of the last period is not fixed). Table 13 presents the comparison results. From this table, we can find that the relative gap between the two total costs is very small for all instances. Therefore, the effect of the end-of-horizon conditions is minor.

**Table 13: Comparison of the results with different end-of-horizon conditions**

Instance	1	2	3	4	5	6	7	8	9	10
EC1	53123.4	44698.2	69741.9	70140.6	43832.6	59505.2	73924.8	63218.9	48125	75927.4
EC2	53129.7	44692.9	69741.6	70136.7	43825.8	59499.6	73932.3	63229.6	48125.5	75927.2
Gap	0.01%	-0.01%	-0.0004%	-0.01%	-0.02%	-0.01%	0.01%	0.02%	0.001%	-0.0003%

## 7 CONCLUSION

In this paper, we consider a two-echelon distribution system in Alibaba with joint inventory replenishment constraints and have studied the optimization of its  $(s, S)$  policies. A novel scenario-based optimization model and an Alternating Direction Method of Multipliers (ADMM) method are proposed to solve this model. ADMM allows solving all subproblems in parallel in each iteration on multiple distributed personal computers or servers. We evaluated the model and the solution method on the instances generated according to real data of Alibaba. Our numerical results show the effectiveness of the model and the method. Compared with the MILP solver of CPLEX 12.9, our ADMM algorithm could find a better solution with a much shorter computation time and could avoid the out-of-memory for large-size instances. The inventory policies found by the algorithm can result in a better average fill rate with a lower expected total cost with respect to those used in Alibaba.

One direction for further study is to extend our proposed model and solution method to more complex distribution systems such as two-echelon distribution systems with non-stationary demands and three-echelon distribution systems. Besides, inventory replenishment planning for Alibaba's supply chain in some special seasons such as the season of double 11 promotion is worthy to be investigated. It is a great challenge for Alibaba to make cost-effective replenishment plans to achieve high customer fill rates in such seasons.

## 8 ACKNOWLEDGEMENT

This work is supported financially by the Alibaba Innovative Research Project 'Optimization of Safety Stock Placement in Supply Chains with Demand and Lead Time Uncertainty' and the Natural Science Foundation of Hunan Province (No. 2022JJ30204).

## 9 REFERENCES

- Andersson, J., Melchior, P. 2001. A two-echelon inventory model with lost sales, *International Journal of Production Economics*, 69(3), pp. 307-315.
- Abouee-Mehrzi, H., Berman, O., Sharma, S. 2015. Optimal joint replenishment and transshipment policies in a multi-period inventory system with lost sales, *Operations Research*, 63, pp. 261-488.
- Al-Rifai, M. H., Rossetti, M. D. 2007. An efficient heuristic optimization algorithm for a two-echelon (R, Q) inventory system, *International Journal of Production Economics*, 109, pp. 195-213.
- Avcı, M.G., Selim, H. 2018. A multi-objective simulation-based optimization approach for inventory replenishment problem with premium freights in convergent supply chains, *Omega*, 80, pp. 153-165.
- Axsäter, S. 1990. Simple solution procedures for a class of two-echelon inventory problems, *Operations Research*, 38, pp. 64-69.
- Axsäter, S., Zhang, W.F. 1999. A joint replenishment policy for multi-echelon inventory control, *International Journal of Production Economics*, 59, pp. 243-250.
- Axsäter, S. 2003a. Approximate optimization of a two-level distribution inventory system, *International Journal of Production Economics*, 81, pp. 545-553.
- Axsäter, S. 2003b. Supply chain operations: serial and distribution systems, *Supply Chain Management: Design, Coordination and Operation*, Amsterdam: Elsevier, pp. 525-556.
- Balkhi, Z. T. 2001. On a finite horizon production lot size inventory model for deteriorating items: An optimal solution, *European Journal of Operational Research*, 132, pp. 210-223.
- Bastos, L. S. L., Mendes, M. L., Nunes, D. R. L., Melo, A. C. S., Carneiro, M. P. 2017. A systematic literature review on the joint replenishment problem solutions: 2006-2015. *Production Journal*, pp. 27.
- Berling, P., Johansson, L., Marklund, J., 2023. Controlling inventories in omni/multi-channel distribution systems with variable customer order-sizes, *Omega*, 114, 102745.
- Berling, P., Marklund, J. 2014. Multi-echelon inventory control: an adjusted normal demand model for implementation in practice, *International Journal of Production Research*, 52, pp. 3331-3347.
- Bijvank, M., Vis, I.F. 2011. Lost-sales inventory theory: A review, *European Journal of Operational Research*, 215, pp. 1-13.
- Bollapragada S., Akella, R., Srinivasan, R. 1998. Centralized ordering and allocation policies in a two-echelon system with non-identical warehouses, *European Journal of Operational Research*, 106, pp.74-81.
- Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J. 2011. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers, *Foundations and Trends in Machine Learning*, 3(1), pp. 1-122.
- Çapar, I. 2013. Joint shipment consolidation and inventory decisions in a two-stage distribution system, *Computers & Industrial Engineering*, 66, pp. 1025-1035.

