



Supply chain design under quality disruptions and tainted materials delivery



AliReza Madadi ^{a,*}, Mary E. Kurz ^{b,1}, Scott J. Mason ^{b,2}, Kevin M. Taaffe ^{b,3}

^a Southwest Airlines Co., P.O. Box 36611, 2702 Love Field Drive, Dallas, TX 75235, United States

^b Department of Industrial Engineering, Clemson University, 110 Freeman Hall, Clemson, SC 29634, United States

ARTICLE INFO

Article history:

Received 27 January 2013

Received in revised form 15 March 2014

Accepted 16 April 2014

Keywords:

Supply chain design

Supply disruptions

Scenario-based optimization

Metaheuristics

ABSTRACT

Events such as the 2008 Heparin tragedy, in which patients lost their lives due to tainted pharmaceuticals, highlight the necessity for supply chain designers and planners to consider the risk of even low probability incidents in supply chains. The goal of this research is to design a single-period, single-product supply chain model with capacitated facilities to hedge against the possibility of sending tainted materials to consumers. Given that our mixed-integer stochastic model is NP-hard, we develop efficient heuristic and metaheuristic algorithms to obtain acceptable solutions. Computational experience is presented and discussed.

© 2014 Elsevier Ltd. All rights reserved.

1. Introduction

Some supply disruptions are not only costly, but may have catastrophic consequences in spite of their low probability of occurrence. Consider a supplier that begins sending tainted or contaminated products without awareness and the product reaches the final consumers. This may cause late delivery or product shortage and in some supply chains, such as healthcare, this disruption may put final consumers i.e., patients' lives in danger. As an example, the disruption of a flu vaccine manufacturer in Bristol, UK in 2004 resulted in disastrous consequences. The UK government stopped production when U.S. regulators inspected a manufacturing plant and found evidence of bacterial contamination problems. This reduced the US's supply of the vaccine by nearly 50% during the 2004–2005 flu season (Everett and Baker, 2004). A healthcare supply chain is also very susceptible to disruptions caused by contamination. Heparin, a widely-used blood-thinning medicine that is made from pig intestines, was contaminated by an undetected outbreak of blue ear pig disease in China in 2008. This led to 81 patient deaths as well as hundreds of allergic reactions in the United States (Usdin, 2009). The investigation engaged several government agencies, university researchers and a biotech company with a generic heparin under FDA review. Although no one at the time knew what was causing the reactions, members of Congress concluded that the issue was the result of "regulatory failure" (Usdin, 2009). In another supply chain disruption, a baby food producer who purchased vitamin supplements from a Chinese supplier found out that the supplements were contaminated by cement (Lyles et al.,

* Corresponding author. Tel.: +1 864 633 6143.

E-mail addresses: amadadi@g.clemson.edu (A. Madadi), mkurz@clemson.edu (M.E. Kurz), mason@clemson.edu (S.J. Mason), taaffe@clemson.edu (K.M. Taaffe).

¹ Tel.: +1 864 656 4652.

² Tel.: +1 864 656 5645.

³ Tel.: +1 864 656 0291.

2008). This incident involved 22 Chinese and 10 global manufacturers and led to kidney problems and kidney stones in Chinese babies, illustrating the result of poor or failed inspections by FDA or production facilities (Schoder, 2010).

A 2008 US Government Accountability Office (GAO) report indicated that in 2007, the FDA inspected approximately 8% of foreign facilities and declared that, at that rate, it would take 13 years to inspect all such facilities (Benson, 2012). On the other hand, inspection of raw materials is usually a significant portion of manufacturing costs whereas manufacturing facilities are searching for cost-reduction opportunities (Green and Brown, 2009). As a result, while consumers may assume all materials are inspected, some are either not inspected or are inadequately inspected.

These types of incidents indicate not all materials entering the supply chain can be inspected by regulatory agencies and accentuate the need to consider risk of receiving contaminated materials in the design and planning stages. It is also vital to contemplate risk of supply disruptions when designing a supply chain so that it can be responsive and resilient in the case of an unplanned incident. However, managers, deceived by the small likelihood of these types of incidents, often tend to underestimate the impact of such mishaps.

The goal of this research is to design a single-period, single-product supply chain model with capacitated facilities to hedge against the risk of supply quality disruptions and sending tainted materials to consumers. Inspired by the aforementioned incidents, we reduce (but do not completely eliminate) tainted materials by introducing producer-implemented inspections. We also consider the loss of all or a significant fraction of the expected supply quantity due to contamination or discarding tainted materials after implementation of inspection. In all cases, we assume some tainted materials are also shipped to consumers. We, therefore, model the risk of shipping tainted materials with a penalty cost. In cases where reducing tainted materials is of the first priority, using a high penalty cost will cause the model to avoid it as much as possible. When we inspect, we also incur costs related to inspection and the disposal of discovered tainted materials.

