Heuristic algorithm for solving large location-inventory problems with demand uncertainty.

Matias Schuster,
Jean-Sébastien Tancrez,
Louvain School of Management
Heuristic algorithm for solving large location-inventory problems with demand uncertainty.

Matias Schuster, Louvain School of Management
Jean-Sébastien Tancrez, Louvain School of Management

Summary

In this paper, we analyze a location-inventory problem for the design of large supply chain networks with uncertain demand. We give a continuous non-linear formulation that integrates location, allocation and inventory decisions, and includes the costs of transportation, cycle inventory, safety stock, ordering and facility opening. Then, relying on the fact that the model becomes linear when fixing some variables, we propose a heuristic algorithm that solves the resulting linear program and uses the solution to improve the variables estimations for the next iteration. In order to show the efficiency of the algorithm, we compare our results with the conic quadratic formulation of the problem. Computational experiments show that the heuristic algorithm can be efficiently used to find fast and close to optimal solutions for large supply chain networks. Finally, we provide managerial insights regarding the ways demand uncertainty and risk pooling affect the design of a supply chain.

Keywords: Location; Supply chain network design; Location-inventory model; Risk pooling; Conic quadratic mixed-integer program.

This work is financially supported by the Belgian Foundation of Research Science (FNRS) through the FRESH grant.

Corresponding author:
Ph.D student Matias Schuster
Center for Excellence: CESCM
Louvain School of Management / Campus Mons
Université catholique de Louvain
151, Chaussée de Binche
7000 Mons, BELGIUM
Email: matias.schuster@uclouvain.be

The papers in the WP series have undergone only limited review and may be updated, corrected or withdrawn without changing numbering. Please contact the corresponding author directly for any comments or questions regarding the paper.

President-ilsm@uclouvain.be, ILSM, UCL, 1 Place des Doyens, B-1348 Louvain-la-Neuve, BELGIUM
Heuristic algorithm for solving large location-inventory problems with demand uncertainty

Matias Schuster Puga\textsuperscript{a,}\textsuperscript{*}, Jean-Sébastien Tancrez\textsuperscript{b}

\textsuperscript{a}Louvain School of Management, Université catholique de Louvain, Belgium

Abstract

In this paper, we analyze a location-inventory problem for the design of large supply chain networks with uncertain demand. We give a continuous non-linear formulation that integrates location, allocation and inventory decisions, and includes the costs of transportation, cycle inventory, safety stock, ordering and facility opening. Then, relying on the fact that the model becomes linear when fixing some variables, we propose a heuristic algorithm that solves the resulting linear program and uses the solution to improve the variables estimations for the next iteration. In order to show the efficiency of the algorithm, we compare our results with the conic quadratic formulation of the problem. Computational experiments show that the heuristic algorithm can be efficiently used to find fast and close to optimal solutions for large supply chain networks. Finally, we provide managerial insights regarding the ways demand uncertainty and risk pooling affect the design of a supply chain.

Keywords: Location; Supply chain network design; Location-inventory model; Risk pooling; Conic quadratic mixed-integer program.

1. Introduction

In today’s competitive environment, supply chain management is essential in order to reduce costs, improve customer service, achieve balance between costs and services, and

\textsuperscript{*}Corresponding author. Address: Université catholique de Louvain, Chaussée de Binche, 151, 7000 Mons, Belgium. Tel.: +32 65 323 357.

Email addresses: matias.schuster@uclouvain.be (Matías Schuster Puga), js.tancrez@uclouvain.be (Jean-Sébastien Tancrez)

Preprint submitted to Elsevier September 22, 2015
thereby give competitive advantage to a company. In particular, the number and the locations of facilities is a strategic and critical factor in the success of any supply chain. In fact, experts suggest that as much as 80% of the costs of the supply chain are locked in with the location of the facilities and the determination of optimal flows of products between them (Watson [28]). In this context, location analysis is an important topic, and a growing research field aims at refining the location models by adding inventory decisions to the classic location-allocation decisions. Inventory decisions may affect the optimal facility location in several ways. For example, they may impact the shipment size and frequency of the vehicles, thus the transportation costs and the facility locations that are best balancing these costs. Inventory decisions further affect the location decisions when demand uncertainty is taken into account, as it forces companies to store safety stocks of products at distribution centers (DCs) close to customers in order to react quickly to variations and meet customers expectations.

In this paper, our goal is to integrate all these considerations by proposing a location-inventory model and a solution procedure that allows the design of large supply chains, while accounting for demand uncertainty. We consider two level supply chain networks composed of retailers, DCs and one central plant, and aim at designing networks with hundreds of retailers and potential DC locations. The studied optimization model simultaneously integrates three decisions: number and locations of facilities opened, allocation of flows between facilities and retailers, and inventory decisions. The objective is to minimize location, transportation and inventory costs while satisfying retailer’s demand and ensuring a specific service level. One of the characteristics of our model is that inventory decisions include the determination of shipment sizes from DCs to retailers. In line with this feature, we compute transportation costs as a cost per vehicle, which often is more realistic in practice, instead of the classic cost per product. Inventory decisions also include the determination of the safety stock level at DCs and retailers.

To include these features, we propose a continuous non-linear formulation for this location-inventory problem. In order to solve it, we present an iterative heuristic algorithm that takes advantage of this specific continuous formulation. We show that by fixing some variables,
the formulation becomes linear. At each iteration, the heuristic estimates these variables, solves the resulting linear program, and uses the solution to improve the variables estimations used in the next iteration. To show the efficiency of the heuristic, we conduct a large set of computational experiments, and compare the results from the heuristic algorithm with those given by the conic quadratic formulation of the same problem. Furthermore, we show, using our computational results, managerial insights related to the risk pooling effect, the impact of demand uncertainty on the strategic design of a supply chain and the trade-offs raised from combining location and inventory decisions.

The main contributions of this paper can be summarized as follows. To begin with, we propose a new model for the location-inventory problem with demand uncertainty. It leads to an efficient solution procedure and allows to include realistic features sparsely addressed in the literature such as the shipment size decision, transportation cost per vehicle, inventory at retailers locations, and vehicles capacities. In addition, and this is an important goal, our approach allows to efficiently design large supply chains. We solve instances for up to 1000 retailers and potential DC locations (in around 5 hours). Furthermore, we show for instances up to 100 retailers that the heuristic algorithm obtains good solutions (less than 2% gap), and much faster than the conic quadratic mixed integer program (CQMIP). Last, we use the computational results to explore the way the supply chain design is affected by demand uncertainty, and show how, when considering the DC location decisions, the risk pooling effect at DCs may be mitigated by the increase of safety stocks at retailers.

The remainder of the paper is organized as follows. The next section reviews the existing literature on supply chain network design and location-inventory models. In Section 3 we present the features of the location-inventory problem and our mathematical formulation. Section 4 is dedicated to the description of the heuristic algorithm. In Section 5 we show the results of the computational experiments to assess the efficiency of our heuristic. We also discuss managerial insights. Finally, we conclude in Section 6 and give ideas for future work.
2. Literature Review

Our research aims at contributing to the literature on integrated location-inventory models and supply chain network design. In this section, we review the existing models, their characteristics and the methodologies used, with a particular interest for the papers that take demand uncertainty into account.

Location-inventory models have been introduced as an extension to classic facility location models. For a detailed review on the facility location literature, we refer the reader to Owen and Daskin [16], ReVelle and Eiselt [20], Bravo and Vidal [4], and Klose and Drexl [9]. The basic location models may be generalized by integrating tactical and operational decisions, such as inventory, production, routing or procurement, to address strategic supply chain network design (Beamon [2]; Melo et al. [12]). Mula et al. [14] also review different modeling approaches for integrating production and transportation decisions to the facility location problem. Shen [23] gives an extensive survey on three different integrated models: location-inventory, location-routing and inventory-routing models. Recently, Farahani et al. [8] provide a review on location-inventory modeling. They present a basic model, which is extended by progressively incorporating features related to tactical and operational decisions.

