Integrated Multi-Echelon Supply Chain Design with Inventories under Uncertainty: MINLP Models, Computational Strategies

Fengqi You, Ignacio E. Grossmann*

Department of Chemical Engineering, Carnegie Mellon University
Pittsburgh, PA 15213, USA

November, 2008

Abstract

We address the optimal design of a multi-echelon supply chain and the associated inventory systems in the presence of uncertain customer demands. By using the guaranteed service approach to model the multi-echelon stochastic inventory system, we develop an optimization model for simultaneously optimizing the transportation, inventory and network structure of a multi-echelon supply chain. We formulate this problem as an MINLP with a nonconvex objective function including bilinear, trilinear and square root terms. By exploiting the properties of the basic model, we reformulate the problem as a separable concave minimization program. A spatial decomposition algorithm based on Lagrangean relaxation and piecewise linear approximation is proposed to obtain near global optimal solutions with reasonable computational expense. Examplers for industrial gas supply chains with up to 5 plants, 100 potential distribution centers and 200 customers are presented.

1. Introduction

Due to the increasing pressure for remaining competitive in the global market place, optimizing inventories across the supply chain has become a major challenge for the process industries to reduce costs and to improve the customer service.[1, 2] This challenge requires integrating inventory management with supply chain network design, so that decisions on the locations to stock the inventory and the associated amount of inventory in each stocking location can be determined simultaneously for lower costs and higher customer service level. However, the integration is usually nontrivial for multi-echelon supply chains and their

* To whom all correspondence should be addressed. E-mail: grossmann@cmu.edu

associated inventory systems in the presence of uncertain customer demands.[3]

The multi-echelon inventory management problem has been extensively studied in the past several decades. However, most of the inventory literature only considers the optimization of inventory decisions without integrating them with supply chain design and planning decisions. Simpson[4] first studied the control of multi-echelon inventory system for a serial supply chain. In that paper, Simpson proposes the guaranteed service approach to describe the mechanics of the serial inventory system, in which each stage operates a base-stock policy in the face of random but bounded demand. Simpson's results showed that the optimal inventory policy for the serial system is an "all or nothing" policy, i.e. each stage either has no safety stock, or carries enough stocks to decouple the downstream stages form the upstream stages. Almost at the same time, Clark and Scarf[5] also studied the optimal control policy of the multi-echelon inventory system with each stage operating with a base-stock policy. In contrast to Simpson's work, Clark and Scarf consider that the uncertain demand at each stage to be unbounded, and thus the service level and replenishment lead time of each stage depend on the upstream adjacent supply stages' stock level. To solve the problem, they proposed a dynamic programming approach based on the calculation of recursions for stage inventory and replenishment amount. Later, Federgruen and Zipkin[6] extended the result of Clark and Scarf[5] to infinite horizon, and showed that a stationary order-up-to level echelon policy is optimal. Some recent extensions of the work by Clark and Scarf[5] include the work by Lee and Billington,[7] which presents a supply chain operation model with multi-echelon inventory system operating under a periodic-review base-stock system at Hewlett Packard, the work by Glasserman and Tayur,[8] which considers capacitated limits in the multi-echelon inventory model, and the work by Ettl et al.,[9] which analyzed the relationship between stochastic lead time and the inventory level for a multi-echelon system.

Meanwhile, the guaranteed service approach, which is based on the idea of maximum service time allowance in each inventory location, has drawn more attention due to its ease for computation for large scale inventory systems. Graves[10] proposed that the problem by Simpson[4] could be solved as a dynamic program. Later, different extensions of Simpson's work for assembly networks, distribution networks and spanning trees were given by Inderfurth,[11, 12] Inderfurth and Minner,[13] Graves and Willems,[14] all of whom use dynamic programming to solve the inventory optimization problem. Recently, Minner[15] used a similar approach to investigate the safety stock placement problem for reverse supply chains. Graves and Willems[16, 17] extended the guaranteed service model for general supply chain configuration problem with safety stock optimization. Further extensions of the guaranteed service approach are given by Humair and Willems[18] that accounts for supply chain network with clusters of commonality, and Bossert and Willems[19] that allows the multi-echelon inventory system to be reviewed and replenished for an arbitrary review period at each stage. Inventory placement problem for general acyclic supply chain network has recently been addressed by Magnanti et al.,[20] who present an efficient mathematical programming approach to solve the guaranteed service multi-echelon inventory model instead of using the traditional dynamic programming approach. Jung et al.[21] considered the safety stock optimization for a chemical supply chain by using a simulation-optimization framework, but the supply chain design decisions are not jointly optimized.

Research on integrated supply chain network design and inventory management is relatively new. Most of the literature in this area tends to approximate the inventory cost coarsely without detailed inventory management policy,[21-25] and most of it does not optimize the safety stocks, but just considers the safety stock level as a given parameter that is treated as the lower bound of the total inventory level,[22] or the safety stock level is considered as the inventory targets that would lead to some penalty costs if violated.[23] There are, however, several related works that offer relevant insights. Erlebacher and Meller[24] developed a nonlinear integer programming model for the joint location-inventory model. To solve the highly nonlinear nonconvex problem efficiently, they use a continuous approximation and a number of other heuristic methods. Daskin et al.[25] and Shen et al.[26] present a joint location-inventory model, which extends the classical uncapacitated facility location model to include nonlinear working inventory and safety stock costs for a two-stage network, so that decisions on the installation of distribution centers (DCs) and the detailed inventory replenishment decisions are jointly optimized. To simplify the problem, inventories in the retailers are neglected, and they also assume that all the DCs have the same constant replenishment lead time, and the demand at each customer has the same variance-to-mean ratio. With the same assumptions, Ozsen et al.[27] have extended the model to consider capacitated limits in the DCs. Their work is further extended by Ozsen et al.[28] to compare with the cases where customers restrict to single sourcing and the case which customers allow multi-sourcing. Another extension is given by Sourirajan et al.,[29] in which the assumption on identical replenishment lead time is relaxed while the assumption on demand uncertainty is still enforced. Recently, You and Grossmann[30] proposed a mixed-integer nonlinear programming (MINLP) approach to study a more general model based on the one developed by Daskin et al.[25] and Shen et al.,[26] relaxing the assumption on the identical variance-to-mean ratio for customer demands.

The objective of this work is to develop optimization models and solution algorithms to address the problem of joint multi-echelon supply chain network design and inventory management. By using the guaranteed service approach to model the multi-echelon inventory system,[4, 11-19] we capture the stochastic nature of the problem and develop an equivalent deterministic optimization model. The model determines the supply chain design decisions such as the locations of distribution centers (DCs), assignments of customer demand zones to DCs, assignments of DCs to plants, shipment levels from plants to the DCs and from DCs to customers, and inventory decisions such as pipeline inventory and safety stock in each node of the supply chain network. The model also captures risk-pooling effects[31] by consolidating the safety stock inventory of downstream nodes to the upstream nodes in the multi-echelon supply chain. The model is first formulated as a mixed-integer nonlinear program (MINLP) with a nonconvex objective function, and then reformulated as a separable concave minimization program after exploiting the properties of the basic model. To solve the problem efficiently, a tailored hierarchical decomposition algorithm based on Lagrangean relaxation and piece-wise linear approximation is developed to obtain near global optimal solutions within 1% optimality gap with modest CPU times. Several computational examples for industrial gases supply chains and performance chemical supply chains are presented to illustrate the application of the model and the performance of the proposed algorithm.

The outline of this paper is as follows. Some basic concepts of inventory management with risk pooling and the guaranteed service model for multi-echelon inventory system are presented in Section 2. The problem statement is given in Section 3. In Section 4, we introduce the joint multi-echelon supply chain design and inventory management model. Two small illustrative examples are given in Section 5. To solve the large scale problem, an efficient decomposition algorithm based on Lagrangean relaxation, piecewise linear approximation and model property is presented in Section 6. In Section 7 we present our computational results and analysis. We conclude the paper with some conclusions and future directions in Section 8.

2. Multi-Echelon Inventory Model

In this section, we briefly review some inventory management models that are related to the problem addressed in this work. Detailed discussion about single stage and multi-echelon inventory management models are given by Zipkin[3] and Graves and Willem,[16] respectively.

2.1. Single Stage Inventory Model under Base-Stock Policy

There are many control policies for single stage inventory systems, such as base stock policy, (s,S) policy, (r, Q) policy, etc.[3] Among these policies, the periodic review base stock policy is most widely used in the inventory control practice. The reason is based on two facts. As shown by Federgruen and Zipkin,[6] the base stock policy is optimal for single stage inventory system facing stationary demand. For multi-echelon inventory system, the base stock policy, though not necessarily optimal, has the advantage of being simple to implement and close to the optimum.[32] Before introducing the multi-echelon inventory model, we first review the single stage base stock policy, which is the common building block for most of the multi-echelon inventory models.

Figure 1 shows the inventory profile for a given product in a stocking facility operated under the periodic review base stock policy. As can be seen, the inventory level decreases due to the customer demand and increases when replenishments arrive. Under the periodic review base stock policy, inventory is reviewed at the beginning of each review period and the amount of the difference between a specified base stock level and the actual inventory position (on-hand inventory plus in-process inventory minus backorders) is ordered for replenishment. It is interesting to note that the well-known continuous review (r, Q) policy can be treated as a special case of base stock policy with base stock level equal to $r+Q$.

[Figure 1, (a), (b)]

Since under the base stock policy the lengths of the review period and the replenishment lead time are determined exogenously, the only control variable is the base stock level. To determine the optimal base stock level for a single stage inventory system, let us denote the review period as p, the replenishment lead time as l and the average demand at each unit of time is μ. Recall that the inventory position is the total material in the system (on-hand plus on-order), and we start each review period with the same inventory position S, which is the base stock level. We must wait p units of time to review the inventory position again and place an order for replenishment, and then the order will take another l units of time to arrive (Figure 1). Therefore, the inventory in the system at the beginning of a review period should be large enough to cover the demand over review period p plus the replenishment lead time l, i.e. the optimal base stock level should be $\mu(p+l)$ if demand is constant.

[Figure 2]

However, under demand uncertainty the demand process is not constant and we need more inventory (safety stock) to hedge against stockout before we get a chance to reorder (Figure 2). The acceptable practice in this field is to assume a normal distribution of the demand, although of course other distribution functions can be specified. If the demand at each unit of time is normally distributed with mean μ and standard deviation σ, the demand over review period p and the replenishment lead time l is also normally distributed with mean $\mu(p+l)$ and standard deviation $\sigma\sqrt{p+l}$. It is convenient to measure safety stock in terms of the number of standard deviations of demand, denoted as safety stock factor, λ. Then the optimal base stock level is given by,

$$S = \mu(p+l) + \lambda\sigma\sqrt{p+l}$$

We should note that if α is the *Type I service level* (the probability that the total inventory on hand is more than the demand) used to measure of service level, the safety stock factor λ corresponds to the α-quantile of the standard normal distribution, i.e. $\Pr(x \le \lambda) = \alpha$.

2.2. Risk Pooling Effect

For single echelon inventory system with multiple stocking locations, Eppen[31] proposed the "risk pooling effect", which states that significant safety stock cost can be saved by grouping the demand of multiple stocking locations. In particular, Eppen considers a single period problem with N retailers and one supplier. Each retailer i has uncorrelated normally distributed demand with mean μ_i and standard deviation σ_i. The replenishment lead times for all these retailers are the same and given as L and all the retailers guarantee the same *Type I service level* with the same safety stock factor λ. Eppen compared two operational modes of the N-retailer system: decentralized mode and centralized mode. In the decentralized mode, each retailer orders independently to minimize its cost. Since in this mode the optimal safety stock in retailer i corresponding to the safety stock factor λ is $\lambda\sqrt{L}\sigma_i$,[31] the total safety stock in the system is given by,

$$\lambda\sqrt{L}\sum_{i=1}^{N}\sigma_i .$$

In the centralized mode, all the retailers are considered as a whole and a single quantity is ordered for replenishment, so as to minimize the total expected cost of the entire system. Since in the centralized mode all the retailers are grouped, and the demand at each retailer follows a normal distribution $N(\mu_i, \sigma_i^2)$, the total uncertain demand of the entire system during the order lead time will also follow a normal distribution with mean $L\sum_{i=1}^{N}\mu_i$ and standard deviation $\sqrt{L}\sqrt{\sum_{i=1}^{N}\sigma_i^2}$. Therefore, the total safety stock of the distribution centers in the centralized mode is given by,

$$\lambda\sqrt{L}\sqrt{\sum_{i=1}^{N}\sigma_i^2}$$

which is less than $\lambda\sqrt{L}\sum_{i=1}^{N}\sigma_i$. Eppen's simple model illustrates the potential saving in safety stock costs due to risk pooling. For example, consider a single echelon inventory system with 100 retailers. Each retailer has uncorrelated normal distributed demand with identical mean μ and standard deviation σ. Thus, the total safety stock of this system is $100z\sigma\sqrt{L}$ under decentralized mode, and $10z\sigma\sqrt{L}$ under centralized mode, i.e. 90% safety stocks in this system can be saved by risk pooling.

2.3. Guaranteed Service Model for Multi-Echelon Inventory

One of the most significant differences between single stage inventory system and multi-echelon inventory system is the lead time. For single stage inventory system, lead time, which may include material handling time and transportation time, is exogenous and generally can be treated as a constant. However, for a multi-echelon inventory system, lead time of a downstream node depends on the upstream node's inventory level and demand uncertainty, and thus the lead time and internal service level are stochastic. Based on this fact, simply propagating the single stage inventory model to multi-echelon system will lead to a suboptimal solution, since the optimization of a multi-echelon inventory system is usually nontrivial.[3]

There are two major approaches to model the multi-echelon inventory system, the stochastic service approach and the guaranteed service approach.[16] For a detailed comparison of these two approaches, see Graves and Willems,[16] and Humair and Willems.[18] Briefly, the stochastic service approach employs a more complicated model that allows for a more exact and detailed understanding of the system. However, the model as well as the solution technique, are not easy to develop and is computationally hard. The guaranteed service approach models the entire system in an approximate fashion and allows a planner to make strategic and tactical decisions without the need to approximate portions of the system that are not captured by a simplified topological representation. Thus, the stochastic service model may be too complicated to be employed for supply chain design problem. Based on this reason, we choose the guaranteed service approach to model the multi-echelon inventory system in this work.

The main idea of the guaranteed service approach is that each node *j* in the multi-echelon inventory system quotes a guaranteed service time T_j, by which this node will satisfy the demands from its downstream customers. That is, the customer demand at time t must be ready to be shipped by time $t+T_j$. For instance, if a chemical supplier's guaranteed service time is 3 days, it means that once the customer places an order, the total amount of chemicals ordered by the customer will be available in "at most" 3 days. The guaranteed service times for internal customers are decision variables to be optimized, while the guaranteed service time for the nodes at the last echelon (facing external customers) is an exogenous input, presumably set by the marketplace. Besides the guarantee service time, we consider that each node has a given deterministic order processing time, t_j, which is independent of the order size. The order processing time, including material handling time, transportation time and review period, represents the time from all the inputs that are available until the outputs are available to serve the demand. Therefore, the replenishment lead time, which represents the time from when we place an order to when all the goods are received, can be determined by the guaranteed service time of direct predecessor T_{j-1} (the time that predecessor requires to have the chemicals ready to be shipped) plus the order processing time t_j (transportation time, handling time and review period). As we can see from Figure 3, the net lead time of node *j*, NLT_j, which is the time span over which safety stock coverage against demand variations is necessary, is defined as the difference between the replenishment lead time of this node and its guaranteed service time to its successor.[15] The reason is that not all the customer demand at time t for node *j* should be satisfied immediately, but only needs to be ready by time $t+T_j$. Thus, the safety stock does not need to cover demand variations over the whole replenishment lead time, but just the difference between the replenishment lead time and the guaranteed service time to the successors that is defined as the net lead time. Therefore, we can calculate the net lead time with the following formula:

$$NLT_j = T_{j-1} + t_j - T_j$$

where node *j*-1 is the direct predecessor of node *j* (Figure 4). Note that this formula suggests that if the service time quoted by node *j* to its successor nodes equals the replenishment lead time $T_{j-1} + t_j$, no safety stock is required in node *j* because all products are received from the predecessors and processed within the guaranteed service time, i.e. this node is operating in "pull" mode. If the guaranteed service time T_j is 0, the node holds the most safety stock because all the orders, once they are placed, are fulfilled immediately, i.e. this node is operating in “push” mode.