- Carvajal, J., Castano, F., Sarache, W., Costa, Y. 2020. Heuristic approaches for a two-echelon constrained joint replenishment and delivery problem, *International Journal of Production Economics*, 220, pp. 107420.
- Chen, H. X., Dai, B., Li, Y.A., Zhang, Y.D., Wang, X.Q., Deng, Y.M. 2022, Stock Allocation in a Two-Echelon Distribution System Controlled by  $(s, S)$  Policies, *International Journal of Production Research*, 60, pp. 894-911.
- Clark, A.J., Scarf, H. 1960. Optimal policies for a multi-echelon inventory problem, *Management Science*, 6, pp. 475-490.
- Dai, B., Chen, H.X., Li, Y.A., Zhang, Y.D., Wang, X.Q., Deng, Y.M. 2018. Optimization of  $(s, S)$  inventory policies in a distribution system with minimum and maximum joint replenishment quantity constraints, *48th International Conference on Computers and Industrial Engineering*, Auckland, New Zealand.
- Dai, B., Chen, H.X., Li, Y.A., Zhang, Y.D., Wang, X.Q., Deng, Y.M. 2019. Inventory replenishment planning in a distribution system with safety stock policy and minimum and maximum joint replenishment quantity constraints, *International Conference on Industrial Engineering and Systems Management*, Shanghai, China.
- Dai, B., Chen, H.X., Li, Y.A., Zhang, Y.D., Wang, X.Q., Deng, Y.M. 2021. Inventory Replenishment Planning of a Distribution System with Storage Capacity Constraints and Multi-Channel Order Fulfilment, *Omega*, 102, 102356.
- De Kok, T., Grob, C., Laumanns, M., Minner, S., Rambau, J., Schade, K. 2018. A typology and literature review on stochastic multi-echelon inventory models, *European Journal of Operational Research*, 269, pp. 955-983.
- Diks, E.B., De Kok, A.G. 1998. Optimal control of a divergent multi-echelon inventory system, *European Journal of Operational Research*, 111, pp. 75-97.
- Eppen, G., Schrage, L. 1981. Centralized ordering policies in a multi-warehouse system with lead times and random demand. In: *Multi-level Production/Inventory Control Systems: Theory and Practice*, Ed. by L. B. Schwarz, *TIMS Studies in the Management Sciences*, Vol. 16, North-Holland, Amsterdam, pp. 51-67.
- Fattahi, M., Mahootchi, M., Moattar Husseini, S.M., Keyvanshokoo, E., Alborzi, F. 2015. Investigating replenishment policies for centralised and decentralised supply chains using stochastic programming approach, *International Journal of Production Research*, 53, pp 41-69.
- Federgruen, A., Zipkin, P. 1984. Allocation policies and cost approximations for multilocation inventory systems, *Naval Research Logistics Quarterly*, 31, pp. 97-129
- Goyal, S. K., Satir, A. T. 1989. Joint replenishment inventory control: Deterministic and stochastic models, *European Journal of Operational Research*, 38, pp. 2-13.
- Hill, R. M., Seifbarghy, M., Smith, D. K. 2007. A two-echelon inventory model with lost sales, *European Journal of Operational Research*, 181(2), pp. 753-766.
- Jalali, H., Nieuwenhuysse, I.V. 2015. Simulation optimization in inventory replenishment: a classification, *IIE Transactions*, 47, pp. 1217-1235.
- Jula, P., Leachman, R.C. 2011. A supply-chain optimization model of the allocation of containerized imports from Asia to the United States, *Transportation Research Part E: Logistics and Transportation Review*, 47, pp. 609-622.
- Khouja, M., Goyal, S. 2008. A review of the joint replenishment problem literature: 1989-2005. *European Journal of Operational Research*, 186, pp. 1-16.