In the present paper, we focus on integrating inventory decisions, and propose a location-inventory model. The pioneering works in this field date from around 2000. Nozick and Turnquist [15] discuss how to account for inventory costs and offer a simple linear approximation of these costs in order to integrate them into a classic facility location model. Erlebacher and Meller [7] formulate the location-inventory problem as a non-linear integer program, and show different versions of a heuristic algorithm for solving it. Furthermore, in the present paper, our model takes into account safety stocks to address demand uncertainty. In this regard, our problem can be related to the location-inventory model with risk pooling presented by Shen et al. [24]. The model is described as follows: given a collection of retailers, each one with uncertain demand, the model determines the location of the distribution centers (DCs), the flows allocation, how often to reorder and which level of safety stock to main-
tain at DCs in order to minimize costs (i.e., opening, transportation and inventory), while ensuring a specified service level. The paper shows that centralizing stocks at DCs leads to inventory reductions, via risk pooling. Their formulation is expressed as a mixed-integer nonlinear location-allocation model. The authors then reformulate the problem as a linear integer set-covering problem. Computational results are provided for instances with 33 to 150 retailers. Daskin et al. [5] solve this location-inventory problem using Lagrangian relaxation, also limited to instances with up to 150 customers. This model is extended by Ozsen et al. [17, 18] with the incorporation of storage capacity and multiple sourcing. In a similar fashion, Miranda and Garrido [14] combine a capacitated facility location problem with inventory decisions, such as safety stock level. The resulting non-linear mixed integer problem is solved using Lagrangian relaxation and sub-gradient methods for instances with 20 customers and 10 potential DC locations. Park et al. [19] formulate the location-inventory problem as a nonlinear integer programming model, for which a two-phase heuristic solution algorithm is derived based on the Lagrangian relaxation approach. Based on the same solution approach, Kumar and Tiwari [10] show the benefits of risk pooling in a three layer supply chain and compare the savings of a centralized system with a decentralized one. Test problems consider a maximum number of 5 potential sites for plants, 10 potential locations for DCs, and 30 retailers to be served. Sourirajan et al. [25] solve their location-inventory model using genetic algorithms and show the trade-offs between lead times and inventory risk pooling. Teo and Shu [27] structure the location-inventory problem as a set-partitioning integer-programming model and solve it using column generation. Thereafter, this work has been extended by Romeijn et al. [21], considering possible stockout and a maximum of 70 retailers. Lin et al. [11] propose a hybrid evolutionary algorithm for solving a four echelon location-inventory problem, and test it for a maximum of 60 customers. Diabat et al. [6] present a Lagrangian relaxation based heuristic for a three level model. They give computational results for up to 250 retailers. Berman et al. [3] solve a coordinated location-inventory problem using a Lagrangian relaxation algorithm. They compute 250 experiments showing an average gap of 1.74% for 96 DC potential locations. Atamtürk et al. [1] study several variants of the location-inventory problem and formulate them as conic quadratic mixed-
integer programs (CQMIPs). The CQMIP approach allows the authors to solve the problem to optimality using standard optimization software packages with up to 150 customers. Shahrabi et al. [22] also reformulate their integer model as a compact CQMIP for a four echelon supply chain network. Their experiments consider a maximum of 10 suppliers, 10 potential warehouse locations, 20 potential hub locations and 200 customers.

Compared to these papers, our research differs in three aspects. First, our model includes features that are not present in the cited papers. In particular, we include shipment size decisions, transportation cost per truck (instead of the classic cost per items) and inventory at retailers locations. Second, our modeling approach leads to an efficient heuristic based on the iterative solution of linear programs. Third, this approach allows us to design large supply chains (up to 1000 retailers and DC potential locations). Finally, we note that our paper builds on Tancrez et al. [26], using a similar modeling framework. However, the present paper extends it and adds several features. Our paper includes safety stocks and demand uncertainty, adapts and improves the heuristic algorithm, proposes a conic quadratic formulation of the problem to assess the efficiency of the heuristic and explores new managerial insights.

3. Location-Inventory Model

In this section, we introduce the main characteristics of our location-inventory model. We aim at designing a two level supply chain network in which products are delivered from a central plant to retailers, passing through distribution centers (DCs). We consider a single-period planning horizon and a single product, and assume that the central plant has unlimited production capacity. The location of the retailers is known and fixed, and retailers may be supplied by more than one DC (multiple sourcing).

Our integrated model simultaneously determines three decisions: (i) number and locations of DCs opened; (ii) flows between DCs and retailers; (iii) inventory decisions. The objective is to minimize facility opening, transportation and inventory costs. Two types of inventory are integrated: cycle and safety inventory. On the one side, the model considers cycle inventory kept at retailers and DCs. It is influenced by two inventory decisions: the
shipment size between DCs and retailers, and the order size from the DCs to the central plant. The shipment size decisions are limited by the vehicle capacities and are taken to balance transportation costs and holding costs at retailers. A larger shipment size would help reducing the shipment frequency, but would induce a larger inventory. Accordingly, in our model, the transportation cost is accounted as a cost per vehicle, proportional to the number of vehicles used, regardless of how loaded the vehicles are. This approach differs from the classic one, in which the transportation costs are proportional to the number of units shipped. The cycle inventory at DCs is determined by the order sizes to the central plant, by balancing ordering and holding costs. Note that our model could be interpreted as locating factories (without a central plant) instead of locating DCs. In this case, the order cost would correspond to a production setup cost, and the order size to a batch size.

On the other side, the model includes safety stocks at the DCs and at the retailers, to respond to demand uncertainty. We suppose that demands are uncorrelated and normally distributed, with a specific mean and variance for each retailer. The level of safety stock depends on the demand variability, on the lead times (i.e., between the central plant and the DCs, and between the DCs and the retailers) and on the required service level. As the lead times are proportional to the distances, the safety stock and the location decisions are interdependent. Moreover, location decisions are affected by safety stocks due to risk pooling effects, when demand is aggregated across locations (see Subsection 5.3 for a discussion). In the following, we present in detail our mathematical formulation for this location-inventory problem, starting by introducing the notation.

3.1. Notation

In this subsection, we list the indexes, parameters and the decision variables used in our model.

Indexes:

- \( r \in \{1, \ldots, n_r\} \) for retailers,
- \( d \in \{1, \ldots, n_d\} \) for potential DC locations.

Parameters:

...
$O^d_r$: transportation cost for one vehicle from DC $d$ to retailer $r$, in €/vehicle,
$K^d$: fixed cost at DC $d$ of placing an order to the central plant, in €/order,
$F^d$: fixed cost of opening a DC $d$, in €/period,
$H_r$: unit inventory holding cost at retailer $r$, in €/(item · period),
$H^d$: unit inventory holding cost at DC $d$, in €/(item · period),
$z_\alpha$: standard normal deviation associated with service level $\alpha$,
$LT^d_r$: lead time between DC $d$ and retailer $r$, in periods,
$LT^d$: lead time between the central plant and DC $d$, in periods,
$\Lambda_r$: mean demand at retailer $r$, in items/period,
$\sigma_r$: standard deviation of demand at retailer $r$, in items/period,
$C^d_r$: capacity of vehicles from DC $d$ to retailer $r$, in items/vehicle.