Typically, the decision making process dealing with supply chain disruptions involves both strategic and tactical considerations (Drezner and Hamacher, 2004). Strategic decisions comprise decisions such as choosing which markets to serve, from which suppliers to source, the location of facilities/suppliers, and how many suppliers to use. Tactical decisions include decisions such as inventory management production planning (Drezner and Hamacher, 2004; Chopra and Sodhi, 2004).

Our model examines the strategic decision of facility selection and the tactical decision of capacity allocation among facilities. Additionally, we consider the implementation of inspection at a facility as a tactical decision. This aspect of the work was inspired by tragedies such as the heparin. If the risk of shipping tainted materials can be minimized prior to such tragedies, producers can decrease liability and improve consumer safety. Insights into how our model should be configured to reduce the risk of tainted materials reaching consumers are of interest to several types of supply chains such as healthcare, pharmaceutical, cosmetic and beauty, and food or dairy industries.

The objective of the model is to minimize the expected overall cost which is composed of the cost of selecting the facility, shipping untainted materials, shipping tainted materials, inspecting the facility, and discarding tainted materials. We formulate this problem using a two-stage stochastic mixed-integer problem. A number of researchers address various solutions of two-stage stochastic problems in the literature such as Bender's Decomposition (MirHassani et al., 2000), Lagrangian relaxation (Daskin et al., 2002), or L-shaped methods (Laporte et al., 1994). However, while this approach may allow for exact solutions in some situations, it can be very challenging to draw concrete analytical insights from such models and to obtain good solutions for large instances within a limited time frame (see Santoso et al., 2005) since the problem is a special case of the two-stage stochastic capacitated facility location problem which is NP-hard (Doerner et al., 2007; Shapiro, 2008). Based on our experience in solving various size problems using commercial software in this paper, we show that the number of facilities used have a significant impact on the solution time. As a result, we develop several heuristics and a metaheuristic approach i.e., simulated annealing, to efficiently solve and handle large size problems.

The paper is organized as follows. In Section 2, the problem description and the mathematical formulation are discussed. In Sections 3 and 4, the solution procedure and data generation method are presented, respectively. Computational results are discussed in Section 5. Finally, Section 6 presents conclusions and future extensions.

2. Problem description and mathematical formulation

2.1. Description

The earliest work in supply chain design was developed by Geoffrion and Graves (1974). They introduced a multi-commodity logistics network design model for optimizing finished product flows from plants to distribution centers to final consumers. Based on this work, a large number of optimization-based approaches have been proposed for the design of supply chain networks. These works have resulted in significant improvements in the modeling of these problems as well as in algorithmic and computational efficiency but they all have assumed that the design parameters for the supply chain are deterministic (Drezner and Hamacher, 2004; Vidal and Goetschalckx, 1997; Demirtas and Üstün, 2008; Brandeau and Chiu, 1989; Caserta and Rico, 2008; Hinojosa et al., 2008). Unfortunately, critical parameters such as consumer demand, supply capacity are generally uncertain. Therefore, traditional deterministic optimization is not suitable for capturing the behavior of the real-world problem.

The significance of uncertainty has encouraged a number of researchers to address stochastic parameters in their research. Most of the stochastic approaches for supply chain design only consider tactical level decisions usually related

to demand uncertainty (see Agrawal and Seshadri, 2000) while supply uncertainty is often ignored and supply capacity assumed to be unlimited (see Owen and Daskin, 1998 for more details). For instance, Santos et al. (2005) utilized a stochastic programming approach for addressing demand uncertainty in supply chain design. Tsiakis et al. (2001) and Alonso-Ayuso et al. (2003) also considered a two-stage stochastic programming model for supply chain design under demand uncertainty. Qi and Shen (2007) introduced an integrated supply chain design model that with unreliable supply where capacity is infinite. Shen et al. (2008) addressed a facility location problem in which some uncapacitated facilities are subject to failure.

In Fig. 1 we provide a hypothetical supply chain with an initial assignment of consumers to facilities, as well as suppliers to facilities. Three facilities were selected and the capacity was sufficient to fulfill all the demand of all the consumers.

In some cases, facilities can ship poor quality items to consumers due to tainted raw material received from suppliers. In Fig. 2, a scenario with product quality disruptions at two facilities is presented.