- Kukreja, A., Schmidt, C.P. 2005. A model for lumpy demand parts in a multi-location inventory system with transshipments, *Computers & Operations Research*, 32, pp. 2059-2075.
- Lagodimos, A.G., Koukoumialos, S. 2008. Service performance of two-echelon supply chains under linear rationing, *International Journal of Production Economics*, 112, pp. 869-884.
- Lin, Y. H., Leung J.M.Y., Zhang, L.M., Gu, J.W. 2020. Single-item repairable inventory system with stochastic new and warranty demands, *Transportation Research Part E: Logistics and Transportation Review*, 142, 102035.
- Nguyen, D.H., Chen, H.X. 2018. Supplier selection and operation planning in biomass supply chains with supply uncertainty, *Computers and Chemical Engineering*, 118, pp. 103-117.
- Noordhoek, M., Dullaert, W., Lai, D.S.W., Leeuw, S.D. 2018. A simulation-optimization approach for a service-constrained multi-echelon distribution network, *Transportation Research Part E: Logistics and Transportation Review*, 114, pp. 292-311.
- Qiu, R., Sun, Y., Sun, M., 2022. A robust optimization approach for multi-product inventory management in a dual-channel warehouse under demand uncertainties, *Omega*, 109, 102591.
- Seifbarghy, M., Jokar, M. R. A. 2006. Cost evaluation of a two-echelon inventory system with lost sales and approximately Poisson demand, *International journal of production economics*, 102(2), pp. 244-254.
- Shapiro, A. 2000. Stochastic programming by monte carlo simulation methods, doi: <https://doi.org/10.18452/8226>.
- Teunter, R. H., Kuipers, S., 2022. Inventory control with demand substitution: new insights from a two-product Economic Order Quantity analysis, *Omega*, 113, 102712.
- Tsai, S.C., Chen, S.T. 2017. A simulation-based multi-objective optimization framework: A case study on inventory management, *Omega*, 70, pp. 148-159.
- Van der Heijden, M.C., Diks, E.B., De Kok, A.G. 1997. Stock allocation in general multi-echelon distribution systems with (R, S) order-up-to-policies, *International Journal of Production Economics*, 49, pp. 157-174.
- Veinott, A. 1966. On the optimality of  $(s, S)$  inventory policies: new conditions and a new proof, *SIAM J. Appl. Math.*, 14, pp. 1067-1083.
- Viswanathan, S. 1997. Periodic Review  $(s, S)$  Policies for Joint Replenishment Inventory Systems, *Management Science*, 43 (10), pp. 1447-1454.
- Wang, Y., Geunes, J., Nie, X., 2022. Optimising inventory placement in a two-echelon distribution system with fulfillment-time-dependent demand, *International Journal of Production Research*, 60(1), pp. 48-72.
- Zhou, W.Q., Chen, L., Ge, H.M. 2012. A multi-product multi-echelon inventory control model with joint replenishment strategy, *Applied Mathematical Modelling*, 37, pp. 2039-2050.
- Zhou, Y. W., Lau, H. S., Yang, S. L. 2004. A finite horizon lot-sizing problem with time-varying deterministic demand and waiting-time-dependent partial backlogging, *International Journal of Production Economics*, 91, pp. 109-119.

## 10 APPENDIX

### I Linearization of the Model

In the following, we transform logical constraints (14) to (19) in model *SBO* into linear constraints so that it becomes a mixed integer linear program.