**Decision Variables:**

$\lambda^d_r$: product flow from DC $d$ to retailer $r$, in items/period,
$q^d_r$: shipment size from DC $d$ to retailer $r$, in items/vehicle,
$Q^d$: size of order to the central plant at DC $d$, in items,
$\Lambda^d$: product flow through DC $d$, later referred to as "DC flow", in items/period,
$\sigma^d$: standard deviation of the product flow through DC $d$, in items/period.

Note that all the decision variables included are continuous (no integer variables are considered). In particular, we do not use a binary variable for the DC opening decision.

### 3.2. Cost function

In this subsection, we describe the various costs included in our model. Overall, we aim at minimizing the total expected transportation, inventory, opening, ordering and safety stock costs per period. The **transportation cost** is computed as the cost per vehicle, $O^d_r$, times the number of vehicles shipped per period, $\lambda^d_r/q^d_r$. The cycle **inventory cost** at retailers is the unit inventory cost $H_r$, times the average inventory level at retailers. The latter equals half the shipment size $q^d_r/2$ when only one DC is supplying it. When a retailer is supplied by several DCs, the shipment sizes are simply weighted by the relative proportion of time where the retailer stock is filled from one of the DCs, that is $\lambda^d_r/\Lambda_r$. The two aforementioned
costs lead to the following equation:

\[ \sum_{d,r} O^d_r \cdot \frac{\lambda^d_r}{q^d_r} + \sum_r H_r \cdot \left( \sum_d \frac{\lambda^d_r \cdot q^d_r}{2} \right) \]  

(1)

In fact, the shipment size decision variables, \( q^d_r \), appears only in these two terms, as it is only impacting the transportation from DCs to retailers and the cycle inventory at retailers. By derivating Equation (1), then equaling to zero, and accounting for the vehicle’s capacity, we obtain the following closed-form formula for the shipment size, which can thus be computed prior to the actual optimization program:

\[ q^d_r = \min \left( C^d_r ; \sqrt{\frac{2 \cdot O^d_r \cdot \Lambda_d}{H_r}} \right) \quad \forall r, d. \]  

(2)

The cycle inventory cost at DC \( d \) is simply computed as the unit holding cost, \( H_d \), times the average inventory level, \( Q_d/2 \). The order cost is given by the fixed ordering cost, \( K_d \), times the number of orders executed per period, \( \Lambda_d/Q_d \). Thus, the ordering and inventory costs are:

\[ \sum_d H_d \cdot \frac{Q_d}{2} + \sum_d K_d \cdot \frac{\Lambda_d}{Q_d} \]  

(3)

The order size variable, \( Q_d \), is only included in these costs, and can be computed as usually in a EOQ structure, leading to \( Q_d = \sqrt{2 \cdot K_d \cdot \Lambda_d / H_d} \). The holding and order costs at DC \( d \) (3) thus become:

\[ \sum_{d, \Lambda_d > 0} \sqrt{\frac{2 \cdot H_d \cdot K_d}{\Lambda_d}} \cdot \sum_r \lambda^d_r \]  

(4)

Next, the objective function includes the fixed opening cost of a DC. As we aim at avoiding integer variables, we do not introduce the classic expression \( \sum_d F_d \cdot y_d \), where \( y_d \) is a binary variable determining whether DC \( d \) is opened or not. Instead, we use \( \sum \lambda^d_r / \Lambda_d \), which equals zero when there is no retailer linked to DC \( d \), and becomes one when DC \( d \) is
opened, leading to the following term:

\[
\sum_{d, \Lambda_d > 0} \frac{\sum_r \lambda^d_r}{\Lambda_d} \cdot F_d \cdot \sum_r \lambda^d_r
\]  

This modeling trick allows us to avoid integer variables, but leads to the introduction of divisions by variable \( \Lambda_d \). However, in our model, we already have divisions by \( \Lambda_d \) (Equation (4)), and thus, it does not further complicate our model.

Finally, the last terms relate to the **safety stock costs** at DCs and at retailers, which equal the unit holding cost times the average safety stock. The safety stock is held in order to be able to satisfy all the demand during the lead time with a probability equal to the service level \( \alpha \). To compute the average safety stock, the corresponding normal standard deviation, \( z_\alpha \), has thus to be multiplied by the standard deviation of the demand during the lead time. At DCs, the latter is given by \( \sigma_d \cdot \sqrt{LT_d} \). Furthermore, we multiply each term by \( \sum_r \lambda^d_r/\Lambda_d \), so that only opened DCs are accounted (avoiding integer variables). Consequently, the safety stock cost at DCs is given by the following term:

\[
\sum_{d, \Lambda_d > 0} H_d \cdot z_\alpha \cdot \sigma_d \cdot \sqrt{LT_d} \cdot \frac{\sum_r \lambda^d_r}{\Lambda_d}
\]  

Similarly, at the retailers, if a retailer \( r \) is served by only one DC (vast majority of cases), the standard deviation of the demand during the lead time is simply given by \( \sigma_r \cdot \sqrt{LT_r^d} \). However, if a retailer is served by several DCs, we weight the various lead times coming from different DCs by their respective flows. Accordingly, the safety stock cost at retailers is:

\[
\sum_r H_r \cdot z_\alpha \cdot \sigma_r \cdot \left( \sum_d \frac{\lambda^d_r}{\Lambda_r} \cdot \sqrt{LT_r^d} \right)
\]
3.3. Mathematical Formulation

With the costs detailed in the previous subsection, we can now introduce the non-linear continuous formulation for the location-inventory problem as follows:

\[
\begin{align*}
\min & \quad \sum_{d,r} O_{d,r} \cdot \frac{\lambda_{d,r}^d}{q_{r}^d} \\
+ & \quad \sum_r H_r \cdot \left( \sum_d \frac{\lambda_{d,r}^d \cdot q_{r}^d}{\Lambda_r} \right) \\
+ & \quad \sum_{d, \Lambda_d > 0} F_d \cdot \frac{\sum_r \lambda_{d,r}^d}{\Lambda_d} \\
+ & \quad \sum_{d, \Lambda_d > 0} \sqrt{2 \cdot H_d \cdot K_d} \cdot \sum_r \lambda_{d,r}^d \\
+ & \quad \sum_r H_r \cdot z_\alpha \cdot \sigma_d \cdot \sqrt{LT_d} \cdot \frac{\sum_r \lambda_{d,r}^d}{\Lambda_d} \\
+ & \quad \sum_r H_r \cdot z_\alpha \cdot \sigma_r \cdot \left( \sum_d \frac{\lambda_{d,r}^d}{\Lambda_r} \cdot \sqrt{LT_r^d} \right)
\end{align*}
\]

s.t  \[
\sum_r \lambda_{r}^d = \Lambda_d, \quad \forall d, \quad (14)
\]
\[
\sqrt{\sum_{r, \lambda_r^d > 0} \sigma_r^2} = \sigma_d, \quad \forall d, \quad (15)
\]
\[
\sum_d \lambda_{r}^d = \Lambda_r, \quad \forall r, \quad (16)
\]
\[
\lambda_{r}^d, \Lambda_d, \sigma_d \geq 0, \quad \forall r, d. \quad (17)
\]

The first 6 terms represent the costs of transportation (8), inventory at retailers (9), fixed opening (10), ordering and inventory at DCs (11) and safety stock (12)-(13). Constraints (14)-(15) define the auxiliary variables, \( \Lambda_d \) and \( \sigma_d \). Constraints (16) ensure retailers demand satisfaction. Finally, (17) gives the non-negativity constraints.