Once inspection is implemented in a facility, a portion of tainted items is discarded and fewer tainted items are delivered to the consumers. However, discarding tainted items might result in consumer demand being unsatisfied due to insufficient capacity. Then, the unmet demand can be fulfilled by adding another facility at a cost, as illustrated in Fig. 3.

2.2. Mathematical model

We utilize a mixed-integer stochastic programming model that is formulated as a two-stage optimization problem. The selection of the facilities is considered at the first stage and modeled as a binary decision. The second-stage decision variables include the binary inspection decisions at each facility as well as the actual capacity allocation, the penalty cost shipping of contaminated or tainted materials to manufacturers due to quality problems which cause disruptions at all or some of facilities, and finally cost of discarding tainted materials after implementation of inspection.

To deal with the uncertainty in the second stage, a scenario-based modeling approach is proposed, which has been widely used in stochastic programming problems (Shapiro, 2008; Alonso-Ayuso et al., 2003). Let scenario s denote an event where all, some or none of the facilities “fail” (i.e., have a quality disruption). Then ρ_s can represent the probability of occurrence for scenario s . We also provide an example of the role of inspection, using notation that is presented in our formulation. Suppose that under a particular scenario s , the extent of tainting at facility l is given by $q_{ls} = 0.20$ and this fraction can be reduced to $r_{ls} = 0.05$ after implementation of inspection. This means that for every 100 items produced at facility l , $100q_{ls} = 20$ are tainted. If no inspection is used, these 20 tainted items will be shipped to consumers. If inspection is used, 15 of these 20 tainted items will be detected and discarded (which indicates the loss of supply quantity), while $k_{lcs} = 100r_{ls} = 5$ of them will remain undetected and be shipped to consumers.

Given a finite set of scenarios, with associated probability ρ_s for scenario s , we can present a deterministic equivalent of the Supply Chain Design (SCD) model. We first summarize the complete notation for the SCD.

Sets

- C The set of consumers, indexed by c
- L The set of candidate facilities, indexed by l
- S The set of realized scenarios, indexed by s

Parameters

- f_l Fixed cost of selecting facility l
- κ_l Capacity of facility l
- b_c Total demand of consumer c
- n_l Fixed cost of implementing an inspection at candidate facility l
- q_{ls} Fraction of tainted materials produced at facility l in scenario s
- r_{ls} Fraction of tainted materials produced at facility l after inspection (we assume $q_{ls} > r_{ls}$) in scenario s
- λ_{lc} Unit cost of shipping untainted material from facility l to consumer c
- o_{lc} Unit penalty cost for shipping tainted material from facility l to consumer c
- γ_{lc} Unit cost of discarding tainted material (after inspection) at facility l originally destined for consumer c
- ρ_s Probability of occurrence for scenario s

Decision Variables

- $x_l = \begin{cases} 1, & \text{if facility } l \text{ is selected,} \\ 0, & \text{otherwise} \end{cases}$
- $z_{ls} = \begin{cases} 1, & \text{if inspection is implemented at facility } l \text{ in scenario } s, \\ 0, & \text{otherwise} \end{cases}$
- p_{lcs} Amount of materials shipped from facility l to consumer c in scenario s
- k_{lcs} Amount of tainted materials produced at facility l intended to be shipped to consumer c in scenario s
- d_{lcs} Amount of tainted materials discarded at facility l intended to be shipped to consumer c after inspection in scenario s

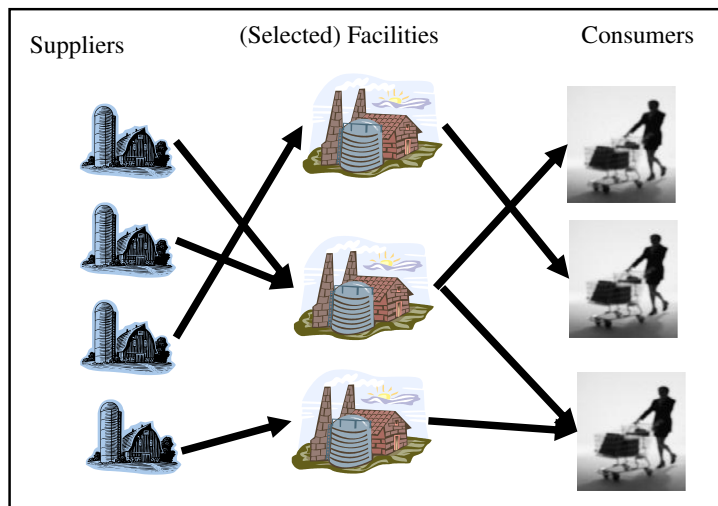


Fig. 1. Initial demand allocation (before disruption).