[Figure 3]
[Figure 4]

In the guaranteed service approach, each node in the multi-echelon inventory system is assumed to operate under a periodic review base stock policy with a common review period. Furthermore, demand over any time interval is also assumed to be normally distributed (mean μ_j and standard deviation σ_j for node j), and “bounded”. This does not imply that demand can never exceed the bound, but this bound reflects the maximum amount of demand that a company wants to satisfy from safety stock. This interpretation is consistent with most practical applications that safety stocks are used to cope with regular demand variations that do not exceed a maximum reasonable bound. Larger demand variations are handled with extraordinary measures, such as spot-market purchases, rescheduling or expediting orders, other than using safety stocks. Thus, under these assumptions, each node sets its base stock so as to meet *all* the orders from its downstream customers within the guaranteed service time. That is, for node *j*, there is an associated safety stock factor λ_j, which is given and corresponds to the maximum amount of demand that company wants to satisfy from safety stock. This yields the base stock level at node *j*:

$$S_j = \mu_j NLT_j + \lambda_j \sigma_j \sqrt{NLT_j}$$

This formula is similar but slightly different from the single stage inventory model in terms of the expression for the lead time. Note that the review period has been taken into account as part of the order processing time and considered in the net lead time.

With the guaranteed service approach, the total inventory cost consists of safety stock cost and pipeline inventory cost. The safety stock of node *j* (SS_j) is given by the following formula as discussed above,

$$SS_j = \lambda_j \sigma_j \sqrt{NLT_j}$$

The expected pipeline inventory is the sum of expected on hand and on-order inventories. Based on Little’s law,[33] the expected pipeline inventory PI_j of node *j* equals to the mean demand over the order processing time, and is given by,

$$PI_j = t_j \mu_j$$

which is not affected by the coverage and guaranteed service time decisions.

3. Problem Statement

We are given a potential supply chain (Figure 5) consisting of a set of plants (or suppliers) $i \in I$, a number of candidate sites for distribution centers $j \in J$, and a set of customer demand zones $k \in K$ whose inventory costs should be taken into account. The customer demand zone can represent a local distributor, a regional warehouse, a dealer, a retailer, or a wholesaler, which is usually a necessary component of the supply chain for specialty chemicals or advanced materials.[19, 34] Alternatively, one might view the customer demand as the aggregation of a group of customers operated with vendor managed inventory (vendor takes care of customers' inventory), which is a common business model in the industrial gases industry and some chemical companies.[35, 36]

[Figure 5]

In the given potential supply chain, the locations of the plants, potential distribution centers and customer demand zones are known and the distances between them are given. The investment costs for installing DCs are expressed by a cost function with fixed charges. Each retailer i has an uncorrelated normally distributed demand with mean μ_i and variance σ_i^2 in each unit of time. Single sourcing restriction, which is common in the specialty chemicals supply chain,[37] is employed for the distribution from plants to DCs and from DCs to customer demand zones. That is, each DC is only served by one plant, and each customer demand zone is only assigned to one DC to satisfy the demand. Linear transportation costs are incurred for shipments from plant i to distribution center j with unit cost $c1_{ij}$, and from distribution center j to customer demand zone k with unit cost $c2_{jk}$. The corresponding deterministic order processing times of DCs and customer demand zone that includes the material handling time, transportation time and inventory review period, are given by $t1_{ij}$ and $t2_{jk}$. The service time of each plant, and the maximum service time of each customer demand zones are known. We are also given the safety stock factor for DCs and customer demand zones, $\lambda 1_j$ and $\lambda 2_k$, which correspond to the standard normal deviate of the maximum amount of demand that the node will satisfy from its safety stock. A common review period is used for the control of inventory in each node. Inventory costs are incurred at distribution centers and customers, and consist of pipeline inventory and safety stock, of which the unit costs are given.

The objective is to determine how many distribution centers (DCs) to install, where to locate them, which plants to serve each DC and which DCs to serve each customer demand zone, how long should each DC quote its service time, and what level of safety stock to maintain at each DC and customer demand zone so as to minimize the total installation, transportation, and inventory costs.

4. Model Formulation

The joint multiechelon supply chain design and inventory management model is a mixed-integer nonlinear program (MINLP) that deals with the supply chain network design for a given product, and considers its multi-echelon inventory management. The definition of sets, parameters, and variables of the model is given below:

Sets/Indices

I Set of plants (suppliers) indexed by i

J Set of candidate DC locations indexed by j

K Set of customer demand zones (wholesaler, regional distributor, dealer, retailer, or customers with vendor managed inventory) indexed by k

Parameters

$c1_{ij}$ Unit transportation cost from plant i to DC j

$c2_{jk}$ Unit transportation cost from DC j to customer demand zone k

f_j Fixed cost of installing a DC at candidate location j (annually)

g_j Variable cost coefficient of installing candidate DC j (annually)

$h1_j$ Unit inventory holding cost at DC j (annually)

$h2_k$ Unit inventory holding cost at customer demand zone k (annually)

R_k Maximum guaranteed service time to customers at customer demand zone k

SI_i Guaranteed service time of plant i

$t1_{ij}$ Order processing time of DC *j* if it is served by plant *i*, including material handling time of DC *j*, transportation time from plant *i* to DC *j*, and inventory review period

$t2_{jk}$ Order processing time of customer demand zone *k* if it is served by DC *j*, including material handling time of DC *j*, transportation time from DC *j* to customer demand zone *k*, and inventory review period

μ_k Mean demand at customer demand zone k (daily)

σ_k^2 Variance of demand at customer demand zone k (daily)

χ Days per year (to convert daily demand and variance values to annual costs)

$\theta 1_{ij}$ Unit cost of pipeline inventory from plant i to DC j (annual)

$\theta 2_{jk}$ Unit cost of pipeline inventory from DC j to customer demand zone k (annual)

$\lambda 1_j$ Safety stock factor of DC j

$\lambda 2_k$ Safety stock factor of customer demand zone k

Binary Variables (0-1)

X_{ij} 1 if DC j is served by plant i, and 0 otherwise

Y_j 1 if we install a DC in candidate site j, and 0 otherwise

Z_{jk} 1 if customer demand zone k is served by DC j, and 0 otherwise

Continuous Variables (0 to $+\infty$)

L_k Net lead time of customer demand zone k

N_j Net lead time of DC j

R_k Guaranteed service time of customer demand zone k

S_j Guaranteed service time of DC j to its successive customer demand zones

4.1. Objective Function

The objective of this model is to minimize the total supply chain design cost including the following items:

- Installation costs of DCs
- transportation costs from plants to DCs and from DCs to customer demand zones
- pipeline inventory costs in DCs and customer demand zones
- safety stock costs in DCs and customer demand zones

The cost of installing a DC in candidate location *j* is expressed by a fixed-charge cost model that captures the economies of scale in the investment. The annual expected demand of DC *j* is ($\sum_{k \in K} \chi Z_{jk} \mu_k$), which equals to the annual mean demand of all the customer demand zones served by DC *j*. Hence, the cost of installing DC *j* consists of fixed cost f_j and variable cost ($g_j \sum_{k \in K} \chi Z_{jk} \mu_k$), which is the product of variable cost coefficient and the expected demand of this DC in one year. Thus, the total installation costs of all the DCs is given by,

$$\sum_{j\in J} f_j Y_j + \sum_{j\in J}\left(g_j \sum_{k\in K} \chi Z_{jk}\mu_k \right) \tag{1}$$

The product of the annual mean demand of DC j ($\sum_{k\in K}\chi Z_{jk}\mu_k$) and the unit transportation cost ($\sum_{i\in I} c1_{ij} X_{ij}$) between DC j and the plant that serves it yields the annual plant to DC transportation cost.

$$\sum_{i\in I}\sum_{j\in J}\left(c1_{ij} X_{ij} \sum_{k\in K} \chi Z_{jk}\mu_k \right) \tag{2}$$

Similarly, the product of yearly expected mean demand of customer demand zone k ($\chi\mu_k$) and the unit transportation cost ($\sum_{j\in J} c2_{jk} Z_{jk}$) between customer demand zone k and the DC that serves it, leads to the annual DC to customer demand zone transportation cost.

$$\sum_{j\in J}\sum_{k\in K} c2_{jk}\chi Z_{jk}\mu_k \tag{3}$$

Based on Little's law,[33] the pipeline inventory PI_j of DC j equals to the product of its daily mean demand ($\sum_{k\in K} Z_{jk}\mu_k$) and its order processing time ($\sum_{i\in I} t1_{ij} X_{ij}$), which is in terms of days. Thus, the annual total pipeline inventory cost of all the DCs is given by,

$$\sum_{i\in I}\sum_{j\in J}\left(\theta 1_j t1_{ij} X_{ij} \sum_{k\in K} Z_{jk}\mu_k \right) \tag{4}$$

where $\theta 1_j$ is the annual unit pipeline inventory holding cost of DC j.

Similarly, the total annual pipeline inventory cost of all the customer demand zones is given by,

$$\sum_{j\in J}\sum_{k\in K} \theta 2_k t2_{jk} Z_{jk}\mu_k \tag{5}$$

where $\theta 2_k$ is the annual unit pipeline inventory holding cost of customer demand zone k, and $\sum_{k\in K} t2_{jk} Z_{jk}\mu_k$ is the pipeline inventory of customer demand zone k.

The demand at customer demand zone k follows a given normal distribution with mean μ_k and variance σ_k^2. Due to the risk-pooling effect,[31] the demand over the net lead time (N_j) at DC j is also normally distributed with a mean of $N_j\sum_{k\in J_k}\mu_k$ and a variance of $N_j\sum_{k\in J_k}\sigma_k^2$, where J_k is the set of customer demand zones k assigned to DC j. Thus, the safety stock required in the DC at candidate location j with a safety stock factor $\lambda 1_j$ is

$\lambda 1_j \sqrt{N_j} \cdot \sqrt{\sum_{k \in K} \sigma_k^2 Z_{jk}}$. Considering the annual inventory holding cost at DC *j* is $h1_j$, we have the annual total safety stock cost at all the DCs equals to

$$\sum_{j \in J} \lambda 1_j h1_j \sqrt{N_j} \cdot \sqrt{\sum_{k \in K} \sigma_k^2 Z_{jk}} \tag{6}$$

Similarly, the demand over the net lead time of customer demand zones *k* (L_k) is normally distributed with a mean of $L_k \mu_k$ and a variance of $L_k \sigma_k^2$. Thus, the annual safety stock cost at all the customer demand zones is given by,

$$\sum_{k \in K} \lambda 2_k h2_k \cdot \sigma_k \sqrt{L_k} \tag{7}$$

Therefore, the objective function of this model (the total supply chain design cost) is given by

$$\begin{aligned} \min: \quad & \sum_{j \in J} f_j Y_j + \sum_{j \in J} \left(g_j \sum_{k \in K} \chi Z_{jk} \mu_k \right) \\ & + \sum_{i \in I} \sum_{j \in J} \left(c1_{ij} X_{ij} \sum_{k \in K} \chi Z_{jk} \mu_k \right) + \sum_{j \in J} \sum_{k \in K} c2_{jk} \chi Z_{jk} \mu_k \\ & + \sum_{i \in I} \sum_{j \in J} \left(\theta 1_j t1_{ij} X_{ij} \sum_{k \in K} Z_{jk} \mu_k \right) + \sum_{j \in J} \sum_{k \in K} \theta 2_k t2_{jk} Z_{jk} \mu_k \\ & + \sum_{j \in J} \lambda 1_j h1_j \sqrt{N_j} \cdot \sqrt{\sum_{k \in K} \sigma_k^2 Z_{jk}} + \sum_{k \in K} \lambda 2_k h2_k \cdot \sigma_k \sqrt{L_k} \end{aligned} \tag{8}$$

where each term accounts for the DC installation cost, transportation costs of plants-DCs and DCs-customer demand zones, pipeline inventory costs of DCs and customer demand zones, and safety stock costs of DCs and customer demand zones.

4.2. Constraints

Three constraints are used to define the network structure. The first one is that if DC *j* is installed, it should be served by only one plant. If it is not installed, it is not assigned to any plant. This can be modeled by,

$$\sum_{i \in I} X_{ij} = Y_j, \qquad \forall j \tag{9}$$

The second constraint states that each customer demand zone *k* is served by only one DC,

$$\sum_{j \in J} Z_{jk} = 1, \qquad \forall k \tag{10}$$

The third constraint is that if a customer demand zone *k* is served by the DC in candidate location j, the DC must exist,

$$Z_{jk} \le Y_j, \qquad \forall j, k \tag{11}$$

Two constraints are used to define the net lead time of DCs and customer demand zones. The replenishment lead time of DC *j* should be equal to the guaranteed service time (SI_i) of plant *i*, which serves DC *j*, plus the order processing time (t_{ij}). Since each DC is served by only one plant, the replenishment lead time of DC *j* is calculated by $\sum_{i \in I}(SI_i + t_{ij}) \cdot X_{ij}$. Thus, the net lead time of DC *j* should be greater than its replenishment lead time minus its guaranteed service time to its successor customer demand zones, and is given by the linear inequality,

$$N_j \geq \sum_{i \in I}(SI_i + t_{ij}) \cdot X_{ij} - S_j, \quad \forall j \tag{12}$$

Similarly, the net lead time of a customer demand zone *k* is greater than its replenishment lead time minus its maximum guaranteed service time, R_k, which is given by the nonlinear inequalitiy

$$L_k \geq \sum_{j \in J}(S_j + t_{jk}) \cdot Z_{jk} - R_k, \quad \forall k \tag{13}$$

Finally, all the decision variables for network structure are binary variables, and the variables for guaranteed service time and net lead time are non-negative variables.

$$X_{ij}, Y_j, Z_{jk} \in \{0,1\}, \qquad \forall i, j, k \tag{14}$$

$$S_j \geq 0, \; N_j \geq 0, \qquad \forall j \tag{15}$$

$$L_k \geq 0, \qquad \forall k \tag{16}$$

4.3. MINLP Model

Grouping the parameters, we can rearrange the objective function and formulate the problem as the following mixed-integer nonlinear program (P0):

$$\text{Min: } \sum_{j \in J} f_j Y_j + \sum_{i \in I}\sum_{j \in J}\sum_{k \in K} A_{ijk} X_{ij} Z_{jk} + \sum_{j \in J}\sum_{k \in K} B_{jk} Z_{jk} + \sum_{j \in J} q1_j \sqrt{N_j \sum_{k \in K} \sigma_k^2 Z_{jk}} + \sum_{k \in K} q2_k \sqrt{L_k} \tag{17}$$

$$\text{s.t. } \sum_{i \in I} X_{ij} = Y_j, \qquad \forall j \tag{9}$$

$$\sum_{j \in J} Z_{jk} = 1, \qquad \forall k \tag{10}$$

$$Z_{jk} \leq Y_j, \qquad \forall j, k \tag{11}$$

$$N_j \geq \sum_{i \in I} \bar{S}_{ij} \cdot X_{ij} - S_j, \; \forall j \tag{12}$$

$$L_k \geq \sum_{j \in J}(S_j + t_{jk}) \cdot Z_{jk} - R_k, \qquad \forall k \tag{13}$$

$$X_{ij}, Y_j, Z_{jk} \in \{0,1\}, \qquad \forall i, j, k \tag{14}$$

$$S_j \geq 0,\ N_j \geq 0, \qquad \forall j \qquad (15)$$

$$L_k \geq 0, \qquad \forall k \qquad (16)$$

where

$\overline{S}_{ij} = SI_i + t_{ij}$

$A_{ijk} = \left(c1_{ij}\chi + \theta 1_j t1_{ij}\right) \cdot \mu_k$

$B_{jk} = \left(g_j \chi + c2_{jk}\chi + \theta 2_k t2_{jk}\right) \cdot \mu_k$

$q1_j = \lambda 1_j \cdot h1_j$

$q2_k = \lambda 2_k \cdot h2_k \cdot \sigma_k$

4.4. Reformulated MINLP Model

The MINLP model (P0) has nonconvex terms, including bilinear and square root terms, in its objective function (17) and constraint (13). Because the bilinear terms are the products of two binary variables (such as $X_{ij} \cdot Z_{jk}$), or the product of a binary variable and a continuous variable (such as $N_j \cdot Z_{jk}$ and $S_j \cdot Z_{jk}$), they can be linearized by introducing additional variables.

To linearize the term ($X_{ij} \cdot Z_{jk}$) in the objective function (17), we first introduce a new continuous non-negative variable XZ_{ijk}. Then the product of X_{ij} and Z_{jk} can be replaced by this term XZ_{ijk} with the following constraints,[38]

$$XZ_{ijk} \leq X_{ij}, \qquad \forall i, j, k \qquad (19)$$

$$XZ_{ijk} \leq Z_{jk}, \qquad \forall i, j, k \qquad (20)$$

$$XZ_{ijk} \geq X_{ij} + Z_{jk} - 1, \qquad \forall i, j, k \qquad (21)$$

$$XZ_{ijk} \geq 0, \qquad \forall i, j, k \qquad (22)$$

where constraints (19), (20) and (22) ensure that if X_{ij} or Z_{jk} is zero, XZ_{ijk} should be zero, and constraint (21) ensures that if X_{ij} and Z_{jk} are both equal to one, XZ_{ijk} should be one.