The model is continuous non-linear. As explained previously, variable \( Q_d \) can be removed from the model (see Equation (4)) and variable \( q_{r}^d \) can be computed a priori (using Equation (2)). The only variables left are thus \( \lambda_{r}^d, \Lambda_d \) and \( \sigma_d \). The model is linear in the main
variables, $\lambda^d_r$. However, the auxiliary variables, $\Lambda_d$ and $\sigma_d$, which can be computed from the variables $\lambda^d_r$, prevent the model to be a linear program. The DC flow $\Lambda_d$ can be found in the fixed opening (10), ordering (11) and safety stock at DCs (12) cost terms. The standard deviation $\sigma_d$ only appears in the safety stock at DCs term (12). Interestingly, when $\Lambda_d$ and $\sigma_d$ are fixed, the problem becomes a linear program in the flow variables $\lambda^d_r$. In the next section, we develop an algorithm that takes advantage of this property.

4. Heuristic algorithm

In this section, we explain the solution procedure used for solving the location-inventory model given in Section 3. The procedure relies on the characteristics of our model, and in particular, on the fact that it simplifies to a continuous linear program when $\Lambda_d$ and $\sigma_d$ are fixed. We propose an iterative heuristic algorithm that, at each iteration, estimates the variables $\Lambda_d$ and $\sigma_d$, solves the resulting linear program to find the flow variables $\lambda^d_r$, then uses these solutions to improve the estimations of the variables $\Lambda_d$ and $\sigma_d$, and ultimately reach a good solution for the problem. Algorithm 1 shows the outline of the heuristic algorithm.

The core of the algorithm is the iterative improvement of the estimations of $\Lambda_d$ and $\sigma_d$ (noted $\Lambda_d^{Est}$ and $\sigma_d^{Est}$). After testing different approaches, we have selected two ways to update $\Lambda_d^{Est}$ and $\sigma_d^{Est}$. They are executed consecutively. To begin with, stage one updates $\Lambda_d^{Est}$ and $\sigma_d^{Est}$ by simply equaling them to the values found in the previous iteration (noted $\Lambda_d^{Sol}$ and $\sigma_d^{Sol}$, directly deduced from the values of $\lambda^d_r$ by using (13) and (14)). This update process is simple and quick but has the drawback to limit the options early in the algorithm. Indeed, when a DC is not opened in one iteration (and $\Lambda_d^{Sol} = 0$), the estimations $\Lambda_d^{Est}$ and $\sigma_d^{Est}$ will remain zero for all the next iterations. In other words, if a DC is not opened in one iteration, it will never be opened in the next iterations, and it will not be part of the final solution. Therefore, after this stage, the algorithm enters a second stage that does not share the same drawback. Stage two updates $\Lambda_d^{Est}$ and $\sigma_d^{Est}$ by equaling them to the average between the solution found ($\Lambda_d^{Sol}$ and $\sigma_d^{Sol}$) and the estimations used to find it (the previous $\Lambda_d^{Est}$ and $\sigma_d^{Est}$). This updating process is thus smoother. When a DC is not opened in one
Algorithm 1: Heuristic

\[ \text{NoImprov} = 0, \ MS = 0, \ nDC = n_r/10, \ \forall \ d. \]
\[ \text{PrevSol, CurrSol and BestSol set different and very large.} \]
\[ \textbf{while } MS \leq \text{MaxMS do} \]
\[ \quad \text{if } MS \leq |\text{SetFactors}| \text{ then} \]
\[ \quad \quad \text{StartFactor}_d = \text{SetFactors}(MS), \ \forall \ d. \]
\[ \quad \text{else} \]
\[ \quad \quad \text{StartFactor}_d \text{ is randomly assigned between min(SetFactors) and max(SetFactors).} \]
\[ \quad \text{end if} \]
\[ \quad \text{Compute } \Lambda_{\text{Est}}^d \text{ and } \sigma_{\text{Est}}^d \text{ using (18)}, \ Stage = 1 \text{ and } MS = MS + 1. \]
\[ \quad \textbf{while } \text{NoImprov} < \text{MaxNoImprov} \text{ and } \text{CurrSol} \neq \text{PrevSol} \text{ do} \]
\[ \quad \quad \text{Solve LP (8)-(17) with } \Lambda_d = \Lambda_{\text{Est}}^d \text{ and } \sigma_d = \sigma_{\text{Est}}^d, \ \forall \ d. \]
\[ \quad \quad \text{Using the solution of the LP, compute } \Lambda_{\text{Sol}}^d \text{ and } \sigma_{\text{Sol}}^d \text{ from (14)-(15), and save the objective value as } \text{CurrSol}. \]
\[ \quad \quad \text{if } \text{CurrSol} < \text{BestSol} \text{ then} \]
\[ \quad \quad \quad \text{BestSol} = \text{CurrSol}, \ \text{NoImprov} = 0 \text{ and } nDC = \sum_{d, \Lambda_{\text{Sol}}^d > 0} 1. \]
\[ \quad \quad \text{else} \]
\[ \quad \quad \quad \text{NoImprov} = \text{NoImprov} + 1. \]
\[ \quad \quad \text{end if} \]
\[ \quad \quad \text{if } Stage = 1 \text{ then} \]
\[ \quad \quad \quad \Lambda_{\text{Est}}^d = \Lambda_{\text{Sol}}^d, \ \forall \ d, \]
\[ \quad \quad \quad \sigma_{\text{Est}}^d = \sigma_{\text{Sol}}^d, \ \forall \ d. \]
\[ \quad \quad \quad \text{if } \text{NoImprov} = \text{MaxNoImprov} \text{ or } \text{CurrSol} = \text{PrevSol} \text{ then} \]
\[ \quad \quad \quad \quad \text{Stage} = 2 \text{ and } \text{NoImprov} = 0. \]
\[ \quad \quad \text{end if} \]
\[ \quad \quad \text{else if } Stage = 2 \text{ then} \]
\[ \quad \quad \quad \Lambda_{\text{Est}}^d = \frac{\Lambda_{\text{Est}}^d + \Lambda_{\text{Sol}}^d}{2}, \ \forall \ d, \]
\[ \quad \quad \quad \sigma_{\text{Est}}^d = \frac{\sigma_{\text{Est}}^d + \sigma_{\text{Sol}}^d}{2}, \ \forall \ d. \]
\[ \quad \quad \text{end if} \]
\[ \quad \text{end while} \]
\[ \text{PrevSol} = \text{CurrSol}. \]
\[ \textbf{end while} \]

The best found solution is given by \textit{BestSol}.

iteration, it has a chance to be opened in later iterations and be part of the final solution, as \( \Lambda_{\text{Est}}^d \) is never equal to zero. The two stages are stopped when the best solution is not improved for a given number of iterations (noted \textit{MaxNoImprov}, equals 8 by default) or when the current solution is not changed (i.e., the algorithm is looping).