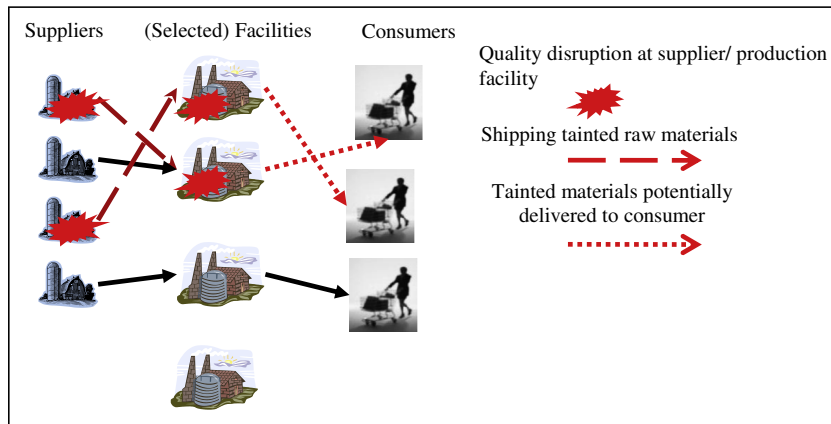


Fig. 2. Shipping tainted materials to consumers after disruptions (no inspection).

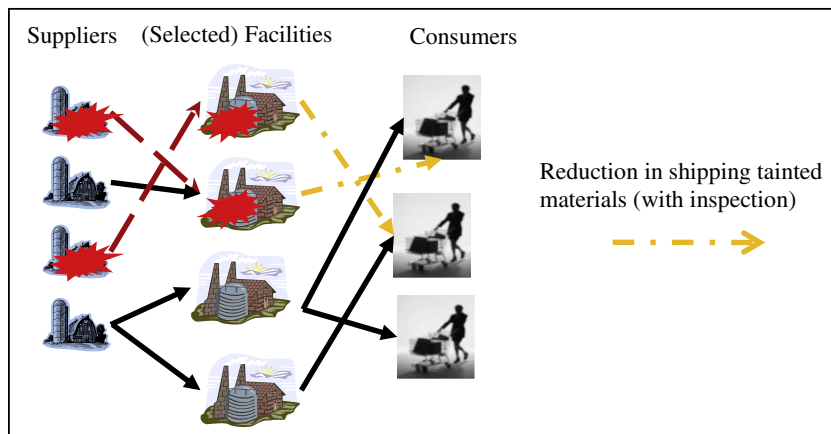


Fig. 3. Reduction of delivered tainted materials (after inspection).

We now present the deterministic equivalent of the formulation as:

$$[\text{SCD}] \quad \min \sum_{l \in L} x_l f_l + \sum_{s \in S} \rho_s \left(\sum_{l \in L} \sum_{c \in C} \lambda_{lc} [(1 - q_{ls}) p_{lcs}] + \sum_{l \in L} \sum_{c \in C} o_{lc} k_{lcs} + \sum_{l \in L} \sum_{c \in C} \gamma_{lc} d_{lcs} + \sum_{l \in L} n_l z_{ls} \right) \quad (1)$$

$$\text{s.t.} \quad \sum_{c \in C} [(1 - q_{ls}) p_{lcs} + k_{lcs} + d_{lcs}] \leq \kappa_l x_l, \quad \forall l \in L, s \in S \quad (2)$$

$$k_{lcs} + d_{lcs} = q_{ls} p_{lcs}, \quad \forall c \in C, l \in L, s \in S \quad (3)$$

$$k_{lcs} - r_{ls} p_{lcs} \leq M(1 - z_{ls}), \quad \forall c \in C, l \in L, s \in S \quad (4)$$

$$d_{lcs} - (q_{ls} - r_{ls}) p_{lcs} \leq M(1 - z_{ls}), \quad \forall c \in C, l \in L, s \in S \quad (5)$$

$$d_{lcs} \leq M(z_{ls}), \quad \forall c \in C, l \in L, s \in S \quad (6)$$

$$\sum_{l \in L} [(1 - q_{ls}) p_{lcs} + k_{lcs}] = b_c, \quad \forall c \in C, s \in S \quad (7)$$

$$z_{ls} \leq x_l, \quad \forall l \in L, s \in S \quad (8)$$

$$k_{lcs}, d_{lcs}, p_{lcs} \geq 0, \quad \forall c \in C, l \in L, s \in S \quad (9)$$