The linearization of ($S_j \cdot Z_{jk}$) in constraint (13) requires two new continuous non-negative variable SZ_{jk} and $SZ1_{jk}$, and the following constraints,[38]

$$SZ_{jk} + SZ1_{jk} = S_j, \qquad \forall j, k \qquad (23)$$

$$SZ_{jk} \leq Z_{jk} \cdot S_j^U, \qquad \forall j, k \qquad (24)$$

$$SZ1_{jk} \le (1 - Z_{jk}) \cdot S_j^U, \qquad \forall j,k \tag{25}$$

$$SZ_{jk} \ge 0, \; SZ1_{jk} \ge 0, \qquad \forall j,k \tag{26.1}$$

where constraints (24), (25) and (26.1) ensure that if Z_{jk} is zero, SZ_{jk} should be zero; if Z_{jk} is one, $SZ1_{jk}$ should be zero. Combining with constraint (23), we can have SZ_{jk} equivalent to the product of S_j and Z_{jk}.

Similarly, the product of N_j and Z_{jk} in the objective function (17) can be linearized as follows,

$$NZ_{jk} + NZ1_{jk} = N_j, \qquad \forall j,k \tag{27}$$

$$NZ_{jk} \le Z_{jk} \cdot N_j^U, \qquad \forall j,k \tag{28}$$

$$NZ1_{jk} \le (1 - Z_{jk}) \cdot N_j^U, \qquad \forall j,k \tag{29}$$

$$NZ_{jk} \ge 0, \; NZ1_{jk} \ge 0, \qquad \forall j,k \tag{26.2}$$

where NZ_{jk} and $NZ1_{jk}$ are two new continuous variables, and NZ_{jk} is equivalent to ($N_j \cdot Z_{jk}$).

The above linearizations introduce more constraints and variables, but significantly reduces the number of nonlinear terms in the model (P0) and potentially reduce the computational effort. To further reduce the nonlinear terms in the objective function (17), the term ($N_j \cdot \sum_{k \in K} \sigma_k^2 \cdot Z_{jk}$) is replaced by a new nonnegative continuous variable NZV_j with the following constraint,

$$NZV_j = \sum_{k \in K} \sigma_k^2 \cdot NZ_{jk}, \qquad \forall j \tag{30}$$

$$NZV_j \ge 0, \qquad \forall j \tag{31}$$

Therefore, incorporating the above linearizations we have the following reformulated MINLP model (P1).

$$\text{Min: } \sum_{j \in J} f_j Y_j + \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} A_{ijk} XZ_{ijk} + \sum_{j \in J} \sum_{k \in K} B_{jk} Z_{jk} + \sum_{j \in J} q1_j \sqrt{NZV_j} + \sum_{k \in K} q2_k \sqrt{L_k} \tag{17}$$

$$\text{s.t. } \sum_{i \in I} X_{ij} = Y_j, \qquad \forall j \tag{9}$$

$$\sum_{j \in J} Z_{jk} = 1, \qquad \forall k \tag{10}$$

$$Z_{jk} \le Y_j, \qquad \forall j,k \tag{11}$$

$$N_j \geq \sum_{i \in I} \overline{S}_{ij} \cdot X_{ij} - S_j\,, \ \forall j \tag{18}$$

$$L_k \geq \sum_{j \in J} SZ_{jk} + \sum_{j \in J} t_{jk} \cdot Z_{jk} - R_k\,, \forall k \tag{32}$$

$$XZ_{ijk} \leq X_{ij}\,, \qquad \forall i, j, k \tag{19}$$

$$XZ_{ijk} \leq Z_{jk}\,, \qquad \forall i, j, k \tag{20}$$

$$XZ_{ijk} \geq X_{ij} + Z_{jk} - 1\,, \qquad \forall i, j, k \tag{21}$$

$$SZ_{jk} + SZ1_{jk} = S_j\,, \qquad \forall j, k \tag{23}$$

$$SZ_{jk} \leq Z_{jk} \cdot S_j^U\,, \qquad \forall j, k \tag{24}$$

$$SZ1_{jk} \leq (1 - Z_{jk}) \cdot S_j^U\,, \qquad \forall j, k \tag{25}$$

$$NZ_{jk} + NZ1_{jk} = N_j\,, \qquad \forall j, k \tag{27}$$

$$NZ_{jk} \leq Z_{jk} \cdot N_j^U\,, \qquad \forall j, k \tag{28}$$

$$NZ1_{jk} \leq (1 - Z_{jk}) \cdot N_j^U\,, \qquad \forall j, k \tag{29}$$

$$NZV_j = \sum_{k \in K} \sigma_k^2 \cdot NZ_{jk}\,, \qquad \forall j \tag{30}$$

$$X_{ij}, Y_j, Z_{jk} \in \{0,1\}\,, \qquad \forall i, j, k \tag{14}$$

$$S_j \geq 0\,, \ N_j \geq 0\,, \qquad \forall j \tag{15}$$

$$L_k \geq 0\,, \qquad \forall k \tag{16}$$

$$XZ_{ijk} \geq 0\,, \qquad \forall i, j, k \tag{22}$$

$$NZV_j \geq 0\,, \qquad \forall j \tag{31}$$

$$SZ_{jk} \geq 0\,, \ SZ1_{jk} \geq 0\,, \ NZ_{jk} \geq 0\,, \ NZ1_{jk} \geq 0\,, \qquad \forall j, k \tag{26}$$

Compared with model (P0), model (P1) is computationally more tractable, because all the constraints in (P1) are linear and the only nonlinear terms are univariate concave terms in the objective function.

4.5. Variable Bounds and Initialization

The bounds of variables are quite important for nonlinear optimization problems. The upper bounds of the key continuous variables are derived below (denoted with the superscript *U*).

The maximum net lead time (N_j^U) of DC *j* is the maximum value of the sum of service time of plant *i* and the order processing time from plant *i* to DC *j*. It means that when the guaranteed service time of DC *j* is zero and it is served by plant *i* that has the maximum ($SI_i + t_{ij}$), the net lead time N_j equals to the maximum value,

$$N_j^U = \max_{i \in I}\left\{SI_i + t_{ij}\right\} = \max_{i \in I}\left\{\bar{S}_{ij}\right\}, \quad \forall j \tag{33.1}$$

The maximum value of guaranteed service time of DC j is equal to the maximum value of its net lead time. It means that when the net lead time of DC *j* is zero and it is assigned to plant *i* that has the maximum ($SI_i + t_{ij}$), the net lead time S_j equals to the maximum value,

$$S_j^U = N_j^U = \max_{i \in I}\left\{\bar{S}_{ij}\right\}, \qquad \forall j \tag{33.2}$$

From constraints (13) and (16), it is easy to see that the maximum net lead time of customer demand zone *k* is as follows,

$$L_k^U = \max\left\{0, \max_{j \in J}\left\{S_j^U + t_{jk}\right\} - R_k\right\}, \forall j \tag{33.3}$$

The upper bounds of the auxiliary variables are easy to derive,

$$SZ_{jk}^U = S_j^U, \ SZ1_{jk}^U = S_j^U, \ \forall j, k \tag{33.4}$$

$$NZ_{jk}^U = N_j^U, \ NZ1_{jk}^U = N_j^U, \forall j, k \tag{33.5}$$

$$NZV_j^U = \sum_{k \in K} \sigma_k^2 \cdot N_j^U, \qquad \forall j \tag{33.6}$$

Besides variable bounds, the initial point is another key issue for numerical optimization of MINLP problems. Since model (P1) has some special structure (linear constraints, univariate concave terms in the objective function), we can use a similar approach as in You and Grossmann[30] to obtain a "good" starting point by solving a mixed-integer linear program (MILP) problem for initialization purposes.

[Figure 6]

As introduced by Falk and Soland,[39] for a univariate square root term $\sqrt{x}$, where the variable x has lower bound 0 and upper bound x^U, its secant $x / \sqrt{x^U}$ represents the convex envelope and provides a valid lower bound of the square root term as shown in Figure 6. Since model (P1) is a minimization problem and all the constraints are linear, replacing all the univariate square root terms with their secants in the objective function (17) will lead to the MILP problem (P2) with a linear objective function,

$$\text{Min:} \ \sum_{j \in J} f_j Y_j + \sum_{i \in I}\sum_{j \in J}\sum_{k \in K} A_{ijk} XZ_{ijk} + \sum_{j \in J}\sum_{k \in K} B_{jk} Z_{jk} + \sum_{j \in J} \frac{q1_j \cdot NZV_j}{\sqrt{NZV_j^U}} + \sum_{k \in K} \frac{q2_k \cdot L_k}{\sqrt{L_k^U}} \tag{34}$$

s.t. all the constraints of (P1)

Since (P1) and (P2) have the same constraints, a feasible solution of (P2) is also a feasible solution of (P1). Furthermore, the optimal objective function value of (P2) is a valid lower bound of the optimal objective function value of (P1). Therefore, by solving the initialization MILP problem (P2), we can obtain a “good” starting point of solving problem (P1).

Note that based on this initialization procedure, a heuristic algorithm is to first solve problem (P2), then using its optimal solution as an initial point to solve problem (P1) with an MINLP solver that relies on convexity assumption (e.g. DICOPT, SBB, etc.). This algorithm can obtain “good” solutions quickly, although global optimality cannot be guaranteed.

5. Illustrative Example

To illustrate the application of this model, we consider two small illustrative examples, an industrial liquid oxygen (LOX) supply chain example and an acetic acid supply chain example.

5.1. Example A: Industrial LOX Supply Chain

The first example is for an industrial LOX supply chain with two plants, three potential DCs and six customers as given in Figure 7. In this supply chain, customer inventories are managed by the vendor, i.e. vendor-management-inventory (VMI), which is a common business model in the gas industry.[34-36] Thus, in this example inventory costs from DCs and customers should be taken into account into the total supply chain cost, and the joint multiechelon supply chain design and inventory management model can be used to minimize total network design, transportation and inventory costs.

[Figure 7]

In this instance, the annual fixed costs to install the DCs (f_j) are \$10,000/year, \$8,000/year and \$12,000/year, respectively. The variable cost coefficient of installing a DC (g_j) at all the candidate locations is \$0.01/(liter · year). The safety stock factors for DCs ($\lambda 1_j$) and customers ($\lambda 2_k$) are the same and equal to 1.96, which corresponds to 97.5% service level if demand is normally distributed. We consider 365 days in a year (χ). The guaranteed service times of the two plants (SI_i) are 2 days and 3 days, respectively. Since the last echelon represents the customers, the guaranteed service time of customers (R_k) are set to 0. The data for demand uncertainty, order processing times and transportation costs are given in Tables 1-5.

[Table 1]
[Table 2]
[Table 3]
[Table 4]
[Table 5]

We consider two instances of this example. In the first instance, we consider zero holding cost for both pipeline inventory and safety stocks, i.e. the problem reduces to a supply chain network design problem without considering inventory cost. In the second instance, the pipeline inventory holding cost of LOX is \$6/(liter · year) for all the DCs ($\theta 1_{ij}$) and customers ($\theta 2_{jk}$), and the safety stock holding cost is \$8/(liter · year) for all the DCs ($h1_j$) and customers ($h2_k$).

Both the initialization MILP model (P2) and the MINLP model (P1) include 27 binary variables, 124 continuous variables and 256 constraints. Since the problem sizes are relatively small, we solve model (P1) directly to obtain the global optimum with 0% optimality margin by using the BARON solver[40] with GAMS,[41] and the initialization MILP model (P2) is solved with GAMS/CPLEX. The resulting optimal supply chain networks with a minimum annual cost of \$108,329/year and the flow rate of each transportation link are given in Figure 8. We can see that for the instance without considering inventory costs, all the three DCs are installed, and each of them serves two customers. The major trade-off is between DC installation costs and the transportation costs. For the instance that takes into account the inventory costs, only two DCs are installed and each serves three customers. The major trade-off is between DC installation costs, transportation costs and inventory costs. The minimum annual cost in this instance is \$152,107/year with the components given in Figure 9. A comparison between these two instances suggests the importance of integrating inventory costs in the supply chain network design. Although the inventory costs might only make up a small portion of the total costs, the optimal network structure could be quite different with and without considering inventory in the supply chain design.

[Figure 8]
[Figure 9]

5.2. Example B: Acetic Acid Supply Chain

The second example is for an acetic acid supply chain with three plants, three potential DCs and four customer demand zones, each of which has a wholesaler (Figure 10). Based on the conclusion of the previous example, we need to take into account the inventory costs from DCs and wholesalers when designing this supply chain, so as to obtain a more accurate "optimal" supply chain.

[Figure 10]

In this instance, the annual fixed DC installation cost (f_j) is \$50,000/year for all the DCs. The variable cost coefficient of installing a DC (g_j) at all the candidate locations is \$0.5/(ton · year). The safety stock factors for DCs ($\lambda 1_j$) and wholesalers ($\lambda 2_k$) are the same and equal to 1.96. We consider 365 days in a year (χ). The guaranteed service times of the three vendors (SI_i) are 3 days, 3 days and 4 days, respectively. The pipeline inventory holding cost is \$1/(ton · day) for all the DCs ($\theta 1_{ij} / \chi$) and wholesalers ($\theta 2_{jk} / \chi$), and the safety stock holding cost is \$1.5/(ton · day) for all the DCs ($h1_j / \chi$) and wholesalers ($h2_k / \chi$). Since the guaranteed service time of wholesalers to end customers (R_k) are set to be 8 days. The data for demand uncertainty, order processing times and transportation costs are given in Tables 6-10.

[Table 6]
[Table 7]
[Table 8]
[Table 9]
[Table 10]

The computational study is carried out on an IBM T40 laptop with Intel 1.50GHz CPU and 512 MB RAM. The original model (P0) has 12 discrete variables, 22 continuous variables and 27 constraints. Both the initialization MILP model (P2) and the MINLP model (P1) include 24 binary variables, 185 continuous variables and 210 constraints. We solve the original model (P0) by using GAMS/BARON with 0% optimality margin, and it takes a total of 1376.9 CPU seconds. We then solve the reformulated model (P1) with the aforementioned initialization process, the CPU time reduces to only 7.6 seconds and the optimal solutions are the same as we obtained by solving model (P0). The possible reason is that initialization process helps BARON to find a "good" feasible solution during the preprocessing step. This feasible solution provides a tighter upper bound that reduces the searching space and speeds up the computation.

The resulting optimal supply chain networks with a minimum annual cost of $ 1,986,148/year and the flow rate of each transportation link are given in Figure 11. We can see that two DCs are selected to install and one of them serves three wholesalers while the other one only serves one wholesaler. The major trade-off is between DC installation costs, transportation costs and inventory costs. A detailed breakdown of the total cost is given in Figure 12.

[Figure 11]
[Figure 12]

6. Solution Algorithm

Although small scale problems can be solved to global optimality effectively by using a global optimizer and the aforementioned initialization method, medium and large-scale joint multi-echelon supply chain design and inventory management problems are often computationally intractable with direct solution approaches due to the combinatorial nature and nonlinear nonconvex terms. In this section, we present an effective solution algorithm based on Lagrangean relaxation and piecewise linear approximation to obtain solutions within 1% of global optimality gap with reasonable computational expense.

6.1. Piecewise Linear Approximation

Instead of using the secants in the objective function of model (P1), a tighter lower convex envelope of the univariate square root terms is the piecewise linear function, which employs a few more continuous and discrete variables. Although this may require longer computational times to solve the initialization MILP problem, with the significant progress of MILP solvers, piecewise linear approximations have recently been increasingly used for approximating different types of nonconvex nonlinear functions,[42, 43] especially for univariate concave functions.[20, 44-46]

There are several different approaches to model the piecewise linear function for a concave term.[20, 43, 47] In this work, we use the well-known "multiple-choice" formulation[20] to approximate the square root term $\sqrt{x}$. Let $P = \{1, 2, 3, \cdots, p\}$ denote the set of intervals in the piecewise linear function $\varphi(x)$, and M_1, M_2,... M_p, M_{p+1} be the upper and lower bounds of *x* for each interval *p*. The "multiple choice" formulation of $\varphi(x)$ is then given by,

$$\varphi(x) = \min \quad \sum_p (F_p v_p + C_p u_p)$$

$$\text{s.t.} \qquad \sum_p v_p = 1$$

$$\sum_p u_p = x$$

$$M_p v_p \le u_p \le M_{p+1} v_p,\ \ p \in P$$

$$v_p \in \{0,1\},\ \ p \in P$$

where $C_p = \dfrac{\sqrt{M_{p+1}} - \sqrt{M_p}}{M_{p+1} - M_p}$ and $F_p = \sqrt{M_p} - C_p M_p,\ \ p \in P$.