To compute the initial values of \( \Lambda_{\text{Est}}^d \) and \( \sigma_{\text{Est}}^d \), we suppose that a distribution center serves the retailers in a circular region around itself. The radius \( R_d \) of the circle around a
DC \( d \) is chosen so that it fills a fraction of the total surface of the map (noted \( S_{Tot} \)). In other words, we assume that a given number of DCs (\( nDC \)) is opened so that each one serves a fraction of the retailers (by default, \( nDC = \frac{n_r}{10} \)). The following expressions are used for computing the initial estimations:

\[
\Lambda_{Est}^d = \sum_{r, D(d, r) \leq R_d} \Lambda_r, \quad \forall \ d;
\]

\[
\sigma_{Est}^d = \sqrt{\sum_{r, D(d, r) \leq R_d} \sigma_r^2}, \quad \forall \ d.
\]

Where \( R_d = \sqrt{\frac{S_{Tot}}{nDC \cdot \pi \cdot StartFactor_d}}, \quad \forall \ d. \) (18)

And where \( D(d, r) \) is the distance between retailer \( r \) and DC \( d. \)

The heuristic performance may be sensitive to the initial values chosen for \( \Lambda_{Est}^d \) and \( \sigma_{Est}^d \). To improve the performance, we include the possibility of multiple starts (noted \( MS \)), i.e., starting with different initial values. It increases the computational time but allows to explore more solutions and thus offers the opportunity to find better solutions. This is the reason why \( StartFactor_d \) is included in the formulas (18). To compute different values for \( \Lambda_{Est}^d \) and \( \sigma_{Est}^d \), we simply modify the size of the region served by a DC \( d \) by changing its radius \( R_d \), using \( StartFactor_d. \) We apply two different approaches to compute \( StartFactor_d. \) In the first one, the value of \( StartFactor_d \) is the same for all DCs (among the values in \( SetFactors = \{0.2, 0.4, 0.6, 0.8, 1, 1.2\} \) by default). In the second one, \( StartFactor_d \) differs among DCs and takes random values (between \( \min(SetFactors) \) and \( \max(SetFactors) \) by default). The number of multiple starts is limited to \( MaxMS \) (by default, \( MaxMS = 10 \)).

5. Computational Results

In this section, we present computational experiments to assess the effectiveness of the heuristic algorithm and then discuss managerial insights.

5.1. Conic quadratic formulation

In order to assess the effectiveness of our algorithm, we first reformulate our problem as a conic quadratic mixed integer program (CQMIP), which can be solved to optimality using
standard optimization software. Similarly to Atamtürk et al. [1], we introduce auxiliary variables, \( v_{1d} \) and \( v_{2d} \), for the non linear terms of the objective function. Two binary variables are also included. Variable \( y_{d} \) represents the location decision (i.e., it equals 1 when DC \( d \) is opened, and 0 otherwise). Variable \( y_r^d \) determines the link between distribution centers and retailers (i.e., it is equal to 1 if DC \( d \) serves retailer \( r \), and 0 otherwise). Accordingly, the flow variables \( \lambda_r^d \) are substituted by \( \Lambda_r \cdot y_r^d \). Thus, in the conic quadratic formulation, retailers can only be served by one DC. Like in our model, the shipment size \( q_r^d \) is computed a priori, with Equation (2). The CQMIP is given in the following:

\[
\begin{align*}
\min \quad & \sum_{d,r} O_{d,r} \cdot \frac{\Lambda_r \cdot y_r^d}{q_r^d} \\
& + \sum_{d,r} H_r \cdot \frac{q_r^d}{2} \cdot y_r^d \\
& + \sum_{d} F_d \cdot y_d \\
& + \sum_{d} \sqrt{2 \cdot H_d \cdot K_d \cdot v_{1d}} \\
& + \frac{\sum_{d,\Lambda_d>0} H_d \cdot z_\alpha \cdot \sqrt{LT_d \cdot v_{2d}}}{2} \\
& + \sum_{r} H_r \cdot z_\alpha \cdot \sigma_r \cdot \sum_{d} \sqrt{LT_r^d \cdot y_r^d}
\end{align*}
\]

s.t \[
\begin{align*}
\sum_{r} \Lambda_r \cdot (y_r^d)^2 & \leq v_{1d}^2, \quad \forall d, \quad (25) \\
\sum_{r} \sigma_r^2 \cdot (y_r^d)^2 & \leq v_{2d}^2, \quad \forall d, \quad (26) \\
\sum_{d} y_r^d & = 1, \quad \forall r, \quad (27) \\
y_r^d & \leq y_d, \quad \forall r, d, \quad (28) \\
v_{1d}, v_{2d} & \geq 0, \quad \forall d, \quad (29) \\
y_r^d, y_d & \in \{0, 1\}, \quad \forall r, d. \quad (30)
\end{align*}
\]
The objective minimizes the total expected cost of transportation (19), inventory at retailers (20), fixed opening (21), ordering and inventory at DCs (22) and safety stock (23)-(24). Constraints (25)-(26) define the auxiliary variables $v_{1d}$ and $v_{2d}$. Constraints (27) ensure that each retailer is assigned to one DC. Constraints (28) guarantee that retailers are linked to opened DCs. Constraints (29) give the non-negativity constraints for the auxiliary variables. Finally, (30) defines the domain of the binary decision variables. Note that the objective function is linear and the constraints are either conic quadratic or linear.

5.2. Heuristic efficiency

To assess the efficiency of our heuristic algorithm, we carry out extensive numerical experiments. A first set of experiments aims to show the accuracy of the heuristic results, by comparing them with those obtained with the conic quadratic model.

In Table 1, we give all the values used for the different parameters. They were chosen in order to lead to a realistic number of opened DCs (i.e., smaller than 20% of the number of retailers). Due to the large computational time that the CQMIP takes to solve large instances, we focus on supply chain networks with 20, 50 and 100 retailers. The retailers are located in a square, whose coordinates are randomly assigned in $[0, \text{Width}]$. The potential DC locations are chosen to be the retailers locations. The distances between retailers and

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indexes</td>
<td></td>
</tr>
<tr>
<td>$n_r = n_d$</td>
<td>(20, 50, 100)</td>
</tr>
<tr>
<td>Network parameters</td>
<td></td>
</tr>
<tr>
<td>$Width$</td>
<td>(1000, 2000, 5000) km</td>
</tr>
<tr>
<td>Demand parameters</td>
<td></td>
</tr>
<tr>
<td>$\Lambda_r$</td>
<td>$\in [100, 150]; \in [50, 200]$ items/week</td>
</tr>
<tr>
<td>$CV$</td>
<td>(0.4, 0.7, 1)</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>97.5 %</td>
</tr>
<tr>
<td>$z_\alpha$</td>
<td>1.96</td>
</tr>
<tr>
<td>Cost parameters</td>
<td></td>
</tr>
<tr>
<td>$F_d$</td>
<td>(100, 200, 400) €/week</td>
</tr>
<tr>
<td>$K_d$</td>
<td>(1000, 2500) €/order</td>
</tr>
<tr>
<td>$H_d = H_r$</td>
<td>0.25 €/item/week</td>
</tr>
</tbody>
</table>

Table 1: Parameter values used for the numerical experiments.
DCs are calculated using the Euclidean norm. Transportation costs (computed as a cost per vehicle) are proportional to these distances, considering a cost of 1 €/km · vehicle. An average speed of 50 km/h is supposed to compute the retailer’s lead times (between DCs and retailers). We assume that the order lead times (from central plant to DCs) are the same for all DCs, and are equal to the average lead time from all potential DCs to all retailers. In order to choose a realistic value for the capacities, we set the vehicle’s capacity to 75 % of the largest shipment size obtained when no capacity is accounted (i.e., using the right-hand term in Equation (2)).

Overall, the parameter values lead to $3 \cdot 3 \cdot 2 \cdot 3 \cdot 2 \cdot 3 \cdot 2 = 324$ parameter combinations (see Table 1). For each parameter combination, we generate 5 maps in which retailer’s expected demands and locations are randomly varied within their respective boundaries, resulting in 1620 experiments. Both the heuristic and the CQMIP are implemented in GAMS and solved in CPLEX on a 2.70 GHz computer with 16 GB of RAM. The computational time per instance of the CQMIP is limited to half an hour for the cases with 50 retailers, and to one hour for the cases with 100 retailers.