$$z_{ls} \in \{0, 1\}, \quad \forall l \in L, s \in S \quad (10)$$

$$x_l \in \{0, 1\}, \quad \forall l \in L \quad (11)$$

The objective function (1) in the first stage problem is the sum of fixed cost of selecting facilities. The second stage consists of four distinct terms. The first term ($\sum_{l \in L} \sum_{c \in C} \lambda_{lc} [(1 - q_{ls}) p_{lcs}]$) represents the expected transportation cost of shipping untainted materials. The second term ($\sum_{l \in L} \sum_{c \in C} o_{lc} k_{lcs}$) and the third term ($\sum_{l \in L} \sum_{c \in C} \gamma_{lc} d_{lcs}$) represent the penalty cost of supplying tainted materials for the consumers and the cost of discarding tainted materials, respectively. Finally, the last term ($\sum_{l \in L} n_l z_{ls}$) is the cost of inspection at all facilities.

Constraint set (2) requires a facility to be open if any portion of consumer demand is served from the facility. In addition, it ensures that the total consumer demand assigned to any facility cannot exceed the facility's capacity. Constraint sets (3)–(6) together calculate the amount of tainted materials that is shipped to the consumer. Hence, without inspection (when $z_{ls} = 0$), constraint set (6) implies that $d_{lcs} = 0$ and given constraint set (3), all of the tainted materials reach the consumer. However if inspection is implemented, constraint sets (4) and (5) imply that only materials passing inspection will be shipped to the consumer. The constant M is a big number and in our model we have used $M = \sum_{l \in L} f_l$. Constraint set (7) requires that the demand of every consumer is met. Constraint set (8) implies that inspection can be applied to only the selected set of facilities. Constraint set (9) requires that k_{lcs} , d_{lcs} and p_{lcs} are positive values, and constraint sets (10) and (11) place binary restrictions on variables z_{ls} and x_l .

3. Solution procedure

3.1. Constructive heuristics

We present a few constructive (greedy) heuristics in this section. In constructive algorithms, we start from an empty solution and construct a solution by assigning values to one decision variable at a time, until a complete solution is generated (Talbi, 2009). Constructive algorithms are popular techniques as they are simple to design and also they are good choices for large instances. However, the performance of constructive algorithms may be low as well. Therefore, we also develop improvement algorithms to improve the quality of the solution achieved by constructive algorithms. In our improvement algorithm, we start with a complete solution (i.e., a constructive algorithm solution) and transform it at each iteration using some search operators to hopefully find a better solution. We refer the interested reader to Sridharan (1995) and Cornuejols et al. (1991) for a comprehensive review of the heuristic approaches to solve these types of problems.

In our solution procedure, we first determine the set of selected facilities, x . We next determine z_{ls} by using one of the heuristics presented in this section. Let \bar{x}_l, \bar{z}_{ls} be the fixed and known values of x_l and z_{ls} , respectively. Eq. (1) reduces to a capacitated transportation problem, which is relatively easy to solve. We call this model SCD-Sub, and its formulation is stated as follows:

$$[\text{SCD-Sub}] \quad \min \sum_{s \in S} \rho_s \left(\sum_{l \in L} \sum_{c \in C} \lambda_{lc} [(1 - q_{ls}) p_{lcs}] + \sum_{l \in L} \sum_{c \in C} o_{lc} k_{lcs} + \sum_{l \in L} \sum_{c \in C} \gamma_{lc} d_{lcs} \right) \quad (12)$$

$$\text{s.t. } \sum_{c \in C} [(1 - q_{ls})p_{lcs} + k_{lcs} + d_{lcs}] \leq \kappa_l \bar{x}_l, \quad \forall l \in L, s \in S \quad (13)$$

$$k_{lcs} + d_{lcs} = q_{ls} p_{lcs}, \quad \forall c \in C, l \in L, s \in S \quad (14)$$

$$k_{lcs} - (r_{ls})p_{lcs} \leq M(1 - \bar{z}_{ls}), \quad \forall c \in C, l \in L, s \in S \quad (15)$$

$$d_{lcs} - (q_{ls} - r_{ls})p_{lcs} \leq M(1 - \bar{z}_{ls}), \quad \forall c \in C, l \in L, s \in S \quad (16)$$

$$d_{lcs} \leq M(\bar{z}_{ls}), \quad \forall c \in C, l \in L, s \in S \quad (17)$$

$$\sum_{l \in L} [(1 - q_{ls})p_{lcs} + k_{lcs}] = b_c, \quad \forall c \in C, s \in S \quad (18)$$

$$k_{lcs}, d_{lcs}, p_{lcs} \geq 0, \quad \forall c \in C, l \in L, s \in S \quad (19)$$