[Figure 13]

As can be seen in Figure 13, the more intervals are used, the better is the approximation of the nonlinear function, but more additional variables and constraints are required. Using this "multiple choice" formulation to approximate the univariate concave terms $\sqrt{NZV_j}$ and $\sqrt{L_k}$ in the objective function of problem (P1), yields the following piecewise linear MILP problem (P3), which provides a tighter lower bounding problem of (P1).

(P3):

$$\begin{aligned}\text{Min: } & \sum_{j\in J} f_j Y_j + \sum_{i\in I}\sum_{j\in J}\sum_{k\in K} A_{ijk} XZ_{ijk} + \sum_{j\in J}\sum_{k\in K} B_{jk} Z_{jk} \\ & + \sum_{j\in J} q1_j \sum_{p_1}\left(F1_{j,p_1} v1_{j,p_1} + C1_{j,p_1} u1_{j,p_1}\right) + \sum_{k\in K} q2_k \sum_{p_2}\left(F2_{k,p_2} v2_{k,p_2} + C2_{k,p_2} u2_{k,p_2}\right)\end{aligned} \tag{35}$$

s.t. $$\sum_{p_1} v1_{j,p_1} = 1, \qquad \forall j \tag{36.1}$$

$$\sum_{p_1} u1_{j,p_1} = NZV_j, \qquad \forall j \tag{36.2}$$

$$M1_{j,p_1} v1_{j,p_1} \le u1_{j,p_1} \le M1_{j,p_1+1} v1_{j,p_1}, \qquad \forall j, p_1 \tag{36.3}$$

$$v1_{j,p_1} \in \{0,1\},\ \ u1_{j,p_1} \ge 0, \qquad \forall j, p_1 \tag{36.4}$$

$$\sum_{p_2} v2_{k,p_2} = 1, \qquad \forall k \tag{37.1}$$

$$\sum_{p_2} u2_{k,p_2} = L_k, \qquad \forall k \tag{37.2}$$

$$M2_{k,p_2} v2_{k,p_2} \le u2_{k,p_2} \le M2_{k,p_2+1} v2_{k,p_2}, \qquad \forall k, p_2 \tag{37.3}$$

$$v2_{k,p_2} \in \{0,1\},\ \ u2_{k,p_2} \ge 0, \qquad \forall k, p_2 \tag{37.4}$$

All the constraints of (P1).

Note that any feasible solution obtained from the problem (P3) is also a feasible solution of (P1), and for each feasible solution the objective function value of (P3) is always less than or equal to the objective function value of (P1).

In summary, the solution obtained by solving (P3) can provide a "good" initial point of solving problem (P1). On the other hand, by substituting the solution of (P3) into the objective function of problem (P1), i.e. function evaluation, or using that solution as an initial point to solve (P1) with a nonlinear programming (NLP) solver with fixed integer variables that are solution of (P3) or MINLP solver. In addition, we can obtain a valid upper bound of the "global" optimal objective function value of (P1).

In order to obtain an initial point "close" enough to the global solution, we can in principle use piecewise linear functions with sufficient large number of intervals to approximate the univariate concave terms in (P1). However, it is a nontrivial task to solve large-scale instances of (P3). To solve the problem effectively, we first exploit some properties of problem (P1).

6.2. Alternative Formulation

Let us first consider an alternative model formulation (AP) of problem (P1), in which (23)–(25) are excluded and represent the inequality (13) by a disaggregated disjunction (39),

$$\text{Min:} \sum_{j\in J} f_j Y_j + \sum_{i\in I}\sum_{j\in J}\sum_{k\in K} A_{ijk} XZ_{ijk} + \sum_{j\in J}\sum_{k\in K} B_{jk} Z_{jk} + \sum_{j\in J} q1_j \sqrt{NZV_j} + \sum_{j\in J}\sum_{k\in K} q2_k \sqrt{L_{jk}} \qquad (38)$$

$$\text{s.t.} \quad \sum_{i\in I} X_{ij} = Y_j, \qquad \forall j \qquad (9)$$

$$\sum_{j\in J} Z_{jk} = 1, \qquad \forall k \qquad (10)$$

$$Z_{jk} \le Y_j, \qquad \forall j,k \qquad (11)$$

$$N_j \ge \sum_{i\in I} \bar{S}_{ij} \cdot X_{ij} - S_j, \qquad \forall j \qquad (18)$$

$$\begin{bmatrix} Z_{jk} \\ L_{jk} \ge S_j + t_{jk} - R_k \end{bmatrix} \vee \begin{bmatrix} \neg Z_{jk} \\ L_{jk} \le 0 \end{bmatrix}, \quad \forall j,k \qquad (39)$$

$$XZ_{ijk} \le X_{ij}, \qquad \forall i,j,k \qquad (19)$$

$$XZ_{ijk} \le Z_{jk}, \qquad \forall i,j,k \qquad (20)$$

$$XZ_{ijk} \ge X_{ij} + Z_{jk} - 1, \qquad \forall i,j,k \qquad (21)$$

$$NZ_{jk} + NZ1_{jk} = N_j, \qquad \forall j,k \qquad (27)$$

$$NZ_{jk} \le Z_{jk} \cdot N_j^U, \qquad \forall j,k \qquad (28)$$

$$NZ1_{jk} \le (1 - Z_{jk}) \cdot N_j^U, \qquad \forall j,k \qquad (29)$$

$$NZV_j = \sum_{k \in K} \sigma_k^2 \cdot NZ_{jk} , \qquad \forall j \qquad (30)$$

$$X_{ij}, Y_j, Z_{jk} \in \{0,1\} , \qquad \forall i, j, k \qquad (14)$$

$$S_j \geq 0 , \ N_j \geq 0 , \qquad \forall j \qquad (15)$$

$$L_{jk} \geq 0 , \qquad \forall j, k \qquad (40)$$

$$XZ_{ijk} \geq 0 , \qquad \forall i, j, k \qquad (22)$$

$$NZV_j \geq 0 , \qquad \forall j \qquad (31)$$

Note that the set of disjunction in (39) is written in terms of the disaggregated variable L_{jk}, which is then substituted in the last term of the objective function (38). Applying the convex hull reformulation[48, 49] to the above disjunctive constraint (39) in (AP) leads to:

$$L_{jk} \leq Z_{jk} \cdot L_{jk}^U , \qquad \forall j, k \qquad (41.1)$$

$$L_{jk} \geq S1_{jk} + t_{jk} \cdot Z_{jk} - R_k \cdot Z_{jk} , \ \forall j, k \qquad (41.2)$$

$$S_j = S1_{jk} + S2_{jk} , \qquad \forall j, k \qquad (41.3)$$

$$S1_{jk} \leq Z_{jk} \cdot S_j^U , \qquad \forall j, k \qquad (41.4)$$

$$S2_{jk} \leq (1 - Z_{jk}) \cdot S_j^U , \qquad \forall j, k \qquad (41.5)$$

$$S1_{jk} \geq 0 , \ S2_{jk} \geq 0 , \qquad \forall j, k \qquad (41.6)$$

where $S1_{jk}$ and $S2_{jk}$ are two new auxiliary variables.

The value of the new continuous variable L_{jk} in (AP) is defined through two new constraints (39) and (40). It means that if customer demand zone *k* is assigned to DC *j*, i.e., $Z_{jk} = 1$, then the new variable L_{jk} represents the net lead time of customer demand zone *k*, otherwise $L_{jk} = 0$. Thus, an important proposition is that any feasible solution of problem (AP) is also a feasible solution of problem (P1), and vice versa.

Proposition 1. *If (* $X_{ij}^*, Y_j^*, Z_{jk}^*, S_j^*, N_j^*, L_k^*$ *) is a feasible solution of problem (P1) with objective function value* W^**, then (* $X_{ij}^*, Y_j^*, Z_{jk}^*, S_j^*, N_j^*, L_{jk}^*$ *), where* $L_{jk}^* = 0$ *if* $Z_{jk}^* = 0$ *and* $L_{jk}^* = L_k^*$ *if* $Z_{jk}^* = 1$*, is a feasible solution of problem (AP) and the associated objective function value of (AP) is* W^**. If (* $X_{ij}^*, Y_j^*, Z_{jk}^*, S_j^*, N_j^*, L_{jk}^*$ *) is a feasible solution of problem (AP) with objective* *function value* W^**, then (* $X_{ij}^*, Y_j^*, Z_{jk}^*, S_j^*, N_j^*, L_k^*$*), where* $L_k^* = \sum_{j \in J} L_{jk}^*$ *, is a feasible solution of problem (P1) and the associated objective function value of (P1) is* W^*

The proof of this proposition is given in the appendix. Proposition 1 shows the equivalence between (P1) and (AP). We can solve one of these two problems to obtain a feasible or optimal solution and then obtain the solution of another problem with the same objective function value after some algebraic derivations. Compared to model (P1), model (AP) has more square root terms in the objective function due to the introduction of variable L_{jk}, although the linearization constraints (23), (24), (25) for the product of S_j and Z_{jk} are not included in (AP).

It might not be a wise idea to solve (AP) with direct solution approach to obtain an optimal solution of (P1) since there are more nonlinear terms in (AP). However, (AP) can be decomposed into DCs with which a Lagrangean relaxation algorithm as will be shown in the next section can be used to speed up the computation. Further, because of the equivalence of (P1) and (AP), we can solve (P1) instead of (AP) in the full or reduced variable space during the solution procedure of the Lagrangean algorithm. In the next section, we describe how to incorporate model (AP) and the piecewise linear approximation into a Lagrangean relaxation algorithm.

6.3. Lagrangean Relaxation Algorithm

In order to obtain near global optimal solutions to problems (P1) and (AP) with modest computational effort, we propose a decomposition algorithm based on Lagrangean relaxation.

6.3.1. The Decomposition Procedure

In this solution procedure, we use a decomposition scheme by dualizing the assignment constraints (10) in (AP) using the Lagrangean multipliers λ_k. As a result, we obtain the following relaxed problem (APL),

$$W = \text{Min: } \sum_{j \in J} \left(f_j Y_j + \sum_{i \in I} \sum_{k \in K} A_{ijk} X Z_{ijk} + \sum_{k \in K} \left(B_{jk} - \lambda_k \right) Z_{jk} + q1_j \sqrt{NZV_j} + \sum_{k \in K} q2_k \sqrt{L_{jk}} \right) + \sum_{k \in K} \lambda_k \tag{42}$$

$$\text{s.t. } \sum_{i \in I} X_{ij} = Y_j, \qquad \forall j \tag{9}$$

$$Z_{jk} \le Y_j, \qquad \forall j,k \tag{11}$$

$$N_j \ge \sum_{i\in I} \bar{S}_{ij} \cdot X_{ij} - S_j, \qquad \forall j \tag{18}$$

$$XZ_{ijk} \le X_{ij}, \qquad \forall i,j,k \tag{19}$$

$$XZ_{ijk} \le Z_{jk}, \qquad \forall i,j,k \tag{20}$$

$$XZ_{ijk} \ge X_{ij} + Z_{jk} - 1, \qquad \forall i,j,k \tag{21}$$

$$NZ_{jk} + NZ1_{jk} = N_j, \qquad \forall j,k \tag{27}$$

$$NZ_{jk} \le Z_{jk} \cdot N_j^U, \qquad \forall j,k \tag{28}$$

$$NZ1_{jk} \le (1 - Z_{jk}) \cdot N_j^U, \qquad \forall j,k \tag{29}$$

$$NZV_j = \sum_{k\in K} \sigma_k^2 \cdot NZ_{jk}, \qquad \forall j \tag{30}$$

$$L_{jk} \le Z_{jk} \cdot L_{jk}^U, \qquad \forall j,k \tag{41.1}$$

$$L_{jk} \ge S1_{jk} + t_{jk} \cdot Z_{jk} - R_k \cdot Z_{jk}, \ \forall j,k \tag{41.2}$$

$$S_j = S1_{jk} + S2_{jk}, \qquad \forall j,k \tag{41.3}$$

$$S1_{jk} \le Z_{jk} \cdot S_j^U, \qquad \forall j,k \tag{41.4}$$

$$S2_{jk} \le (1 - Z_{jk}) \cdot S_j^U, \qquad \forall j,k \tag{41.5}$$

$$S1_{jk} \ge 0, \ S2_{jk} \ge 0, \qquad \forall j,k \tag{41.6}$$

$$X_{ij}, Y_j, Z_{jk} \in \{0,1\}, \qquad \forall i,j,k \tag{14}$$

$$S_j \ge 0, \ N_j \ge 0, \qquad \forall j \tag{15}$$

$$L_{jk} \ge 0, \qquad \forall j,k \tag{16}$$

$$XZ_{ijk} \ge 0, \qquad \forall i,j,k \tag{22}$$

$$NZV_j \ge 0, \qquad \forall j \tag{31}$$

where W is the objective function value. Note that (**P(λ)**) can be decomposed into $|J|$ subproblems, one for each candidate DC site $j \in J$, where each one is denoted by (APL$_j$) and is shown for a specific subproblem for candidate DC site j as follows:

$$W_j = \text{Min: } f_j Y_j + \sum_{i\in I}\sum_{k\in K} A_{ijk} XZ_{ijk} + \sum_{k\in K}\left(B_{jk} - \lambda_k\right) Z_{jk} + q1_j \sqrt{NZV_j} + \sum_{k\in K} q2_k \sqrt{L_{jk}} \tag{43}$$

s.t. $\sum_{i\in I} X_{ij} = Y_j$,

$$Z_{jk} \le Y_j, \qquad \forall k$$

$$N_j \ge \sum_{i \in I} \overline{S}_{ij} \cdot X_{ij} - S_j,$$

$$XZ_{ijk} \le X_{ij}, \qquad \forall i,k$$

$$XZ_{ijk} \le Z_{jk}, \qquad \forall i,k$$

$$XZ_{ijk} \ge X_{ij} + Z_{jk} - 1, \qquad \forall i,k$$

$$NZ_{jk} + NZ1_{jk} = N_j, \qquad \forall k$$

$$NZ_{jk} \le Z_{jk} \cdot N_j^U, \qquad \forall k$$

$$NZ1_{jk} \le (1 - Z_{jk}) \cdot N_j^U, \qquad \forall k$$

$$NZV_j = \sum_{k \in K} \sigma_k^2 \cdot NZ_{jk},$$

$$L_{jk} \le Z_{jk} \cdot L_{jk}^U, \qquad \forall k$$

$$L_{jk} \ge S1_{jk} + t_{jk} \cdot Z_{jk} - R_k \cdot Z_{jk}, \quad \forall k$$

$$S_j = S1_{jk} + S2_{jk}, \qquad \forall k$$

$$S1_{jk} \le Z_{jk} \cdot S_j^U, \qquad \forall k$$

$$S2_{jk} \le (1 - Z_{jk}) \cdot S_j^U, \qquad \forall k$$

$$S1_{jk} \ge 0, \;\; S2_{jk} \ge 0, \qquad \forall k$$

$$X_{ij}, Y_j, Z_{jk} \in \{0,1\}, \qquad \forall i,k$$

$$S_j \ge 0, \;\; N_j \ge 0,$$

$$L_{jk} \ge 0, \qquad \forall k$$

$$XZ_{ijk} \ge 0, \qquad \forall i,k$$

$$NZV_j \ge 0,$$

Hence, (APL) can be decomposed into $|J|$ subproblems (APL$_j$), and one for each candidate DC site $j \in J$. Let W_j denote the globally optimal objective function value of problem (APL$_j$). As a result of the decomposition procedure, the global minimum of (APL), which corresponds to a lower bound of the global optimum of problem (AP), can be calculated by:

$$W = \sum_{j\in J} W_j + \sum_{k\in K} \lambda_k \ . \qquad (44)$$

For each fixed value of the Lagrangean multipliers λ_k, we solve problem (APL$_j$) for each candidate DC location j. Then, based on (44), the optimal objective function value of problem (APL) can be calculated for each fixed value of λ_k. Using a standard subgradient method[50, 51] to update the Lagrangean multiplier λ_k, the algorithm iterates until a preset optimality tolerance is reached.

6.3.2. Lagrangean Relaxation Subproblems

In each iteration with fixed values of the Lagrangean multipliers λ_k, the binary variables for installing a DC (Y_j) are optimized separately in each subproblem (APL$_j$) in the aforementioned decomposition procedure. For each subproblem (APL$_j$), we can observe that the objective function value of (APL$_j$) is 0 if and only if $Y_j = 0$ (i.e. we do not install DC j). In other words, there is a feasible solution that leads to the objective of subproblem (APL$_j$) equal to 0. Therefore, the global minimum of subproblem (APL$_j$) should be less than or equal to zero. Given this observation, it is possible that for some value of λ_k (such as $\lambda_k = 0$, $k \in K$) the optimal objective function values for all the subproblem (APL$_j$) are 0 (i.e., $Y_j = 0$, $j \in J$, we do not select any DC). However, the original assignment constraint (8) implies that at least one DC should be selected to meet the demands, i.e.

$$\sum_{j\in J} Y_j \geq 1 \, . \qquad (45)$$

Once constraint (10) is relaxed, constraint (45) becomes "not redundant" and should be taken into account in the algorithm.[30, 52] To satisfy the constraint (45) in the Lagrangean relaxation procedure, we make the following modifications to the aforementioned step of solving problem (APL$_j$) for each candidate DC location j.