Table 2 shows the gap between the solutions given by the heuristic algorithm and by the conic quadratic formulation for 100 retailers, computed as $100 \cdot (Sol_{Heur} - Sol_{CQMIP}/Sol_{CQMIP})$. This gap is, on average, 0.32 %. Our heuristic algorithm takes an average time of 136 seconds to solve one instance. As we limit the computational time to one hour, many instances are not solved to optimality by the CQMIP. The optimality gap of the CQMIP is, on average, 2.75 %. The heuristic algorithm leads to better solutions than the CQMIP in 197 cases (out of 540). It is interesting to observe that the coefficient of variation (CV) does not have a clear impact on the heuristic results. The results on Table 2 also show that the accuracy of our heuristic deteriorates when a large number of DCs is opened. For larger values of Width, and lower values of facility opening ($F_d$) and ordering ($K_d$) costs, the gap tends to increase with the number of opened DCs.

To explore this tendency more deeply, Table 3 gives the heuristic accuracy as a function of the number of opened DCs, for 20, 50 and 100 retailers and potential DC locations. We observe that when the number of opened DCs is smaller than one tenth of the number of
<table>
<thead>
<tr>
<th>$\Lambda_r$</th>
<th>Width</th>
<th>CV</th>
<th>$F_d=100$</th>
<th>$F_d=200$</th>
<th>$F_d=400$</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$K_d=1000$</td>
<td>$K_d=2500$</td>
<td>$K_d=1000$</td>
<td>$K_d=2500$</td>
<td>$K_d=1000$</td>
<td>$K_d=2500$</td>
</tr>
<tr>
<td>0.40</td>
<td>0.07</td>
<td>-1.24</td>
<td>0.32</td>
<td>-1.97</td>
<td>0.49</td>
<td>0.25</td>
</tr>
<tr>
<td>4.46</td>
<td>2.84</td>
<td>-0.51</td>
<td>4.05</td>
<td>2.63</td>
<td>3.03</td>
<td>1.62</td>
</tr>
<tr>
<td>0.29</td>
<td>-0.31</td>
<td>-0.23</td>
<td>0.84</td>
<td>-0.10</td>
<td>-0.10</td>
<td></td>
</tr>
<tr>
<td>5.27</td>
<td>3.04</td>
<td>0.22</td>
<td>0.71</td>
<td>-0.11</td>
<td>0.37</td>
<td>0.59</td>
</tr>
<tr>
<td>0.04</td>
<td>4.86</td>
<td>3.84</td>
<td>6.05</td>
<td>3.04</td>
<td>2.84</td>
<td>3.03</td>
</tr>
<tr>
<td>1000</td>
<td>0.70</td>
<td>0.93</td>
<td>-0.77</td>
<td>0.05</td>
<td>-0.23</td>
<td>1.12</td>
</tr>
<tr>
<td>8.21</td>
<td>4.06</td>
<td>5.68</td>
<td>4.05</td>
<td>6.05</td>
<td>4.65</td>
<td>3.84</td>
</tr>
<tr>
<td>0.40</td>
<td>-1.09</td>
<td>0.55</td>
<td>-0.45</td>
<td>0.92</td>
<td>0.88</td>
<td>0.22</td>
</tr>
<tr>
<td>7.81</td>
<td>4.07</td>
<td>6.27</td>
<td>4.25</td>
<td>4.45</td>
<td>3.64</td>
<td>0.06</td>
</tr>
<tr>
<td>0.34</td>
<td>-0.41</td>
<td>-0.29</td>
<td>-0.09</td>
<td>0.38</td>
<td>0.22</td>
<td>0.03</td>
</tr>
<tr>
<td>1</td>
<td>8.01</td>
<td>5.86</td>
<td>6.48</td>
<td>4.05</td>
<td>5.46</td>
<td>3.40</td>
</tr>
<tr>
<td>100 - 150</td>
<td>2000</td>
<td>0.70</td>
<td>3.96</td>
<td>0.17</td>
<td>1.16</td>
<td>-0.79</td>
</tr>
<tr>
<td>22.2</td>
<td>7.61</td>
<td>10.42</td>
<td>6.01</td>
<td>7.29</td>
<td>4.85</td>
<td>0.74</td>
</tr>
<tr>
<td>3.64</td>
<td>-0.11</td>
<td>1.56</td>
<td>0.31</td>
<td>0.74</td>
<td>0.85</td>
<td>1.16</td>
</tr>
<tr>
<td>21.6</td>
<td>8.16</td>
<td>10.2</td>
<td>7.29</td>
<td>7.28</td>
<td>5.46</td>
<td>10.01</td>
</tr>
<tr>
<td>1</td>
<td>2.86</td>
<td>-1.00</td>
<td>0.79</td>
<td>0.46</td>
<td>1.04</td>
<td>0.40</td>
</tr>
<tr>
<td>18.4</td>
<td>8.01</td>
<td>10.6</td>
<td>7.61</td>
<td>7.68</td>
<td>4.87</td>
<td>9.51</td>
</tr>
<tr>
<td>5000</td>
<td>0.70</td>
<td>0.43</td>
<td>-0.91</td>
<td>0.43</td>
<td>-0.41</td>
<td>0.43</td>
</tr>
<tr>
<td>7.81</td>
<td>4.86</td>
<td>5.47</td>
<td>4.25</td>
<td>4.05</td>
<td>3.63</td>
<td>5.06</td>
</tr>
<tr>
<td>0.09</td>
<td>-1.29</td>
<td>0.48</td>
<td>-0.43</td>
<td>0.69</td>
<td>0.08</td>
<td>-0.06</td>
</tr>
<tr>
<td>8.21</td>
<td>4.06</td>
<td>6.48</td>
<td>4.26</td>
<td>4.45</td>
<td>3.43</td>
<td>5.16</td>
</tr>
<tr>
<td>0.10</td>
<td>-0.09</td>
<td>0.25</td>
<td>-0.24</td>
<td>0.39</td>
<td>0.11</td>
<td>0.09</td>
</tr>
<tr>
<td>1</td>
<td>8.21</td>
<td>5.46</td>
<td>6.48</td>
<td>4.25</td>
<td>4.45</td>
<td>3.44</td>
</tr>
<tr>
<td>50 - 200</td>
<td>2000</td>
<td>0.40</td>
<td>0.56</td>
<td>0.70</td>
<td>0.75</td>
<td>-0.45</td>
</tr>
<tr>
<td>28.8</td>
<td>8.61</td>
<td>10.81</td>
<td>6.29</td>
<td>8.45</td>
<td>6.06</td>
<td>11.31</td>
</tr>
<tr>
<td>2.97</td>
<td>-1.40</td>
<td>1.09</td>
<td>-0.08</td>
<td>0.94</td>
<td>-0.10</td>
<td>0.57</td>
</tr>
<tr>
<td>24.2</td>
<td>8.41</td>
<td>11.41</td>
<td>7.09</td>
<td>7.48</td>
<td>5.27</td>
<td>10.61</td>
</tr>
<tr>
<td>0.88</td>
<td>-0.85</td>
<td>1.18</td>
<td>-0.21</td>
<td>0.98</td>
<td>0.32</td>
<td>0.72</td>
</tr>
<tr>
<td>1</td>
<td>22.0</td>
<td>7.41</td>
<td>11.81</td>
<td>8.01</td>
<td>8.09</td>
<td>5.47</td>
</tr>
<tr>
<td></td>
<td>Avg.</td>
<td></td>
<td>1.31</td>
<td>-0.61</td>
<td>0.53</td>
<td>-0.29</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>11.9</td>
<td>5.37</td>
<td>7.08</td>
<td>4.76</td>
</tr>
</tbody>
</table>

Table 2: Average gaps (in percentage) between heuristic and CQMIP solutions when the demands, the map widths, the demand coefficients of variation, the facility opening costs and the order costs are varied, for 100 retailers. Below each gap, we include the average number of opened DCs by our heuristic and by the CQMIP, respectively. The values in the table are averages over 5 maps.

retailers (< 10%$n_r$, i.e., when DCs serve on average 10 or more retailers) the gap between our heuristic and the CQMIP solutions is 0.17% (0.49% for < 15%$n_r$, i.e., around 7 or more retailers served per DC). An important conclusion is thus that our heuristic is particularly accurate and efficient when the number of opened DCs is not too large in proportion of the number of retailers. Moreover, we observe in Table 3 that increasing the number of retailers (and potential DC locations) deteriorates the performance of the CQMIP.
Table 3: Average gap between the heuristic and the CQMIP solution as a function of the number of opened DCs (in percentage of the number of retailers), average optimality gap of the CQMIP (Opt), and average computational time. They are given for $n_r = 20$, $50$ or $100$ retailers, as well as their average.