Note that k_{lcs} and d_{lcs} are auxiliary decision variables that depend solely on p_{lcs} and z_{ls} . Given the fact that z_{ls} is already determined and fixed and by using Eqs. (14)–(17), we can rewrite the SCD-Sub as follows:

$$\min \sum_{s \in S} \rho_s \left(\sum_{l \in L} \sum_{c \in C} \{ \lambda_{lc}(1 - q_{ls}) + o_{lc}q_{ls}(1 - \bar{z}_{ls}) + o_{lc}r_{ls}\bar{z}_{ls} + \gamma_{lc}(1 - \bar{z}_{ls})(q_{ls} - r_{ls}) \} p_{lcs} \right) \quad (20)$$

s.t. (13)–(19)

Our constructive heuristics operate in three stages. In stage one, we determine the set of selected facilities. In stage two, we determine z_{ls} , and finally in the last stage, we solve SCD-Sub in two phases. In the following, all three stages are presented.

3.1.1. Stage one methods

We develop three constructive heuristics to determine the set of selected facilities as follows:

- **Basic Greedy Heuristic (BGH):** One way to determine vector x is to simply open all the facilities. Therefore, we have $x_l = 1, \forall l \in L$. The computational complexity of BGH is $O(n)$ in the number of facilities.
- **Selective Greedy Heuristic (SGH):** In this method, we evaluate the selection cost of every candidate facility and we first select the facilities with the least total cost of selection until constraint (7) is satisfied. We start with an empty set for selected facilities. In Step 1, we calculate the expected available capacity of a facility. Assume that a facility operates with full capacity with probability χ_l and let $\bar{\kappa}_l$ denote the random variable for the available capacity where $p(\bar{\kappa}_l = \kappa_l) = \chi_l$ and $p(\bar{\kappa}_l = (1 - q_l)\kappa_l) = (1 - \chi_l)$. Therefore, the expected available capacity of facility l is calculated by $E(\bar{\kappa}_l) = \chi_l \kappa_l + (1 - \chi_l)(1 - q_l)\kappa_l$. We set χ_l equal to the probability of a facility producing perfect items. The total cost of selecting a facility is estimated in Step 2. The estimated total cost includes the fixed cost of selecting a facility plus the average of transportation cost of shipping untainted materials plus the average penalty cost of supplying tainted materials, and the average cost of discarding tainted materials. These average values are calculated over all customers. To select the facilities based on least total cost of selection, facilities are sorted in increasing order of total costs in Step 3. The computational complexity of SGH is $O(n \log n)$ in the number of facilities. Steps are illustrated in Fig. 4.
- **Capacity-Based Greedy Heuristic (CBGH):** In this method, we first start with an empty set of selected facilities. Then we choose a facility from the set of remaining candidates that reduces the total demand of the consumers ($\sum_{c \in C} b_c$) the most. Steps are illustrated in Fig. 5. The computational complexity of CBGH is $O(n \log n)$ in the number of facilities.

3.1.2. Stage two methods

Once the selected facilities have been set, we determine z_{ls} for each scenario. We develop the following construction heuristics:

- **Failed Scenario Inspection Heuristic (FSIH):** Consider S'_l for each l as the set of scenarios where facility l produce perfect items and $S''_l, \forall l \in L$ as the set of scenarios where facility l will produce tainted materials where $S'_l \cup S''_l = S$. We define z_{ls} as follows: $z_{ls} = \begin{cases} 1, & s \in S''_l \wedge x_l = 1, \forall l \in L, \forall s \in S \\ 0, & s \in S'_l \vee x_l = 0, \forall l \in L, \forall s \in S \end{cases}$.

The definition of z_{ls} means that we perform inspection for all facilities that are selected in stage one and produce tainted materials. The computational complexity of FSIH is $O(n)$ in the number of facilities.

- **Greedy Inspection Heuristic (GIH):** In this method, we define a desired shipping untainted level $\Delta (\Delta \in (0, 1])$, where $(\Delta)100\%$ of the shipping materials to consumers must be untainted. Let us start with an empty set for z . Given a scenario s where $s \in S'_l$, the following relation should be satisfied $\sum_{l \in L} [q_{ls}x_l\kappa_l(1 - z_{ls}) + r_{ls}x_l\kappa_l z_{ls}] \leq (1 - \Delta)\sum_{c \in C} b_c$. Otherwise, we

perform inspection until we reach the desired level of untainted materials. Note that, we consider facilities in decreasing order of maximum reduction in the fraction tainted materials or $\max \{(q_l - r_l) | \forall l \in L\}$. We consider $\Delta = 0.90$ in our computations. The steps are defined in Fig. 6, which is a $O(n \log n)$ algorithm for each scenario s .