Logical constraints (14) and (15) can be transformed into linear constraints in the following way. Firstly, we introduce binary variables  $z_{i_0}^{t,k}$  to indicate whether  $\sum_{j=1}^J q_{ij}^{t,k} > I_{i_0}^{t,k}$ , such that  $z_{i_0}^{t,k} = 1$  if  $\sum_{j=1}^J q_{ij}^{t,k} > I_{i_0}^{t,k}$  and  $z_{i_0}^{t,k} = 0$  if  $\sum_{j=1}^J q_{ij}^{t,k} \leq I_{i_0}^{t,k}$ . This can be done by adding the following constraints, where  $M_i^{31}$  and  $M_i^{32}$  are big numbers.

$$\sum_{j=1}^J q_{ij}^{t,k} - I_{i_0}^{t,k} \leq M_i^{31} \cdot z_{i_0}^{t,k}, i \in I, t \in T, k \in K \quad (46)$$

$$M_i^{32} \cdot (z_{i_0}^{t,k} - 1) \leq \sum_{j=1}^J q_{ij}^{t,k} - I_{i_0}^{t,k}, i \in I, t \in T, k \in K \quad (47)$$

Two cases may happen for the logical constraints. Firstly, in case without inventory rationing, i.e.,  $\sum_{j=1}^J q_{ij}^{t,k} \leq I_{i_0}^{t,k}$ , we have  $q_{ij}^{u,k} = q_{ij}^{t,k}$ . This can be ensured by adding the following constraints, where  $M_i^{33}$  are big numbers.

$$q_{ij}^{u,k} \leq q_{ij}^{t,k}, i \in I, j \in J \setminus \{0\}, t \in T, k \in K \quad (48)$$

$$q_{ij}^{u,k} \geq q_{ij}^{t,k} - M_i^{33} \cdot z_{i_0}^{t,k}, i \in I, j \in J \setminus \{0\}, t \in T, k \in K \quad (49)$$

Secondly, in case with inventory rationing, i.e.,  $\sum_{j=1}^J q_{ij}^{t,k} > I_{i_0}^{t,k}$ , we have  $q_{ij}^{u,k} = q_{ij}^{t,k} - f_{ij} \cdot \max\left(\sum_{j=1}^J q_{ij}^{t,k} - I_{i_0}^{t,k}, 0\right)$ . By introducing auxiliary variables  $\xi_i^{t,k}$  to represent

$\max\left(\sum_{j=1}^J q_{ij}^{t,k} - I_{i_0}^{t,k}, 0\right)$ , we have the following constraints, where  $M_i^{34}$  are big numbers.

$$q_{ij}^{u,k} = q_{ij}^{t,k} - f_{ij} \cdot \xi_i^{t,k}, i \in I, j \in J \setminus \{0\}, t \in T, k \in K \quad (50)$$

$$\xi_i^{t,k} \geq \sum_{j=1}^J q_{ij}^{t,k} - I_{i_0}^{t,k}, i \in I, t \in T, k \in K \quad (51)$$

$$\xi_i^{t,k} \leq \sum_{j=1}^J q_{ij}^{t,k} - I_{i_0}^{t,k} + M_i^{34} \cdot (1 - z_{i_0}^{t,k}), i \in I, t \in T, k \in K \quad (52)$$

$$\xi_i^{t,k} \geq 0, i \in I, t \in T, k \in K \quad (53)$$



Note that: 1) If  $z_{i0}^{t,k} = 1$ , from (47), we have  $\sum_{j=1}^J q_{ij}^{t,k} - I_{i0}^{t,k} \geq 0$ ; from (51) and (52), we have  $\xi_i^{t,k} = \sum_{j=1}^J q_{ij}^{t,k} - I_{i0}^{t,k}$ , so  $\xi_i^{t,k} = \max\left(\sum_{j=1}^J q_{ij}^{t,k} - I_{i0}^{t,k}, 0\right)$ . 2) If  $z_{i0}^{t,k} = 0$ , from (46), we have  $\sum_{j=1}^J q_{ij}^{t,k} - I_{i0}^{t,k} \leq 0$ ; from (48) and (49), we have  $q_{ij}^{u,k} = q_{ij}^{t,k}$ , so we have  $\xi_i^{t,k} = 0$  from (50), therefore  $\xi_i^{t,k} = \max\left(\sum_{j=1}^J q_{ij}^{t,k} - I_{i0}^{t,k}, 0\right)$ . This implies  $\xi_i^{t,k} = \max\left(\sum_{j=1}^J q_{ij}^{t,k} - I_{i0}^{t,k}, 0\right)$  holds in both cases.