First, consider the problem (APLR$_j$), which is actually a special case of (APL$_j$) when $Y_j = 1$. The formulation for a specific *j* is given as:

$$\overset{\square}{W}_j = \text{Min: } f_j + \sum_{i\in I}\sum_{k\in K} A_{ijk} XZ_{ijk} + \sum_{k\in K}\left(B_{jk} - \lambda_k\right) Z_{jk} + q1_j \sqrt{NZV_j} + \sum_{k\in K} q2_k \sqrt{L_{jk}} \qquad (46)$$

s.t. the same constraints as (APL$_j$) with $Y_j = 1$.

where $\overset{\square}{W}_j$ is denoted as the optimal objective function value of the problem (APLR$_j$).

With a similar scheme as introduced in our previous work,[30] we use the following solution procedure to deal with the implied constraint (45): for each fixed value of λ_k, we solve (APLR$_j$) for every candidate DC location j. Then select the DCs in candidate location j (i.e. let $Y_j = 1$), for which $\overset{\square}{W}_j \leq 0$. For all the remaining DCs for which $\overset{\square}{W}_j > 0$, we do not select them and set $Y_j = 0$. If all the $\overset{\square}{W}_j > 0$, $\forall j \in J$, we select only one DC with the minimum $\overset{\square}{W}_j$, i.e. $Y_{j*} = 1$ for the $j*$ such that $\overset{\square}{W}_{j*} = \min_{j \in J}\{\overset{\square}{W}_j\}$. By doing this at each iteration of the Lagrangean relaxation (for each value of the multiplier λ_i), we ensure that the optimal solution always satisfies $\sum_{j \in J} Y_j \geq 1$. Thus, the globally optimal objective function of (APL$_j$) can be recalculated as:

$$W = \sum_{j \in J, Y_j = 1} \overset{\square}{W}_j + \sum_{k \in K} \lambda_k \quad . \tag{57}$$

Since (APLR$_j$) includes $|k| + 1$ univariate concave terms in the objective function solving it to global optimality might be computationally expensive if |*k*| is large. To improve the computational efficiency, piecewise linear approximations can be used. Similarly as in (P3), we use the "multiple choice" formulation to approximate the concave terms in (APLR$_j$), and obtain the following MILP model (APLP$_j$),

$$\begin{aligned} \overline{W}_j = \text{Min: } & f_j + \sum_{i \in I} \sum_{k \in K} A_{ijk} XZ_{ijk} + q1_j \sum_{p_1} \left(F1_{j,p_1} v1_{j,p_1} + C1_{j,p_1} u1_{j,p_1} \right) \\ & + \sum_{k \in K} \left(B_{jk} - \lambda_k \right) Z_{jk} + \sum_{k \in K} q2_k \sum_{p3} \left(F3_{j,k,p_3} v3_{j,k,p_3} + C3_{j,k,p_3} u3_{j,k,p_3} \right) \end{aligned} \tag{48}$$

$$\text{s.t. } \sum_{p_1} v1_{j,p_1} = 1, \tag{36.1}$$

$$\sum_{p_1} u1_{j,p_1} = NZV_j, \tag{36.2}$$

$$M1_{j,p_1} v1_{j,p_1} \leq u1_{j,p_1} \leq M1_{j,p_1+1} v1_{j,p_1}, \qquad \forall p_1 \tag{36.3}$$

$$v1_{j,p_1} \in \{0,1\}, \quad u1_{j,p_1} \geq 0, \qquad \forall p_1 \tag{36.4}$$

$$\sum_{p_3} v3_{j,k,p_3} = 1, \qquad \forall k \tag{49.1}$$

$$\sum_{p_3} u3_{j,k,p_3} = L_{jk}, \qquad \forall k \tag{49.2}$$

$$M3_{j,k,p_3} v3_{j,k,p_3} \leq u3_{j,k,p_3} \leq M3_{j,k,p_3+1} v3_{j,k,p_3}, \qquad \forall k, p_3 \tag{49.3}$$

$$v3_{j,k,p_3} \in \{0,1\}, \quad u3_{j,k,p_3} \geq 0, \quad \forall k, p_3 \tag{49.4}$$

All the constraints of (APLR$_j$).

where $\overline{W}_j$ is the optimal objective function value of problem (APLP$_j$).

As discussed in Section 6.1., $\overline{W}_j$ provides a valid "global" lower bound of $\overset{\square}{W}_j$. Thus, the globally optimal objective function value of problem (AP) has a globally lower bound $\overline{W}$, which can be calculated by:

$$\overline{W} = \sum_{j \in J, Y_j = 1} \overline{W}_j + \sum_{k \in K} \lambda_k \ . \tag{50}$$

Based on this result, instead of solving each (APLR$_j$) to globally optimality, we can use the optimal solution of (APLP$_j$) as an initial point to solve (APLR$_j$) with an MINLP solver that relies on convexity assumptions, such as DICOPT, SBB, etc., so as to improve the computational efficiency. In this way, the optimal objective function of (APL$_j$), calculated by $W = \sum_{j \in J, Y_j = 1} \overset{\square}{W}_j + \sum_{k \in K} \lambda_k$, is no longer a valid global lower bound of the optimal objective function value of problem (AP). However, the lower bound $\overline{W}$, obtained from piecewise linear lower bounding problem (APLP$_j$), still provides a "globally" lower bound of the optimal objective function value of problem (AP).

6.3.3. Upper Bounding

A feasible solution of problem (AP) naturally provides a valid upper bound of its global optimal objective function value. To obtain a feasible solution, a common approach is to fix the values of the binary variables (X_{ij} , Y_j and Z_{jk}) and solve (AP) in the reduced variable space with an NLP solver. However, solving a large-scale NLP problem could be computationally expensive.

Based on Proposition 1, to obtain a valid upper bound, we can solve problem (P1) instead of (AP) with fixed values of binary variables (X_{ij} , Y_j and Z_{jk}) to reduce the computational effort, because (P1) has fewer nonlinear terms than (AP). To avoid solving the large scale NLP problem, we first solve the piecewise linear lower bounding MILP problem (P3) with fixed binary variables, and then substitute the optimal solution into the objective function of problem (P1) to calculate the associated objective function value. As discussed in Section 6.1., the optimal solution of (P3) is a feasible solution of (P1). Thus, the objective function value of (P1) obtained by function evaluation yields an upper bound of the global minimum of problem (AP). By using this approach, we avoid using an NLP solver and improve the computational efficiency and robustness without sacrificing the solution quality.

6.3.4. The Solution Algorithm

To summarize, the solution algorithm is as follows:

Step 1: (Initialization)

Use an arbitrary guess as the initial vector of Lagrange multipliers λ^1, or else the dual of constraint (10) of a local optimum of the NLP relaxation of model (P1). Let the incumbent upper bound be $UB = +\infty$, incumbent lower bound be $LB = -\infty$, global lower bound be $GLB = -\infty$, and iteration number be $t = 1$. Set the step length parameter $\theta = 2$.

Step 2:

With fixed Lagrange multipliers λ^t, first solve the piecewise linear lower bounding problem of the modified Lagrangean relaxation subproblem (APLP$_j$) with an MILP solver. Then fix the value of binary variables X_{ij} and Z_{jk}, use the optimal solution as the initial point and solve the modified Lagrangean relaxation subproblem (APLR$_j$) with an NLP solver (not necessary global optimizer), or calculate the objective function value of (APLR$_j$) directly using the optimal solution of (APLP$_j$).

Denote the optimal objective function value of (APLP$_j$) as $\overline{W}_j^{(\lambda^t)}$, the optimal objective function value of (APLR$_j$) as $\overset{\square}{W}_j^{(\lambda^t)}$, and the optimal solutions of (APLR$_j$) as ($\overset{\square}{X}_{ij}^{(\lambda_t)}, \overset{\square}{Z}_{jk}^{(\lambda_t)}$).

If all $\overset{\square}{W}_j^{(\lambda^t)} > 0$, $\forall j \in J$, let $Y_{j^*}^{(\lambda_t)} = 1$, $X_{ij^*}^{(\lambda_t)} = \overset{\square}{X}_{ij^*}^{(\lambda_t)}$, for *j** with $\overset{\square}{W}_{j^*}^{(\lambda^t)} = \min_{j \in J} \left\{ \overset{\square}{W}_j^{(\lambda^t)} \right\}$.

Else, let $Y_j^{(\lambda_t)} = 1$, $X_{ij}^{(\lambda_t)} = \overset{\square}{X}_{ij}^{(\lambda_t)}$ for all j with $\overset{\square}{W}_j^{(\lambda^t)} \le 0$, and $Y_j^{(\lambda_t)} = 0$, $X_{ij}^{(\lambda_t)} = 0$ for all j such that $\overset{\square}{W}_j^{(\lambda^t)} > 0$.

Calculate $W^{(\lambda^t)} = \sum_{j \in J, Y_j^{(\lambda_t)} = 1} \overset{\square}{W}_j + \sum_{k \in K} \lambda_k$, $\overline{W}^{(\lambda^t)} = \sum_{j \in J, Y_j^{(\lambda_t)} = 1} \overline{W}_j + \sum_{k \in K} \lambda_k$.

If $W^{(\lambda^t)} > LB$, update lower bound by setting $LB = W^{(\lambda^t)}$. If more than 2 iterations of the subgradient procedure[50] are performed without an increment of LB, then halve the step length parameter by setting $\theta = \dfrac{\theta}{2}$.

If $\overline{W}(\lambda^t) > GLB$, update lower bound by setting $GLB = \overline{W}(\lambda^t)$.

Note that the strictly global lower bound is given by GLB instead of LB, if (APLR$_j$) is not solved with a global optimizer.

Step 3:

Fixing the design variable values as $X_{ij} = X_{ij}^{(\lambda_t)}$, $Y_j = Y_j^{(\lambda_t)}$ and using $\overset{\square}{Z}_{jk}^{(\lambda_t)}$ as the initial values of the binary variables Z_{jk}, solve the piecewise linear lower bounding problem (P3) in the reduced space with fixed X_{ij}, Y_j and λ^t using an MILP solver. Denote the optimal solution as ($Z_{jk}^{(\lambda_t)}$, $S_j^{(\lambda_t)}$, $N_j^{(\lambda_t)}$, $L_k^{(\lambda_t)}$). Substitute the optimal solution ($X_{ij}^{(\lambda_t)}, Y_j^{(\lambda_t)}, Z_{jk}^{(\lambda_t)}, S_j^{(\lambda_t)}, N_j^{(\lambda_t)}, L_k^{(\lambda_t)}$) into problem (P1) and calculate its objective function value, which is denoted as $\overset{\square}{W}^{(\lambda^t)}$.

If $\overset{\square}{W}^{(\lambda^t)} < UB$, update the upper bound by setting $UB = \overset{\square}{W}^{(\lambda^t)}$.

Step 4:

Calculate the subgradient (G_k) using $G_k{}^t = 1 - \sum_{j \in J} \overset{\square}{Z}_{jk}^{(\lambda_t)}$, $\forall k$.

Compute the step size T,[50, 51] $T^t = \dfrac{\theta \cdot (UB - LB)}{\sum_{k \in K} (G_k{}^t)^2}$.

Update the multipliers, $\lambda^{t+1} = \lambda^t + T^t \cdot G^t$.

Step 5:

If $gap = \dfrac{UB - LB}{UB} < tol$ (e.g. 10^{-3}), or $\left\| \lambda^{t+1} - \lambda^t \right\|^2 < tol$ (e.g. 10^{-2}) or the maximum number of iterations has been reached, set UB as the optimal objective function value, and set ($X_{ij}^{(\lambda_t)}, Y_j^{(\lambda_t)}, Z_{jk}^{(\lambda_t)}, S_j^{(\lambda_t)}, N_j^{(\lambda_t)}, L_k^{(\lambda_t)}$) as the optimal solution.

Else, increment t as $t+1$, go to Step 2.

We should note that the entire procedure requires at least an MILP solver. An NLP solver can be used to solve problem (APLR$_j$) in the reduced variable space in Step 2, but the NLP solver is not required. The reason is that the solution of the nonlinear optimization problem (APLR$_j$), which reduces to an NLP from MINLP after X_{ij} and Z_{jk} are fixed, can be substituted by simple function evaluation as stated in Step 2, although the solution quality may be sacrificed. Also, the above algorithm is guaranteed to provide rigorous global lower bounds by GLB as discussed in Step 2. Thus, the gap for global optimum is given by $Ggap = (UB - GLB) / GLB$. Due to the duality gap, this algorithm stops after a finite number of iterations. As will be shown in the computational results, the global gaps are quite small.

7. Computational Results

In order to illustrate the application of the proposed solution strategies, we present computational experiments for five medium and large scale instances on an IBM T40 laptop with Intel 1.50GHz CPU and 512 MB RAM. The proposed solution procedure is coded in GAMS 22.8.1. The MILP problems are solved using CPLEX 11.0.1, the NLP problems in Step 2 of the Lagrangean solution approach are solved with solver CONOPT 3.3, and the global optimizer used in the computational experiments is BARON 8.1.4.

7.1. Input Parameters

Since we consider large size problems, most of the input data are generated randomly. The safety stock factors for DCs ($\lambda 1_j$) and customers ($\lambda 2_k$) are the same and equal to 1.96, which corresponds to 97.5% service level is demand is normally distrubted. We consider 365 days in a year (χ). The guaranteed service time of the last echelon customer demand zones (R_k) are set to 0. The annual fixed costs ($/year) to install the DCs (f_j) are generated uniformly on *U*[150,000, 160,000] and the variable cost coefficient (g_j, $/ ton · year) are generated uniformly on *U*[0.01, 0.1]. The guaranteed service times of the plants (SI_i, days) are set as integers uniformly on *U*[1, 5]. The order processing time ($t1_{ij}$, days) between plants and DCs are generated as integers uniformly on *U*[1, 7], and the order processing time ($t2_{jk}$, days) between DCs and customer demand zones are generated as integers uniformly on *U*[1, 3]. The unit transportation cost from plants to DCs ($c1_{ij}$, $/ton) and from DCs to customer demand zones ($c2_{jk}$, $/ton) are set to

$$c1_{ij} = t1_{ij} \times U[0.05,\ 0.1]$$

$$c2_{jk} = t2_{jk} \times U[0.05,\ 0.1]$$

The expected demand (μ_i ,ton/day) is generated uniformly on U[75, 150] and its standard deviation (σ_i , ton/day) is generated uniformly on U[0, 50]. The daily unit pipeline and safety stock inventory holding costs ($\theta 1_{ij} / \chi$, $\theta 2_{jk} / \chi$, $h1_j / \chi$ and $h2_k / \chi$) are generated uniformly on U[0.1, 1].

In this instance, the annual fixed DC installation cost (f_j) is \$50,000/year for all the DCs. The variable cost coefficient of installing a DC (g_j) at all the candidate locations is \$0.5/(ton · year). The safety stock factors for DCs ($\lambda 1_j$) and wholesalers ($\lambda 2_k$) are the same and equal to 1.96. We consider 365 days in a year (χ). The guaranteed service times of the three vendors (SI_i) are 3 days, 3 days and 4 days, respectively. The pipeline inventory holding cost is \$1/(ton · day) for all the DCs ($\theta 1_{ij} / \chi$) and wholesalers ($\theta 2_{jk} / \chi$), and the safety stock holding cost is \$1.5/(ton · day) for all the DCs ($h1_j / \chi$) and wholesalers ($h2_k / \chi$).

The number of intervals P_1, P_2, P_3 , which are required to approximate each of the univariate concave terms $\sqrt{NZV_j}$, $\sqrt{L_k}$ and $\sqrt{L_{jk}}$ in (P3) and (APLP$_j$), are all set to twenty. All the breakpoints of the piecewise linear function are evenly distributed between the lower and upper bound of the variables to be approximated. We perform the testing on 5 problem instances of the model by varying the parameter settings.