- **Random Greedy Inspection Heuristic (RGIH):** The basic idea of this method is to estimate how much we can save by implementing inspection in a facility. We first evaluate ϕ_{ls} , the amount of savings for each individual facility. This is the ratio of how much savings can be obtained by preventing the shipment of tainted materials to the cost of implementing inspection in the facility. If the savings are high enough, then inspection for a facility will be implemented. The steps are defined in Fig. 7, which is $O(n \log n)$ in the number of facilities.

3.1.3. Stage three methods

In this stage we solve SCD-Sub in two phases. In the first phase, we solve the transportation problem between selected facilities and consumers. Initially, the unit transportation cost to the consumers (λ_{lc}) are ranked in an increasing order and then the transportation amounts i.e., p_{lcs} , are assigned considering the capacity limits of the selected facilities. In Fig. 8, a_c represents the demand of consumer c , and g_l represents the capacity for facility l . In the second phase, given the obtained value for p_{lcs} , we simply compute the values of auxiliary variables k_{lcs} and d_{lcs} . After p_{lcs} values are determined, and given the values of z_{ls} that are computed by one of the methods in stage two, we can calculate k_{lcs} and d_{lcs} by using Eqs. (14)–(17). Stage Three's method is $O(n \log n)$ in the number of facilities.

3.2. Improvement heuristics

In this section we develop improvement heuristics to improve the solution obtained from one of the heuristic methods presented above. First, we present hill-climbing heuristic (HC) that begins with a feasible solution and seeks to improve upon it. HC iteratively closes one facility if the facility is already selected and opens a facility if a facility is not selected. This iteration enables us to generate a new neighborhood and explore if the new set of selected facilities provides a cheaper solution or not. In order to maintain feasibility, only moves are allowed which provide enough capacity to satisfy the total demand of the consumers. The details of this heuristic are shown in Fig. 9. HC's time complexity depends on the use of FSIH, GIH or RGIH and the number of iterations performed.

In the second improvement algorithm, we apply a Variable Neighborhood Search (VNS) to change the flows from facilities to demands. The basic idea of VNS is to find a set of predefined neighborhoods to achieve a better solution. It explores either at random or deterministically a set of neighborhoods to get different local optima and to escape from local optima (for general pseudocode of VNS see (Glover and Kochenberger, 2003)). The purpose of VNS is to minimize transportation cost for each individual scenario, i.e., minimize $\sum_{l \in L} \sum_{c \in C} (\lambda_{lc} [(1 - q_{ls}) p_{lcs}])$ by changing p values.

The neighborhood strategy that we apply is structured by randomly choosing a point in the transportation cost matrix. The first step in our algorithm is the shaking step. At each iteration, an initial solution (p) is perturbed from the current neighborhood in order to generate a new solution (u). For this purpose we identify the closed path leading to that point which consists of horizontal and vertical lines as illustrated in Fig. 10. In order to generate a new solution, we move \hat{R} units from the chosen point and another point at a corner of the closed path and modify the remaining points at the other corners of the closed path to reflect this move. Note that \hat{R} is a random variable over the range of zero and the minimum value of the four selected points. Note that the selected point is shown by *. The implemented VNS for our problem is illustrated in Fig. 11.

3.3. Simulated annealing

Simulated Annealing (SA) is a metaheuristic approach inspired by nature. In this case, the process of a heated metal being cooled at a controlled rate (annealed) to improve its physical properties is simulated. The method was popularized by the work of Kirkpatrick et al. (1983) which continued the earlier work of Metropolis et al. (1953). The fundamental idea is to allow moves resulting in solutions of worse quality than the current solution in order to escape from local optima (Talbi, 2009). Fig. 12 outlines the implementation of SA in more detail.

3.3.1. Defining initial temperature and cooling schedule

An important consideration in SA is to set the initial value of the temperature T_0 . If the initial temperature is set very high, the search may be relatively close to a random local search. Otherwise, if the initial temperature is very low, the search might degenerate to an improving local search algorithm (Talbi, 2009) as the probability of accepting worse moves decreases too quickly. Another important factor is cooling schedule. The choice of a suitable cooling schedule is crucial for the performance of the algorithm. The cooling schedule defines the value of temperature T at every iteration.