<table>
<thead>
<tr>
<th>$n_r$</th>
<th>0-5% $n_r$</th>
<th>5-10% $n_r$</th>
<th>10-15% $n_r$</th>
<th>&gt;15% $n_r$</th>
<th>Opt</th>
<th>Heuristic</th>
<th>CQMIP</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>0.05</td>
<td>0.29</td>
<td>2.13</td>
<td>8.32</td>
<td>0.06</td>
<td>36</td>
<td>256</td>
</tr>
<tr>
<td>50</td>
<td>0.10</td>
<td>0.40</td>
<td>0.93</td>
<td>6.24</td>
<td>1.83</td>
<td>81</td>
<td>1746</td>
</tr>
<tr>
<td>100</td>
<td>0.07</td>
<td>0.11</td>
<td>0.37</td>
<td>3.46</td>
<td>2.75</td>
<td>137</td>
<td>3607</td>
</tr>
<tr>
<td>Avg.</td>
<td>0.07</td>
<td>0.26</td>
<td>1.14</td>
<td>6.01</td>
<td>1.55</td>
<td>85</td>
<td>1870</td>
</tr>
</tbody>
</table>

in terms of accuracy and computational time. The accuracy of the heuristic is not affected significantly when the number of retailers is increased, and its computational time increases less steeply. Another interesting insight can be inferred from the particular cases in which the CQMIP reaches the optimal solution. The CQMIP finds the optimal solution in 671 (out of 1620) instances, in an average time of 888 seconds. For these instances, the gap between the heuristic and the optimal solution is 1.36 %, and our heuristic algorithm finishes in 59 seconds on average. Overall, the experimental results suggest that our heuristic algorithm solves the location-inventory problem efficiently.

A second set of experiments is run to show the computational time of our heuristic when designing large supply chain networks, with 200, 300, 500, 700 and 1000 retailers (and potential DC locations). In these cases, the conic quadratic model can not be solved in reasonable computational time. The problem configurations are chosen as a subset of those used in the first set of experiments (for a total of 60 experiments). Retailers are located in a square of 1000 km or 2000 km width. Demands are assigned between 100 and 150 products per week and the coefficients of variation are 0.50 or 1. The opening facility costs are 100, 200 or 400 € and the order cost is 1000 € per week.

Table 4 shows how the computational time increases with the problem size. We give it for three versions of the heuristic, where $H_1$ is the default version which has been used to solve the first set of experiments. Firstly, we see that our heuristic can solve larger instances in reasonable computational time. It solves cases with 500 retailers in one hour, and 1000 retailers in around 5 hours. Depending on the objective at hand, a balance between computational
time and solution accuracy can be made in order to find the best setting of the heuristic algorithm. We illustrate this with two alternative, simplified, versions of the heuristic, noted $H_2$ and $H_3$. These versions are made quicker by reducing $\text{MaxNoImprov}$ (number of iterations without a new best solution) and $\text{MaxMS}$ (number of multiple starts), but they consequently lose in accuracy. In particular, $H_2$ uses $\text{MaxMS} = 9$ and $\text{MaxNoImprov} = 7$, while $H_3$ takes $\text{MaxMS} = 6$ and $\text{MaxNoImprov} = 5$. We see on Table 4 that the version $H_2$ decreases the computational time with respect to $H_1$ and has a limited impact on the accuracy. On the other hand, $H_3$ reduces significantly the time but deteriorates significantly the accuracy compared to $H_1$ and $H_2$.

5.3. Managerial Insights

In this subsection, we derive managerial insights from the experiments introduced in the previous subsection. A first insight can be inferred observing the relation between the different costs included in the model. Table 5 shows the number of distribution centers (DCs) opened and the breakdown of costs, for different values of the unit holding costs ($H_r, H_d$), the map width, the unit facility opening cost ($F_d$) and the unit ordering cost ($K_d$). It presents the total facility opening costs ($F$) computed from Equation (10), the total transportation costs ($T$) from Equation (8), the total inventory and ordering costs (IO) from Equations (9) and (11), and the total safety stock costs (SS) from Equations (12) and (13). To begin with, we see that when the width of the map increases, the supply chain design is adapted to avoid a large increase of the transportation ($T$) and safety stock

<table>
<thead>
<tr>
<th>$n_r$</th>
<th>Time (sec.)</th>
<th>$H_1$</th>
<th>$H_2$</th>
<th>$H_3$</th>
<th>Gap (%)</th>
<th>$H_1$</th>
<th>$H_2$</th>
<th>$H_3$</th>
<th>#DCs</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>6421</td>
<td>4499</td>
<td>2363</td>
<td></td>
<td>0.14</td>
<td>13.65</td>
<td>13.65</td>
<td>12.23</td>
<td></td>
</tr>
<tr>
<td>300</td>
<td>19772</td>
<td>13514</td>
<td>7094</td>
<td></td>
<td>0.17</td>
<td>20.50</td>
<td>20.42</td>
<td>18.17</td>
<td></td>
</tr>
<tr>
<td>500</td>
<td>3157</td>
<td>2285</td>
<td>1196</td>
<td></td>
<td>-0.03</td>
<td>13.25</td>
<td>13.50</td>
<td>12.83</td>
<td></td>
</tr>
<tr>
<td>700</td>
<td>8125</td>
<td>5926</td>
<td>3112</td>
<td></td>
<td>0.29</td>
<td>16.67</td>
<td>16.25</td>
<td>13.92</td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td>274</td>
<td>205</td>
<td>112</td>
<td></td>
<td>0.20</td>
<td>8.00</td>
<td>7.58</td>
<td>7.33</td>
<td></td>
</tr>
<tr>
<td></td>
<td>274</td>
<td>205</td>
<td>112</td>
<td></td>
<td>0.20</td>
<td>8.00</td>
<td>7.58</td>
<td>7.33</td>
<td></td>
</tr>
<tr>
<td></td>
<td>776</td>
<td>565</td>
<td>300</td>
<td></td>
<td>0.05</td>
<td>9.83</td>
<td>10.50</td>
<td>8.92</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3157</td>
<td>2285</td>
<td>1196</td>
<td></td>
<td>-0.03</td>
<td>13.25</td>
<td>13.50</td>
<td>12.83</td>
<td></td>
</tr>
<tr>
<td></td>
<td>8125</td>
<td>5926</td>
<td>3112</td>
<td></td>
<td>0.29</td>
<td>16.67</td>
<td>16.25</td>
<td>13.92</td>
<td></td>
</tr>
<tr>
<td>Avg.</td>
<td>6421</td>
<td>4499</td>
<td>2363</td>
<td></td>
<td>0.14</td>
<td>13.65</td>
<td>13.65</td>
<td>12.23</td>
<td></td>
</tr>
</tbody>
</table>

Table 4: Average computational time, average gap compared to $H_1$, and average number of opened DCs.
Table 5: Number of opened DCs (#DCs), total facility opening (F), transportation (T), inventory and ordering (IO) and safety stock (SS) costs for the cases with 100 retailers. Below each cost, it is also expressed as a percentage of the total cost. The values in the table are averages over 30 instances (5 maps for each value of $\Lambda_r$, $CV$).