Similarly, logical constraints (16)-(19) can be transformed into linear constraints. Firstly, we introduce binary variables  $z_j^{u,k}$  to indicate whether  $\sum_{i=1}^I v_i \cdot q_{ij}^{u,k} > C_j^{\max}$ ,  $j \in \mathcal{J}$ , such that  $z_j^{u,k} = 1$  if  $\sum_{i=1}^I v_i \cdot q_{ij}^{u,k} > C_j^{\max}$ , and  $z_j^{u,k} = 0$  if  $\sum_{i=1}^I v_i \cdot q_{ij}^{u,k} \leq C_j^{\max}$ . This can be done by adding the following constraints, where  $M_j^{41}$ ,  $M_j^{42}$ ,  $M_0^{41}$ , and  $M_0^{42}$  are big numbers.

$$\sum_{i=1}^I v_i \cdot q_{ij}^{u,k} - C_j^{\max} \leq M_j^{41} \cdot z_j^{u,k}, j \in J \setminus \{0\}, t \in T, k \in K \quad (54)$$

$$M_j^{42} \cdot (z_j^{u,k} - 1) \leq \sum_{i=1}^I v_i \cdot q_{ij}^{u,k} - C_j^{\max}, j \in J \setminus \{0\}, t \in T, k \in K \quad (55)$$

$$\sum_{i=1}^I v_i \cdot q_{i0}^{t,k} - C_0^{\max} \leq M_0^{41} \cdot z_0^{t,k}, t \in T, k \in K \quad (56)$$

$$M_0^{42} \cdot (z_0^{t,k} - 1) \leq \sum_{i=1}^I v_i \cdot q_{i0}^{t,k} - C_0^{\max}, t \in T, k \in K \quad (57)$$

Two cases may happen for the logical constraints. Firstly, in case without transportation capacity rationing, i.e.,  $\sum_{i=1}^I v_i \cdot q_{ij}^{u,k} \leq C_j^{\max}$ , we have  $q_{ij}^{u,k} = q_{ij}^{t,k}$ . This can be ensured by adding the following constraints, where  $M_j^{43}$  and  $M_0^{43}$  are big numbers.

$$x_{ij}^{t,k} \leq q_{ij}^{u,k}, i \in I, j \in J \setminus \{0\}, t \in T, k \in K \quad (58)$$

$$x_{ij}^{t,k} \geq q_{ij}^{u,k} - M_j^{43} \cdot z_j^{u,k}, i \in I, j \in J \setminus \{0\}, t \in T, k \in K \quad (59)$$

$$x_{i0}^{t,k} \leq q_{i0}^{t,k}, i \in I, t \in T, k \in K \quad (60)$$

$$x_{i0}^{t,k} \geq q_{i0}^{t,k} - M_0^{43} \cdot z_0^{t,k}, i \in I, t \in T, k \in K \quad (61)$$

Secondly, in case with transportation capacity rationing, i.e., if  $\sum_{i=1}^I v_i \cdot q_{ij}^{t,k} > C_j^{\max}$ , we have  $v_i \cdot x_{ij}^{t,k} = v_i \cdot q_{ij}^{t,k} - f'_{ij} \left( \sum_{i=1}^I v_i \cdot q_{ij}^{t,k} - C_j^{\max} \right)$ . By introducing auxiliary variables  $\eta_j^{t,k}$  to represent  $\max \left( \sum_{i=1}^I v_i \cdot q_{ij}^{t,k} - C_j^{\max}, 0 \right)$ , we have the following constraints, where  $M_j^{44}$  and  $M_0^{44}$  are big numbers.