7.2. Performance

The problem sizes for the five instances we considered in this example are given in Table 11. The numbers in *i*, *j*, *k* columns stand for the number of plants, potential DCs and customer demand zones of each instance. The computational results of the five instances are given in Table 12. We solve each instance with three solution approaches. The first approach is solving the original MINLP problem (P0) directly with the global optimizer BARON. The second approach is to first solve the initialization MILP problem (P2) with at most one hour computational time, then use the best solution obtained from (P2) as the initial point to solve the MINLP problem (P1) with DICOPT and SBB solvers. The third approach is the Lagrangean relaxation algorithm as presented in Section 6.3.4. In the Lagrangean relaxation algorithm, we solve the NLP relaxation of model (P1) to obtain the initial vector of Lagrangean multipliers in Step 1, and solve the modified Lagrangean relaxation subproblem (APLR$_j$) with CONOPT 3.3 in Step 2. As we can see, global optimal solutions could not be obtained for 10 hours by solving these problems directly with BARON, or solving the reformulated problem (P1) with the initialization scheme. However, optimal solutions within 1% global optimality are obtained for the instances by using the proposed algorithm.

[Figure 14, (a), (b)]

Furthermore, we can see from Table 12 that only the medium size instance with 2 plants, 20 potential DCs and 20 customer demand zones, can be solved directly with BARON to obtain a feasible solution. For all the other instances, BARON fails to return any bounds or feasible solutions due to the large size of these problems. By solving the reformulated problem (P1) of the medium scale instance with convex MINLP solvers DICOPT and SBB after the initialization scheme, we can see that a better feasible solution can be obtained with smaller global optimality gap and much shorter computational times (140.3 and 163.6 CPU seconds, respectively). However, the proposed Lagrangean relaxation and decomposition algorithm can obtain much better solution ($1,776,969 vs. $1,889,577 and $1,820,174) within 0.06% of global optimality in much shorter time 175.0 CPU seconds. For the other large scale instances, BARON cannot finish preprocessing after 10 hours if solving the reformulated problem (P1). In contrast the proposed Lagrangean relaxation algorithm can solve all the instances with 1% optimality in less than 10 hours of computational time as shown in Table 12. Therefore, we can see the significant advantage of using the proposed Lagrangean relaxation algorithm for solving medium and large scale instances.

The change of bounds during the Lagrangean relaxation and decomposition algorithm for a large scale instance with 3 plants, 50 potential DCs and 150 customer demand zones are given in Figure 14. We can see that as iterations proceed, the upper bound, which is the objective function value of a feasible solution, keeps decreasing, and both the incumbent lower bound and the global lower bound keep increasing, until the termination criterion is satisfied. We should note that the global optimality gap is given by the upper bound and the global lower bound instead of the incumbent lower bound. The reason is that the incumbent lower bound comes from the MILP subproblems ($APLR_j$), which are not solved with BARON but CPLEX , but the global lower bounds come from the piecewise linear lower bounding MILP problem ($APLP_j$), which is solved to global optimal using CPLEX, and thus provides a valid lower bound of the global optimal objective function value of the full problem.

[Table 11]
[Table 12]

8. Conclusion

In this paper, we present an MINLP model that determines the optimal network structure, transportation and inventory levels of a multi-echelon supply chain with the presence of customer demand uncertainty. The well-known guaranteed service approach is used to model the multi-echelon inventory system. The risk pooling effect is also taken into account in the model by consolidating the demands in the downstream nodes to their upstream nodes. Examples on supply chains for industrial gases and performance chemicals are presented to illustrate the applicability of the proposed model. To solve the resulting MINLP problem efficiently for large scale instances, a decomposition algorithm, based on Lagrangean relaxation and piecewise linear approximation was proposed. Computational experiments on large scale problems show that the proposed algorithm can obtain global or near-global optimal solutions (typically within 1% of the global optimum) in modest computational expense without the need of a global optimizer.

This research can be extended to address the responsiveness issue[53, 54] of supply chains and will be given in a further paper.

Acknowledgment

The authors acknowledge the financial supports from the National Science Foundation under Grant No. DMI-0556090 and No. OCI- 0750826, and Pennsylvania Infrastructure Technology Alliance (PITA). Fengqi You is grateful to Prof. S. Willems (Boston University) and Prof. S. C. Graves (MIT) for helpful discussions.

Appendix:

Proposition 1. *If ($X^*_{ij}, Y^*_j, Z^*_{jk}, S^*_j, N^*_j, L^*_k$) is a feasible solution of problem (P1) with objective function value W^*, then ($X^*_{ij}, Y^*_j, Z^*_{jk}, S^*_j, N^*_j, L^*_{jk}$), where $L^*_{jk} = 0$ if $Z^*_{jk} = 0$ and $L^*_{jk} = L^*_k$ if $Z^*_{jk} = 1$, is a feasible solution of problem (AP) and the associated objective function value of (AP) is W^*. If ($X^*_{ij}, Y^*_j, Z^*_{jk}, S^*_j, N^*_j, L^*_{jk}$) is a feasible solution of problem (AP) with objective function value W^*, then ($X^*_{ij}, Y^*_j, Z^*_{jk}, S^*_j, N^*_j, L^*_k$), where $L^*_k = \sum_{j \in J} L^*_{jk}$, is a feasible solution of problem (P1) and the associated objective function value of (P1) is W^**

Proof:

If ($X_{ij}^*, Y_j^*, Z_{jk}^*, S_j^*, N_j^*, L_k^*$) is a feasible solution of problem (P1), it is easy to see that this solution satisfies all the constraints of (AP) except (39) and (40). Constraints (39) and (40) imply that the variable $L_{jk} = 0$ if $Z_{jk} = 0$, and if $Z_{jk} = 1$, and that we should have $L_{jk} \geq S_j + t_{jk} - R_k$, which is the equivalent to constraint (32) for $Z_{jk} = 1$. Thus, by setting $L_{jk}^* = 0$ if $Z_{jk}^* = 0$ and $L_{jk}^* = L_k^*$ if $Z_{jk}^* = 1$, we have the solution ($X_{ij}^*, Y_j^*, Z_{jk}^*, S_j^*, N_j^*, L_{jk}^*$) that satisfies all the constraints of (AP), and thus it is a feasible solution of (AP).

Because $L_{jk}^* = 0$ if $Z_{jk}^* = 0$ and $L_{jk}^* = L_k^*$ if $Z_{jk}^* = 1$, and the following constraint included in both (P1) and (AP),

$$\sum_{j \in J} Z_{jk} = 1, \qquad \forall k \qquad (10)$$

which means for each *k*, there is only one $Z_{jk}^* = 1$. So we can have

$$\sqrt{L_k^*} = \sum_{j \in J} \sqrt{L_{jk}^*}, \quad \forall k \qquad (A1)$$

Because W^* is the objective function value of (P1) corresponding to the solution ($X_{ij}^*, Y_j^*, Z_{jk}^*, S_j^*, N_j^*, L_k^*$), we have

$$W^* = \sum_{j \in J} f_j Y_j^* + \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} A_{ijk} XZ_{ijk}^* + \sum_{j \in J} \sum_{k \in K} B_{jk} Z_{jk}^* + \sum_{j \in J} q1_j \sqrt{NZV_j^*} + \sum_{k \in K} q2_k \sqrt{L_k^*} \qquad (A2)$$

Given (A1) and (A2), we can have

$$W^* = \sum_{j \in J} f_j Y_j^* + \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} A_{ijk} XZ_{ijk}^* + \sum_{j \in J} \sum_{k \in K} B_{jk} Z_{jk}^* + \sum_{j \in J} q1_j \sqrt{NZV_j^*} + \sum_{j \in J} \sum_{k \in K} q2_k \sqrt{L_{jk}^*} \quad (A3)$$

which is the objective function value of (AP) corresponding to the solution ($X_{ij}^*, Y_j^*, Z_{jk}^*, S_j^*, N_j^*, L_{jk}^*$).

Similarly, it is easy to show that if ($X_{ij}^*, Y_j^*, Z_{jk}^*, S_j^*, N_j^*, L_{jk}^*$) is a solution of (AP), by setting $L_k^* = \sum_{j \in J} L_{jk}^*$, we can have ($X_{ij}^*, Y_j^*, Z_{jk}^*, S_j^*, N_j^*, L_k^*$) satisfying all the constraints of (P1), and thus it is a feasible solution of (P1). On the other hand, given (A1) and (A3), we can easily derive (A2). ■

Reference

(1) Chopra, S.; Meindl, P., *Supply Chain Management: Strategy, Planning and Operation*. Prentice Hall: Saddle River, NJ, 2003.

(2) Grossmann, I. E., Enterprise-wide Optimization: A New Frontier in Process Systems Engineering. *AIChE Journal* **2005,** 51, 1846.

(3) Zipkin, P. H., *Foundations of Inventory Management*. McGraw-Hill: Boston, MA, 2000.

(4) Simpson, K. F., In-process inventories. *Operations Research* **1958,** 6, 863.

(5) Clark, A.; Scarf, S., Optimal policies for a multi-echelon inventory problem. *Management Science* **1960,** 6, 475.

(6) Federgruen, A.; Zipkin, P. H., Computational Issues in an Infinite-Horizon, Multiechelon Inventory Model. *Operations Research* **1984,** 32, (4), 818.

(7) Lee, H. L.; Billington, C., Material management in decentralized supply chains. *Operations Research* **1993,** 41, (5), 835.

(8) Glasserman, P.; Tayur, S., Sensitivity analysis for base-stock levels in multiechelon production-inventory systems. *Management Science* **1995,** 41, (2), 263.

(9) Ettl, M.; Feigin, G. E.; Lin, G. Y.; Yao, D. D., A supply network model with base-stock control and service requirements. *Operations Research* **2000,** 49, (2), 216.

(10) Graves, S. C., Safety Stocks in Manufacturing Systems. *Journal of Manufacturing and Operations Management* **1988,** 1, 67.

(11) Inderfurth, K., Safety Stock Optimization in Multi-Stage Inventory Systems. *International Journal of Production Economics* **1991,** 24, 103.

(12) Inderfurth, K., Safety stock optimization in multi-stage inventory systems. *International Journal of Production Economics* **1993,** 24, 103.

(13) Inderfurth, K.; Minner, S., Safety stocks in multi-stage inventory systems under different service measures. *European Journal of Operations Research* **1998,** 106, 57.

(14) Graves, S. C.; Willems, S. P., Optimizing strategic safety stock placement in supply chains. *Manufacturing & Service Operations Management* **2000,** 2, (1), 68.

(15) Minner, S., Strategic safety stocks in reverse logistics supply chains. *International Journal of Production Economics* **2001,** 71, 417.

(16) Graves, S. C.; Willems, S. P., Supply chain design: safety stock placement and supply chain configuration. In *Handbooks in Operations Research and Management Science*, de Kok, A. G.; Graves, S. C., Eds. Elsevier: North-Holland, Amsterdam, 2003; Vol. 11, pp 95.

(17) Graves, S. C.; Willems, S. P., Optimizing the supply chain configuration for new products. *Management Science* **2005,** 51, (8), 1165.

(18) Humair, S.; Willems, S. P., Optimizing strategic safety stock placement in supply chains with clusters of commonality. *Operations Research* **2006,** 54, (4), 725.

(19) Bossert, J. M.; Willems, S. P., A periodic-review modeling approach for guaranteed service supply chains. *Interfaces* **2007,** 37, (5), 420.

(20) Magnanti, T. L.; Shen, Z.-J. M.; Shu, J.; Simchi-Levi, D.; Teo, C.-P., Inventory placement in acyclic supply chain networks. *Operations Research Letters* **2006,** 34, 228.

(21) Jung, J. Y.; Blau, G.; Pekny, J. F.; Reklaitis, G. V.; Eversdyk, D., Integrated safety stock management for multi-stage supply chains under production capacity constraints. In *Computers & Chemical Engineering*, Accepted: 2007.

(22) Bok, J.-K.; Grossmann, I. E.; Park, S., Supply Chain Optimization in Continuous Flexible Process Networks. *Industrial & Engineering Chemistry Research* **2000,** 39, (5), 1279.

(23) Jackson, J. R.; Grossmann, I. E., Temporal decomposition scheme for nonlinear multisite production

planning and distribution models. *Industrial & Engineering Chemistry Research* **2003,** 42, (13), 3045.

(24) Erlebacher, S. J.; Meller, R. D., The interaction of location and inventory in designing distribution systems. *IIE Transactions* **2000,** 32, 155.

(25) Daskin, M. S.; Coullard, C.; Shen, Z.-J. M., An inventory-location model: formulation, solution algorithm and computational results. *Annals of Operations Research* **2002,** 110, 83.

(26) Shen, Z.-J. M.; Coullard, C.; Daskin, M. S., A joint location-inventory model. *Transportation Science* **2003,** 37, 40.

(27) Ozsen, L.; Daskin, M. S.; Coullard, C. R., Capacitated warehouse location model with risk pooling. In *Naval Research Logistics*, Accepted: 2006.

(28) Ozsen, L.; Daskin, M. S.; Coullard, C. R., Location-inventory planning with risk pooling and multi-sourcing. In *Transportation Science*, Submitted: 2006.

(29) Sourirajan, K.; Ozsen, L.; Uzsoy, R., A single-product network design model with lead time and safety stock considerations. *IIE Transactions* **2007,** 39, 411.

(30) You, F.; Grossmann, I. E., Mixed-Integer Nonlinear Programming Models and Algorithms for Large-Scale Supply Chain Design with Stochastic Inventory Management. *Industrial & Engineering Chemistry Research* **2008,** 47, (20), 7802.

(31) Eppen, G., Effects of centralization on expected costs in a multi-echelon newsboy problem. *Management Science* **1979,** 25, (5), 498.

(32) Rao, U.; Scheller-Wolf, A.; Tayur, S., Development of A Rapid-Response Supply Chain at Caterpillar. *Operations Research* **1999,** 48, (2), 189.

(33) Little, J. D. C., A Proof of the Queueing Formula L = λ W. *Operations Research* **1961,** 9, 383.

(34) http://www.effectiveinventory.com.

(35) Campbell, A. M.; Clarke, L. W.; Savelsbergh, M. W. P., Inventory Routing in Practice. In *The Vehicle Routing Problem*, Toth, P.; Vigo, D., Eds. SIAM: Philadelphia, PA, USA, 2001; pp 309.

(36) Savelsbergh, M.; Song, J.-H., Inventory routing with continuous moves *Computers & Operations Research* **2007,** 34, (6), 1744.

(37) Hübner, R., *Strategic Supply Chain Management in Process Industries: An Application to Specialty Chemicals Production Network Design*. Springer-Verlag: Heidelberg, 2007.

(38) Glover, F., Improved Linear Integer Programming Formulations of Nonlinear Integer Problems. *Management Science* **1975,** 22, (4), 455.

(39) Falk, J. E.; Soland, R. M., An Algorithm for separable nonconvex programming problems. *Management Science* **1969,** 15, 550.

(40) Sahinidis, N. V., BARON: A general purpose global optimization software package. *Journal of Global Optimization* **1996,** 8, (2), 201.

(41) Brooke, A.; Kendrick, D.; Meeraus, A.; Raman, R., GAMS- A User's Manual. In GAMS Development Corp.: 1998.

(42) Croxton, K. L.; Gendron, B.; Magnanti, T. L., A Comparison of Mixed-Integer Programming Models for Nonconvex Piecewise Linear Cost Minimization Problems. *Management Science* **2003,** 49, (9), 1268.

(43) Wicaksono, D. S.; Karimi, I. A., Piecewise MILP under- and overestimators for global optimization of bilinear programs. *AIChE Journal* **2008,** 54, (4), 991

(44) Bergamini, M. L.; Grossmann, I. E.; Scenna, N.; Aguirre, P., An improved piecewise outer-approximation algorithm for the global optimization of MINLP models involving concave and bilinear terms. *Computers & Chemical Engineering* **2008,** 32, 477.

(45) Lin, X.; Floudas, C. A.; Kallrath, J., Global solution approach for a nonconvex MINLP problem in product portfolio optimization. *Journal of Global Optimization* **2005,** 32, 417.

(46) Vidyarthi, N.; Celebi, E.; Elhedhli, S.; Jewkes, E., Integrated Production-Inventory-Distribution System Design with Risk Pooling: Model Formulation and Heuristic Solution. *Transportation Science* **2007,** 41, (3), 392.

(47) Padberg, M. W., Approximating separable nonlinear functions via mixed zero-one programs. *Operations Research Letters* **2000,** 27, 1.

(48) Raman, R.; Grossmann, I. E., Modeling and Computational Techniques for Logic-Based Integer Programming. *Computers & Chemical Engineering* **1994,** 18, (7), 563.

(49) Balas, E., Disjunctive Programming and a Hierarchy of Relaxations for Discrete Continuous Optimization Problems. *SIAM Journal on Algebraic and Discrete Methods* **1985,** 6, (3), 466.

(50) Fisher, M. L., The Lagrangean relaxation method for solving integer programming problems. *Management Science* **1981,** 27, (1), 18.