(SS) costs, which are proportional to distances (and lead times). More DCs are opened to reduce the distances between DCs and retailers, and the shipment size is increased in order to decrease the transportation frequency. This leads to increasing the total facility opening (F), inventory and ordering (IO) costs. Then, we observe in Table 5 that when the unit ordering cost ($K_d$) is increased, the supply chain design adapts to prevent the total inventory order cost (IO) to rise out of proportion. The number of opened DCs is decreased so that the orders to the central plant can be aggregated and thus be less frequent. This decreases the total facility opening costs (F) and limits the inventory and ordering costs (IO), but increases the distances between DCs and retailers, and consequently the transportation (T).
### Table 6: Number of opened DCs (#DCs) and total safety stock, in number of products, held at retailers (SSRet.) and at DCs (SSDCs) when the map width, facility opening cost and coefficient of variation are varied, for the instances with 100 retailers. The values in the table are averages over 20 instances (5 maps for each value of $\Lambda_r, K_d$).

<table>
<thead>
<tr>
<th>Width</th>
<th>$F_d$</th>
<th>$CV=0.4$</th>
<th>$CV=0.7$</th>
<th>$CV=1$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>#DCs</td>
<td>SSRet.</td>
<td>SSDCs</td>
</tr>
<tr>
<td>1000</td>
<td>100</td>
<td>3.75</td>
<td>2938</td>
<td>957</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>3.35</td>
<td>3033</td>
<td>906</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>2.50</td>
<td>3373</td>
<td>808</td>
</tr>
<tr>
<td>2000</td>
<td>100</td>
<td>6.20</td>
<td>3540</td>
<td>1716</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>4.80</td>
<td>3838</td>
<td>1547</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>4.00</td>
<td>4044</td>
<td>1430</td>
</tr>
<tr>
<td>5000</td>
<td>100</td>
<td>16.60</td>
<td>4279</td>
<td>4084</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>8.35</td>
<td>5100</td>
<td>3209</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>6.60</td>
<td>5392</td>
<td>2885</td>
</tr>
</tbody>
</table>

Table 6: Number of opened DCs (#DCs) and total safety stock, in number of products, held at retailers (SSRet.) and at DCs (SSDCs) when the map width, facility opening cost and coefficient of variation are varied, for the instances with 100 retailers. The values in the table are averages over 20 instances (5 maps for each value of $\Lambda_r, K_d$).

and safety stock (SS) costs. Similarly, as seen in Table 5, when the unit facility opening cost ($F_d$) increases, the number of opened DCs decreases and the supply chain configuration is adapted (e.g., IO costs decrease). Furthermore, we observe that when the unit holding costs ($H_r, H_d$) double, the shipment sizes are reduced to avoid the total inventory and ordering cost (IO) to increase in the same proportion than the unit holding cost. This leads to an increase of the transportation frequency and thus of the transportation costs (T). From all these observations, we see that the various costs are deeply interrelated, and that they balance each other out. This illustrates the need of integrated models in supply chain network design.

A second interesting insight regards the impact of demand uncertainty on the supply chain configuration, and risk pooling in particular. Table 6 shows how the number of opened DCs and the safety stock are affected when parameters such as the map width, the unit facility opening cost ($F_d$) and the coefficient of variation of demand ($CV$) are varied. We see that when the unit facility opening cost ($F_d$) increases, the number of opened DCs decreases, and so does the total safety stock held at DCs. The total safety stock at DCs decreases while the overall uncertainty, coming from all the retailers, is the same. This illustrates the risk pooling effect: demand variability is reduced at DCs when demand is aggregated from more retailers. A lower number of opened DCs allows to pool the demand uncertainty and decreases the need of safety stock at DCs to handle this demand uncertainty (while keeping...
the same service level). However, another interesting perspective arises when considering the safety stock at retailers. In Table 6, when the unit opening cost \( F_d \) increases, the number of opened DCs and the total safety stock at DCs decreases, but the total safety stock at retailers increases. As the number of opened DCs reduces, the distances and lead times between DCs and retailers increase, leading to an increase of the safety stock level at retailers. Thus, when considering safety stock at retailers, we observe that the risk pooling benefits at DCs tend to be balanced by the safety stock increase at retailers. If the supply chain is not adapted when using pooling strategies (for example, by increasing the vehicles’ speed), the total safety stock of the global supply chain will be less reduced than expected, if not increased. This trend can also be seen in Table 6 from the impact of the coefficient of variation \( CV \). When the variability of the demand increases, the number of opened DCs often increases. Strategies focusing on risk pooling suggest to aggregate demand in fewer DCs when demand variability increases. However, when considering safety stocks at retailers, the proximity to retailers (and short lead times) might actually be more important, leading to a larger number of opened DCs. In Figure 1, we illustrate the trade-offs between the safety stock costs at retailers and at DCs, by showing how they evolve when the number of opened DCs increases (by changing the unit facility opening cost \( F_d \)), for different coefficients of variation \( CV \) and holding costs at retailers \( H_r \). First, we clearly see that risk pooling
leads to decrease the total safety stock at DCs when the number of opened DCs decreases. However, it is interesting to observe that the total safety stock cost is not decreased when reducing the number of opened DCs, as the safety stocks at retailers increase (due to larger distances).

6. Conclusions

This paper presents an approach to design large supply chain networks while integrating inventory decisions and demand uncertainty. We propose a continuous non-linear formulation for addressing the location-inventory problem, integrating realistic features such as shipment size, transportation cost per vehicle or safety stocks. We show that the model simplifies to a linear program when the DC flow and the standard deviation at DCs are fixed. Thus, we propose an iterative heuristic algorithm that estimates these variables, solves the resulting linear program and uses the solution to iteratively improve the variables estimations. In order to show the effectiveness of our heuristic algorithm, we perform extensive computational experiments to compare our heuristic results with those of a CQMIP reformulation of the problem. For instances with 100 retailers, our heuristic takes a bit more than 2 minutes while the CQMIP takes one hour, and the average gap is 0.32%. We also show that our heuristic performs better when the number of opened DCs is not too large: the gap is 0.09% when a DC serves, on average, at least 10 retailers. An important feature of our heuristic is that it can be used to design larger supply chain networks in reasonable computational time. For example, we solve instances with 1000 retailers (and 1000 potential DC locations) in around 5 hours. Finally, we discuss managerial insights that can be inferred from the computational experiments. In particular, we study the interaction between the various costs, showing that they are deeply interrelated and that integrating location and inventory decisions in the same model is indeed justified. We also analyze how demand uncertainty impacts the design of the supply chain and show that, when looking at the global supply chain design, risk pooling benefits at DCs may be mitigated by the increase of safety stocks at retailers. In the future, this research could be extended by adding other features to our location-inventory model. Features such as multiple modes of transportation,
capacity constraints at retailers and DCs, or multiple products could be integrated in our model. Our approach could also be adapted to design three echelon supply chains, allowing multiple production sites.

Acknowledgments

This research was supported by the Belgian Foundation of Research Science (FNRS) through the FRESH grant.

References