$$x_{ij}^{t,k} = q_{ij}^{t,k} - \frac{f'_{ij}}{v_i} \eta_j^{t,k}, i \in I, j \in J \setminus \{0\}, t \in T, k \in K \quad (62)$$

$$\eta_j^{t,k} \geq \sum_{i=1}^I v_i \cdot q_{ij}^{t,k} - C_j^{\max}, j \in J \setminus \{0\}, t \in T, k \in K \quad (63)$$

$$\eta_j^{t,k} \leq \sum_{i=1}^I v_i \cdot q_{ij}^{t,k} - C_j^{\max} + M_j^{44} \cdot (1 - z_j^{t,k}), j \in J \setminus \{0\}, t \in T, k \in K \quad (64)$$

$$\eta_j^{t,k} \geq 0, j \in J \setminus \{0\}, t \in T, k \in K \quad (65)$$

$$x_{i0}^{t,k} = q_{i0}^{t,k} - \frac{f'_{i0}}{v_i} \eta_0^{t,k}, i \in I, t \in T, k \in K \quad (66)$$

$$\eta_0^{t,k} \geq \sum_{i=1}^I v_i \cdot q_{i0}^{t,k} - C_0^{\max}, t \in T, k \in K \quad (67)$$

$$\eta_0^{t,k} \leq \sum_{i=1}^I v_i \cdot q_{i0}^{t,k} - C_0^{\max} + M_0^{44} \cdot (1 - z_0^{t,k}), t \in T, k \in K \quad (68)$$

$$\eta_0^{t,k} \geq 0, t \in T, k \in K \quad (69)$$

Note that the validation of  $\eta_j^{t,k} = \max \left( \sum_{i=1}^I v_i \cdot q_{ij}^{t,k} - C_j^{\max}, 0 \right)$  can be proved in a similar way as that of  $\xi_i^{t,k} = \max \left( \sum_{j=1}^J q_{ij}^{t,k} - I_{i0}^{t,k}, 0 \right)$ .

## II Parameter setting of the big numbers

In the linear constraints (24) to (27) and (46) to (69) transformed from the logical constraints of model *SBO*, all big numbers must be defined. Let  $s_{ij}^{\max}$  and  $S_{ij}^{\max}$  be the maximum possible value of  $s_{ij}$  and  $S_{ij}$ , respectively, where  $j = 0$  corresponds to the CDC and  $j > 0$  corresponds to FDC  $j$ . From constraints (24) and (25),  $M_{ij}^{11}$  and  $M_{ij}^{12}$  must be set to satisfy  $-M_{ij}^{11} \leq s_{ij} - IP_{ij}^{t,k}$  and  $s_{ij} - IP_{ij}^{t,k} \leq M_{ij}^{12}$ , i.e.,  $M_{ij}^{11} \geq IP_{ij}^{t,k} - s_{ij}$  and  $M_{ij}^{12} \geq s_{ij} - IP_{ij}^{t,k}$  for any possible values of  $s_{ij}$  and  $IP_{ij}^{t,k}$ . Since  $0 \leq IP_{ij}^{t,k} \leq S_{ij}$  for inventory model *SBO* with lost sales,

we have  $IP_{ij}^{t,k} - s_{ij} \leq S_{ij} - s_{ij}$  and  $s_{ij} - IP_{ij}^{t,k} \leq s_{ij}$ . Moreover, since  $S_{ij} - s_{ij} \leq S_{ij}^{\max}$ , we can set  $M_{ij}^{11} = S_{ij}^{\max}$  and  $M_{ij}^{12} = s_{ij}^{\max}$ . Similarly, from constraints (26) and (27),  $M_i^{21}$  and  $M_i^{22}$  must be set to satisfy  $-M_i^{21} \leq s_{i0} - IP_{i0}^{t,k}$  and  $s_{i0} - IP_{i0}^{t,k} \leq M_i^{22}$  for any possible values of  $s_{i0}$  and  $IP_{i0}^{t,k}$ . Then, by analogy with the setting of  $M_{ij}^{11}$  and  $M_{ij}^{12}$ , we can set  $M_i^{21} = S_{i0}^{\max}$  and  $M_i^{22} = s_{i0}^{\max}$ .