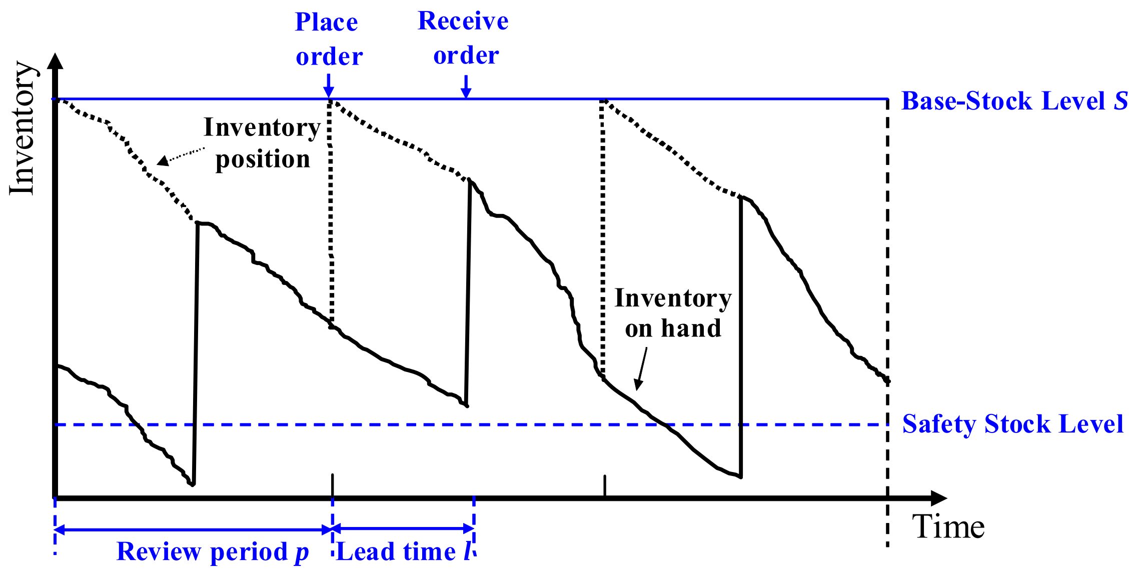
(51) Fisher, M. L., An application oriented guide to Lagrangian relaxation. *Interfaces* **1985,** 15, (2), 2.

(52) Beasley, J. E., Lagrangean heuristics for location problems. *European Journal of Operations Research* **1993,** 65, (3), 383.

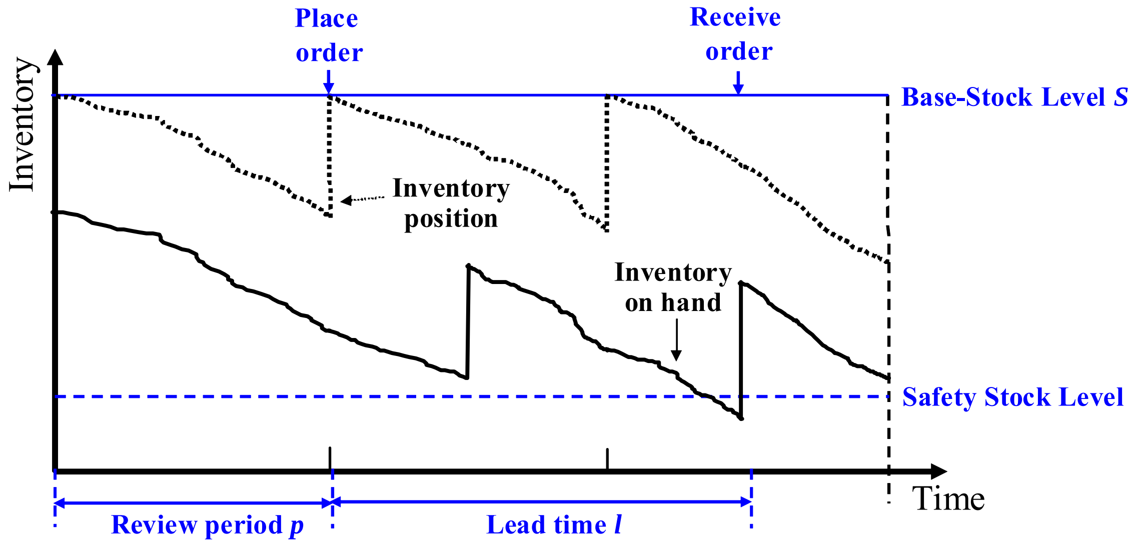
(53) You, F.; Grossmann, I. E., Optimal Design and Operational Planning of Responsive Process Supply Chains. In *Process System Engineering: Volume 3: Supply Chain Optimization*, Papageorgiou; Georgiadis, Eds. Wiley-VCH: Weinheim, 2007; pp 107.

(54) You, F.; Grossmann, I. E., Design of Responsive Process Supply Chains under Demand Uncertainty. *Computers & Chemical Engineering* **2008,** 32, (12), 2839.

List of Figure Captions



a) Review period greater than replenishment lead time



b) Review period less than replenishment lead time

Figure 1 Periodical review base stock inventory policy

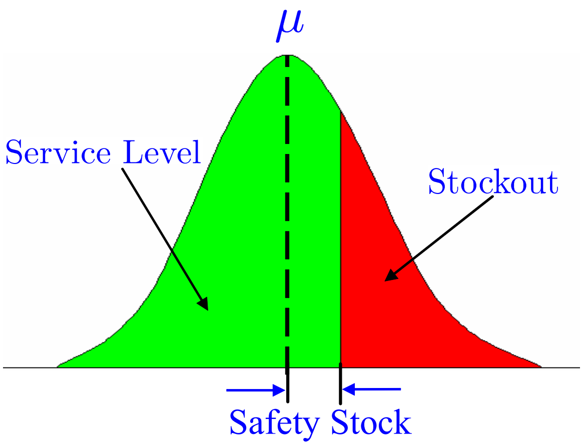


Figure 2 Safety stock level for normally distributed demand

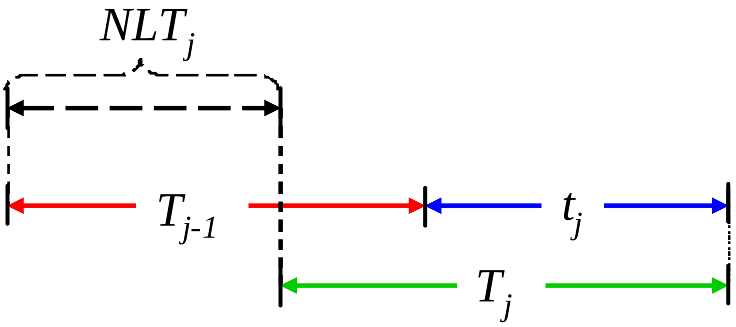


Figure 3 Timing Relationship in guaranteed service approach

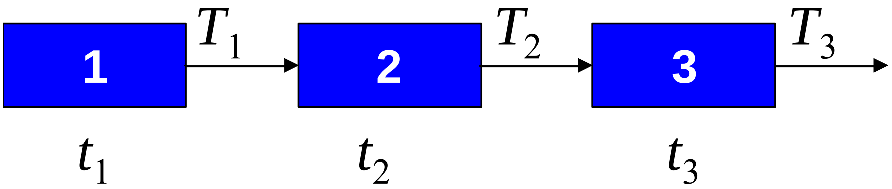


Figure 4 Times in guaranteed service approach in a three-stage serial inventory system

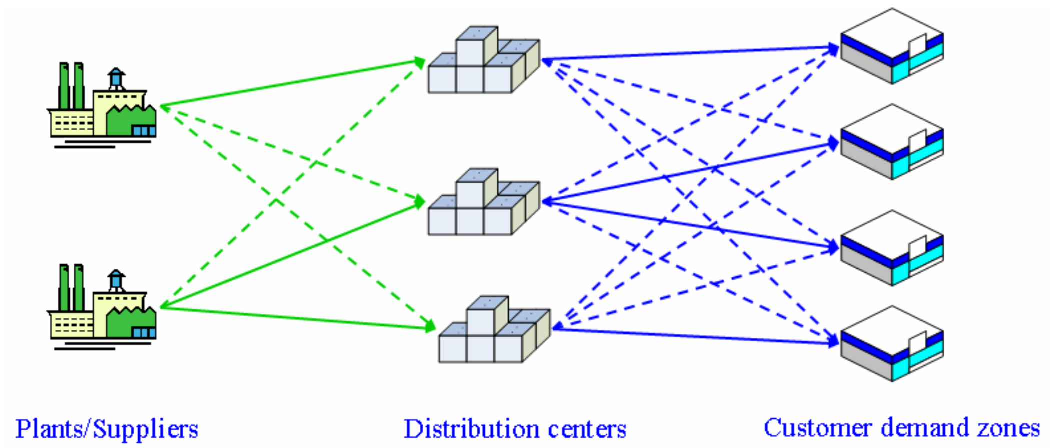


Figure 5 Supply chain network structure (three echelons)

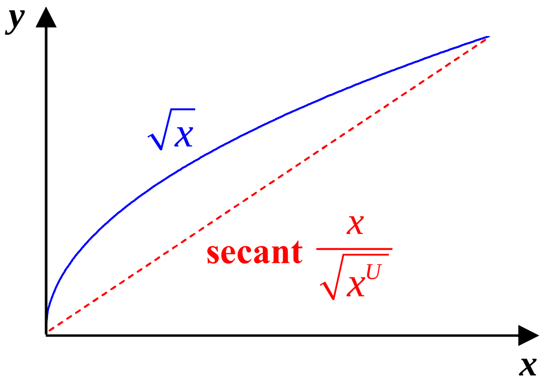


Figure 6   Secant of a univariate square root term

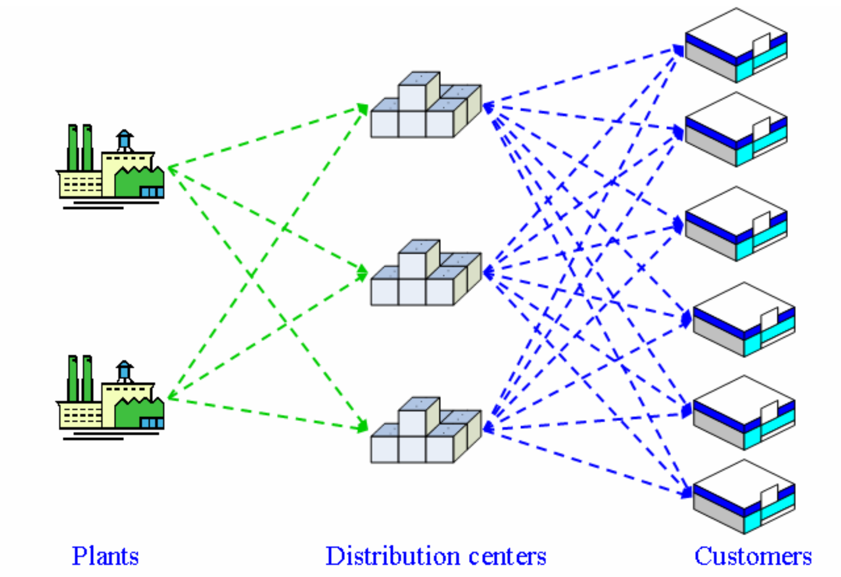
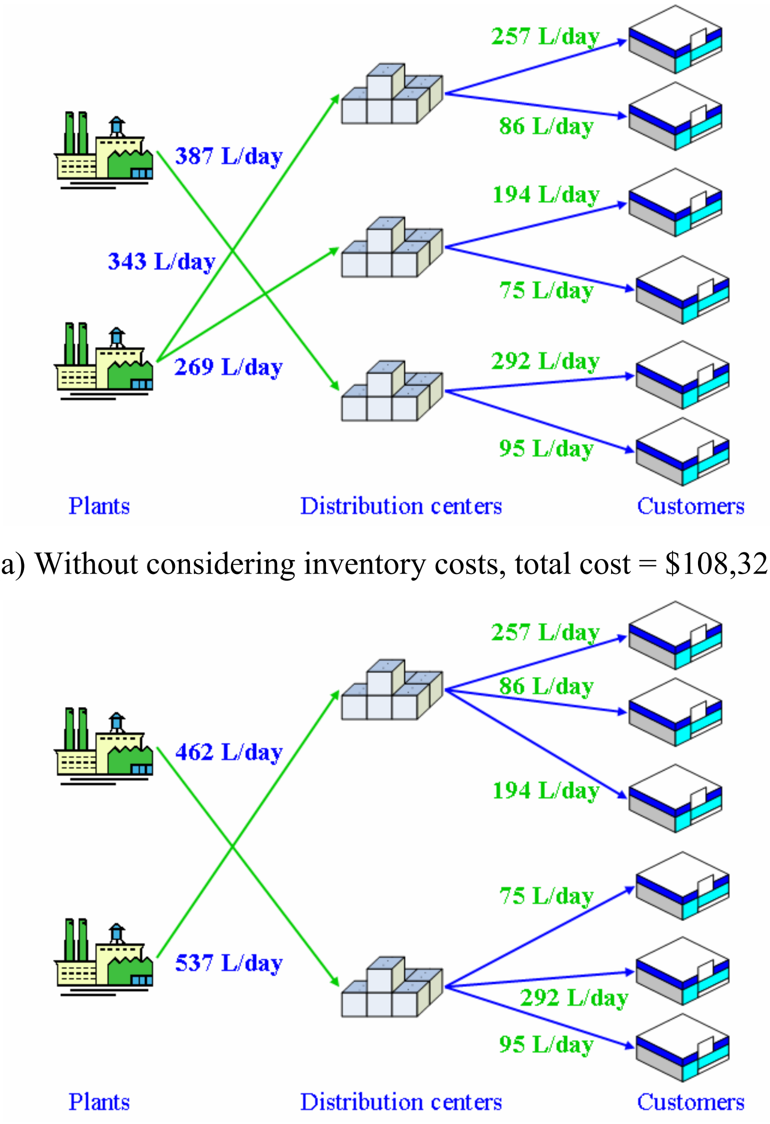


Figure 7 LOX supply chain network superstructure for Example A



(a) Without considering inventory costs, total cost = \$108,329

(b) Considering inventory costs, total cost = \$152,107

Figure 8 Optimal network structure for the LOX supply chain in Example A

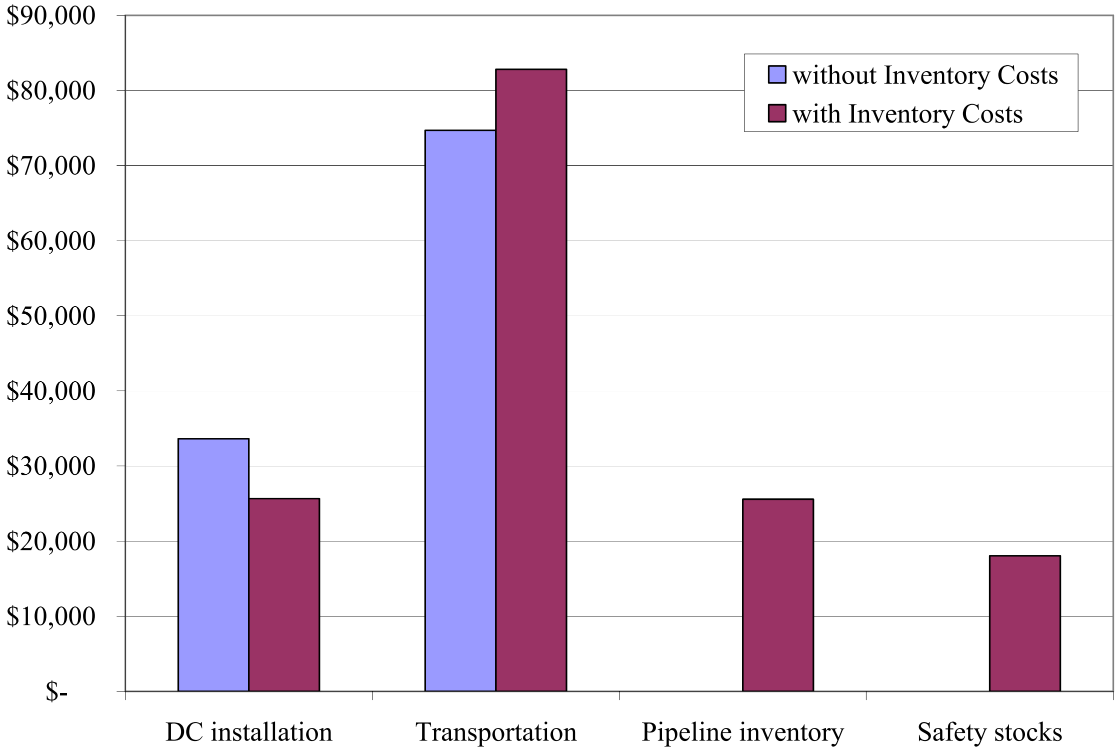


Figure 9 Cost component of the LOX supply chain in Example A

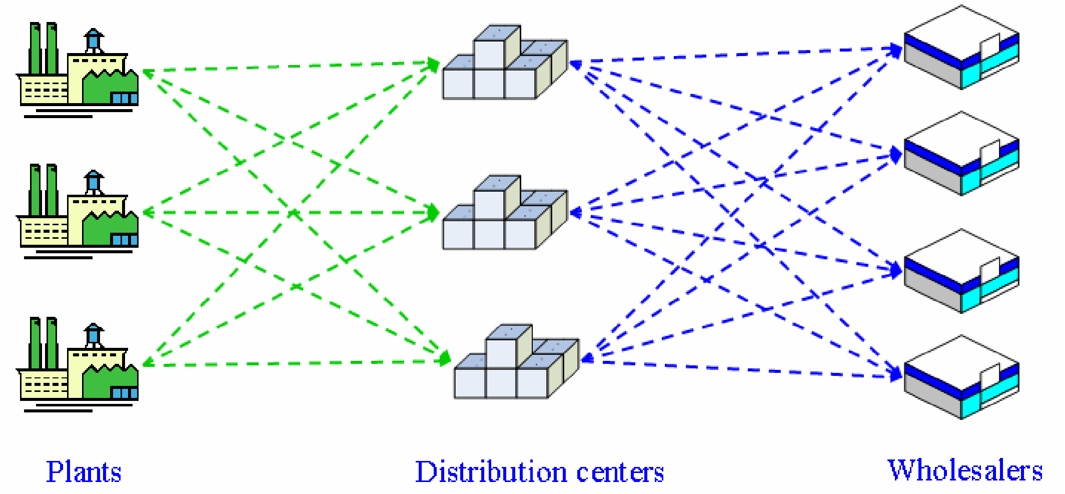


Figure 10 Acetic acid supply chain network superstructure for Example B

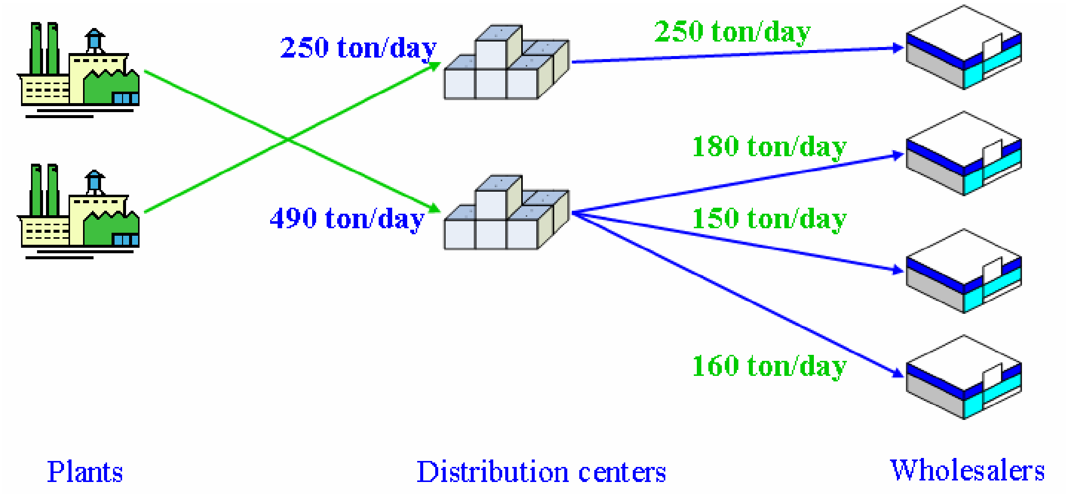


Figure 11 Optimal network structure for the acetic acid supply chain (Example B)

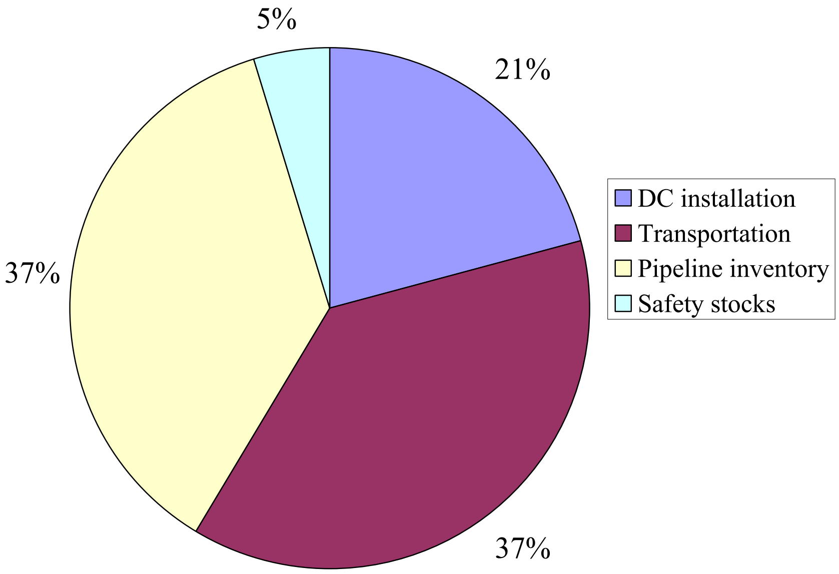


Figure 12 Cost component of the acetic acid supply chain in Example B

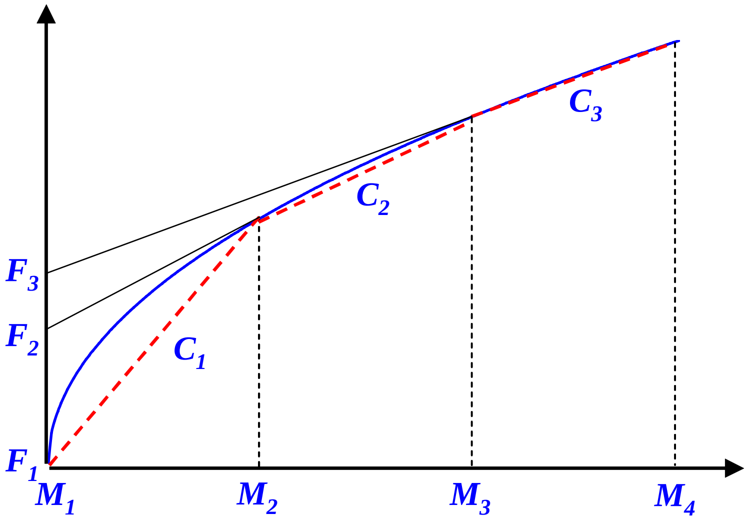
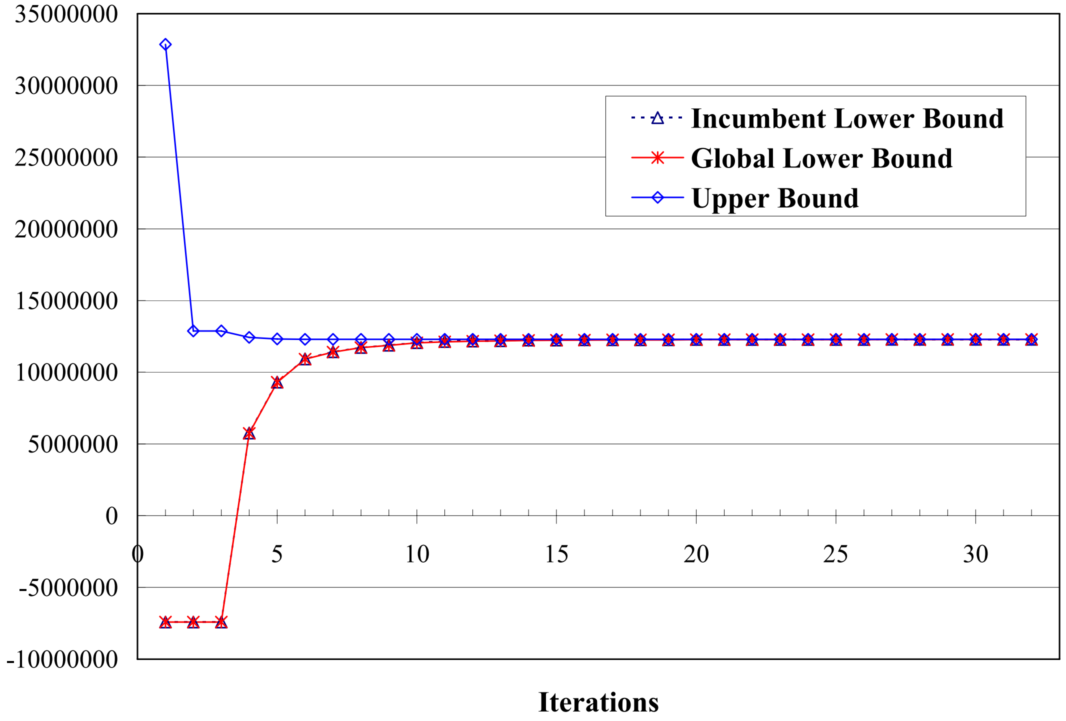
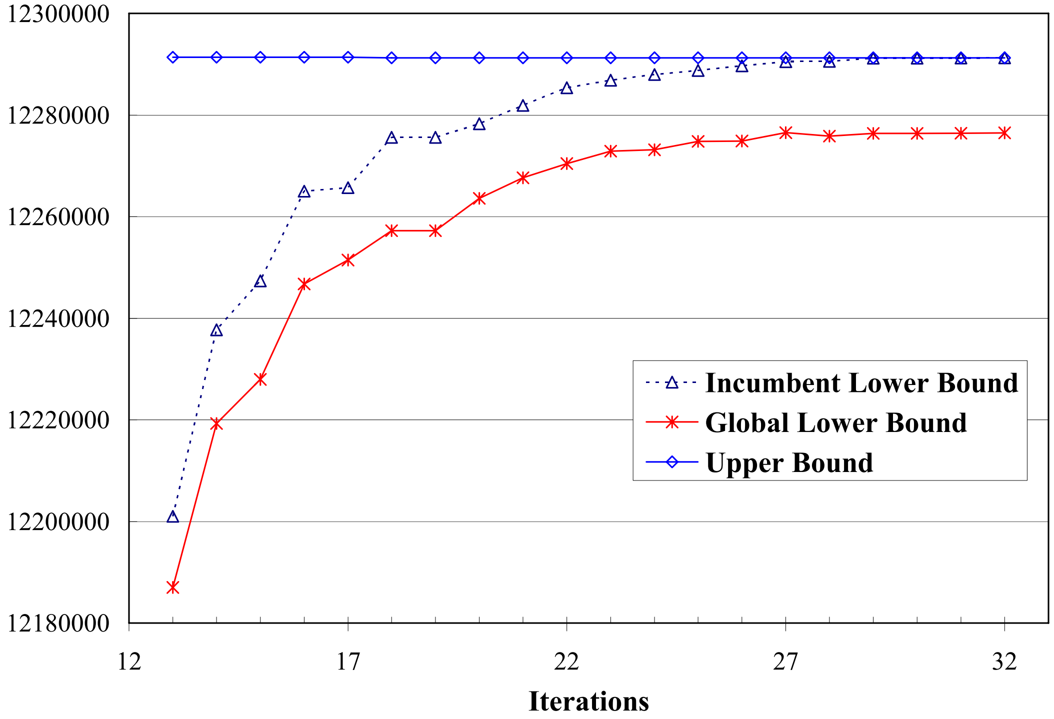


Figure 13 Piecewise linear function to approximate square root term



(a) Bounds for Iteration 1 to Iteration 33 (the last iteration)



(b) Bounds for Iteration 13 to Iteration 33 (the last iteration)

Figure 14 Bounds of each iteration of the Lagrangean relaxation algorithm for instance with 3 plants, 50 potential DCs and 150 customer demand zones

Table 1 Parameters for demand uncertainty for Example A

	Mean demand μ_i (liters/day)	Standard Deviation σ_i (liters/day)
Customer 1	257	150
Customer 2	86	25
Customer 3	194	120
Customer 4	75	45
Customer 5	292	64
Customer 6	95	30

Table 2 Order processing time ($t1_{ij}$) between plants and DCs (days) for Example A

	DC1	DC2	DC3
Plant 1	7	4	2
Plant 2	2	4	7

Table 3 Order processing time ($t2_{jk}$) between DCs and customers (days)

	Cust. 1	Cust. 2	Cust. 3	Cust. 4	Cust. 5	Cust. 6
DC1	2	2	3	3	4	4
DC2	4	4	1	1	4	4
DC3	4	4	3	3	2	2

Table 4 Unit transportation Cost ($c1_{ij}$) from plants to DCs ($/liter)

	DC1	DC2	DC3
Plant 1	0.24	0.20	0.20
Plant 2	0.18	0.19	0.23

Table 5 Unit transportation Cost ($c2_{jk}$) from DCs to customers ($/liter)

	Cust. 1	Cust. 2	Cust. 3	Cust. 4	Cust. 5	Cust. 6
DC1	0.01	0.03	0.10	0.44	1.60	2.30
DC2	1.50	0.25	0.01	0.02	0.25	1.50
DC3	2.27	1.73	0.51	0.10	0.01	0.03

Table 6 Parameters for demand uncertainty for Example B

	Mean demand μ_i (ton/day)	Standard Deviation σ_i (ton/day)
Wholesaler 1	250	150
Wholesaler 2	180	75
Wholesaler 3	150	80
Wholesaler 4	160	45

Table 7 Order processing time ($t1_{ij}$) between vendors and DCs (days) for Example B

	DC1	DC2	DC3
Vendor 1	4	4	2
Vendor 2	2	4	3
Vendor 3	3	4	4

Table 8 Order processing time ($t2_{jk}$) between DCs and wholesalers (days)

	Wholesaler 1	Wholesaler 2	Wholesaler 3	Wholesaler 4
DC1	2	2	3	3
DC2	4	4	1	1
DC3	4	4	3	3

Table 9 Unit transportation Cost ($c1_{ij}$) from vendors to DCs (\$/ton)

	DC1	DC2	DC3
Vendor 1	1.8	1.6	2.0
Vendor 2	2.4	2.2	1.3
Vendor 3	2.0	1.3	2.5

Table 10 Unit transportation Cost ($c2_{jk}$) from DCs to wholesalers (\$/ton)

	Wholesaler 1	Wholesaler 2	Wholesaler 3	Wholesaler 4
DC1	1.0	3.3	4.0	7.4
DC2	1.0	0.5	0.1	2.0
DC3	7.7	7.3	5.1	0.1

Table 11 Problem sizes for the medium and large scale instances

i	*j*	*k*	MINLP (P0)			MINLP (P1) and MILP (P2)			MILP (APLP_j), 20 intervals			NLP (APLR_j), *X* and *Z* fixed		
			Dis. Var.	Con. Var.	Const.	Dis. Var.	Con. Var.	Const.	Dis. Var.	Con. Var.	Const.	Con. Var.	Const.	NL-NZ
2	20	20	460	60	480	460	2,480	5,300	442	564	1,165	144	282	20
5	30	50	1,680	110	1,660	1,680	13,640	33,190	1,105	2,658	3,355	504	1,152	50
10	50	100	5,550	200	5,300	5,550	70,250	185,350	2,210	5,814	8,205	1,504	3,802	100
20	50	100	6,050	200	5,300	6,050	120,250	335,350	2,220	6,824	11,205	2,504	6802	100
3	50	150	7,700	250	7,900	7,700	52,800	120,450	3,303	4,352	9,155	1,204	2,552	150

"Dis. Var." = discrete variables; "Con. Var." = continuos variable; "Const." = constraints; "NL-NZ" = nonlinear nonzeros

Table 12 Comparison of the performance of the algorithms for medium and large scale instances

* No solution or bounds were returned due to solver failure.

i	*j*	*k*	Solve (P0) directly with BARON				Solve (P2) with CPLEX for at most 1 hour, then solve (P1) with DICOPT or SBB				Lagrangean Relaxation Algorithm				
							DICOPT		SBB						
			Solution	LB	Gap	Time (s)	Solution	Time(s)	Solution	Time(s)	Solution	Global LB	Global Gap	Time (s)	Iter.
2	20	20	1,889,577	1,159,841	62.92%	36,000	1,820,174	140.3	1,813,541	163.6	1,776,969	1,775,957	0.06%	175.0	11
5	30	50	---*	---*	---*	36,000	---**	36,000	---**	36,000	4,417,353	4,403,582	0.31%	3,279	24
10	50	100	---*	---*	---*	36,000	---**	36,000	---**	36,000	7,512,609	7,477,584	0.47%	27,719	42
20	50	100	---*	---*	---*	36,000	---**	36,000	---**	36,000	5,620,045	5,576,126	0.79%	27,748	53
3	50	150	---*	---*	---*	36,000	---**	36,000	---**	36,000	12,291,296	12,276,483	0.12%	16,112	32

** No solution was returned after 10 hours