From constraints (46) to (49), big number  $M_i^{31}$ ,  $M_i^{32}$ ,  $M_i^{33}$ , and  $M_i^{34}$  must be set to satisfy  $M_i^{31} \geq \sum_{j=1}^J q_{ij}^{t,k} - I_{i0}^{t,k}$ ,  $M_i^{32} \geq I_{i0}^{t,k} - \sum_{j=1}^J q_{ij}^{t,k}$ ,  $M_i^{33} \geq q_{ij}^{t,k} - q_{ij}^{u,k}$ , and  $M_i^{34} \geq I_{i0}^{t,k} - \sum_{j=1}^J q_{ij}^{t,k}$  for any possible values of  $q_{ij}^{t,k}$ ,  $I_{i0}^{t,k}$ , and  $q_{ij}^{u,k}$ , so  $M_i^{31}$  can be set to  $\max \sum_{j=1}^J q_{ij}^{t,k}$ ,  $M_i^{32}$  can be set to  $\max I_{i0}^{t,k}$ ,  $M_i^{33}$  can be set to  $\max q_{ij}^{t,k}$ , and  $M_i^{34}$  can be set to  $\max I_{i0}^{t,k}$ . Since  $q_{ij}^{t,k} \leq S_{ij}^{\max}$  and  $I_{i0}^{t,k} \leq S_{i0}^{\max}$ , we can set  $M_i^{31} = \sum_{j=1}^J S_{ij}^{\max}$ ,  $M_i^{32} = S_{i0}^{\max}$ ,  $M_i^{33} = \max_j S_{ij}^{\max}$ , and  $M_i^{34} = S_{i0}^{\max}$ .

From constraints (54), (55), (59) and (64),  $M_j^{41}$ ,  $M_j^{42}$ ,  $M_j^{43}$ , and  $M_j^{44}$  must be set to satisfy  $M_j^{41} \geq \sum_{i=1}^I v_i \cdot q_{ij}^{u,k} - C_j^{\max}$ ,  $M_j^{42} \geq C_j^{\max} - \sum_{i=1}^I v_i \cdot q_{ij}^{t,k}$ ,  $M_j^{43} \geq q_{ij}^{u,k} - x_{ij}^{t,k}$ , and  $M_j^{44} \geq C_j^{\max} - \sum_{i=1}^I v_i \cdot q_{ij}^{t,k}$  for any possible values of  $q_{ij}^{u,k}$ ,  $q_{ij}^{t,k}$ , and  $x_{ij}^{t,k}$ . Since  $\sum_{i=1}^I v_i \cdot q_{ij}^{u,k} \leq \sum_{i=1}^I v_i \cdot S_{ij}^{\max}$  and  $q_{ij}^{u,k} \leq S_{ij}^{\max}$ , we can set  $M_j^{41} = \sum_{i=1}^I v_i \cdot S_{ij}^{\max}$ ,  $M_j^{42} = C_j^{\max}$ ,  $M_j^{43} = \max_i S_{ij}^{\max}$ , and  $M_j^{44} = C_j^{\max}$ . Similarly, from constraints (56), (57), (61) and (68),  $M_0^{41}$ ,  $M_0^{42}$ ,  $M_0^{43}$ , and  $M_0^{44}$  must be set to satisfy  $M_0^{41} \geq \sum_{i=1}^I v_i \cdot q_{i0}^{t,k} - C_0^{\max}$ ,  $M_0^{42} \geq C_0^{\max} - \sum_{i=1}^I v_i \cdot q_{i0}^{u,k}$ ,  $M_0^{43} \geq q_{i0}^{t,k} - x_{i0}^{t,k}$ , and  $M_0^{44} \geq C_0^{\max} - \sum_{i=1}^I v_i \cdot q_{i0}^{u,k}$ . Consequently, we can set  $M_0^{41} = \sum_{i=1}^I v_i \cdot S_{i0}^{\max}$ ,  $M_0^{42} = C_0^{\max}$ ,  $M_0^{43} = \max_i S_{i0}^{\max}$ , and  $M_0^{44} = C_0^{\max}$ .