The temperature T is decreased during the search process, thus at the beginning of the search the probability of accepting uphill moves is high and it gradually decreases. As stated, the choice of an appropriate cooling schedule and initial value of temperature are crucial for the performance of the algorithm. The cooling schedule defines the value of T at each iteration k , $T_{k+1} = R(T_k, k)$, where $R(T_k, k)$ is a function of the temperature at the previous step and of the iteration number. In this paper, we use one of the most common cooling schedule which follows a geometric law as $T_{k+1} = \theta T_k$, where $\theta \in (0, 1)$, which corresponds to an exponential decay of the temperature (Suman and Kumar, 2005). Furthermore, experience has shown that θ

should be between 0.5 and 0.99 (see Talbi, 2009). Hence, we considered four values, $\theta = 0.95, 0.90, 0.80$ and 0.75 ; and we obtained the best minimum regret in less computational time at $\theta = 0.75$.

An important consideration in SA is to set the initial value of the temperature T_0 properly. There is a tradeoff between a very high initial temperature and a lower one. The high temperature explores more of the solution space at the cost of increased running time. For this research, we use acceptance deviation methods. In this method, the starting temperature is computed by $t\sigma$ using preliminary experimentations on each data set, where σ represents the standard deviation of difference between values of objective functions and $t = -3/\ln(\bar{\omega})$ with the acceptance probability of $\bar{\omega}$, which is greater than 3σ (see Talbi, 2009). Finally, a sufficient number of iterations at each temperature should be performed. If too few iterations are performed at each temperature, the algorithm may not be able to reach the global optimum. Given the presented formula and after several experiments, we set the value for the initial temperature as $T_0 = 8000$.

3.3.2. Neighborhood selection

The manner in which a metaheuristic technique moves from one solution to its neighbor is a critical component. In our SA algorithm the neighborhood is defined by using one of the following neighborhood structures: (1) swapping one randomly selected facility with another randomly selected facility (SA-swap), (2) selecting one more facility (SA-add), (3) closing one selected facility (SA-remove), and finally (4) closing two facilities while selecting another two (SA-2swap). Note that we apply the same neighbor strategy to determine z_{ls} and afterward compute the values of p_{lcs}^{iter} , k_{lcs}^{iter} , and d_{lcs}^{iter} by using algorithm presented in Fig. 8.

3.3.3. Stopping criterion

Various stopping criteria have been developed in the literature. A popular stopping criteria and the temperature reaches a set value (such as 0.01). Another criterion can be completing a predetermined number of iterations. In this paper, a combination of these two criteria is considered in which we stop at the earlier of the temperature reaching 0.01 or the completion of 100 (350) iterations for small (large) size problems.

3.3.4. Commercial software

The optimization problem (SCD) is modeled with the AMPL mathematical programming language and solved with Gurobi version 4.5.6. Each problem instance is solved on 4 cores (threads = 4) of a Dell Optiplex 980 with an Intel Core i7 860 Quad @ 2.80 GHz, and 16 GB RAM. The operating system is Windows 7 Enterprise 64 bit. In our computational analysis, we terminate Gurobi when the CPU time limit of 14,400 s is reached. Table 2 summarizes the results from the solution, and the discussion is presented in Section 5.

4. Generation of test data

Let 0 indicate the state that a facility is producing no tainted materials; and 1 indicate the state that a facility is producing some tainted materials. Let $\Theta_l \in [0.50, 0.95]$ be the probability of facility l being in state 0. Therefore, the assumption that all facilities have an identical probability of working or failing is relaxed (Snyder and Daskin, 2005). If a facility is in state 1, the proportion of tainted material is in the range of $[0.10, 0.30]$. The proportion of tainted material that is detected after inspection is in the range of $[0.01, 0.09]$.

We define a scenario as an event where a subset of facilities L' are in state 0, and facilities in the set $L \setminus L'$ are in state 1. Given the number of facilities $|L|$, the total number of scenarios in which at least one facility is in state 1 is given by $\sum_{i=1}^{|L|} \binom{|L|}{i} = 2^{|L|} - 1$. For our computations in this paper, we also include the scenario in which all facilities are in state 0, then the total number of scenarios becomes $2^{|L|}$. Hence, the probability of realizing a scenario $s \in S$ is defined as $\rho_s = \prod_{l \in L'} \Theta_l \prod_{l \in L \setminus L'} (1 - \Theta_l)$.

Finally, recall that we discussed the calculation of the expected available capacity of a facility on step 1 of the pseudo code of SGH in Fig. 4. Since a facility can be either in state 0 with probability Θ_l or state 1 with probability $(1 - \Theta_l)$. We consider $\chi_l = \Theta_l$ in our computations. We list other assumptions as follows:

- The ratio of total demand to total capacity is tight and it is 30% higher than the total demand before performing inspection and discarding tainted items.
- The cost to discard is equal to 25% of the cost of shipping untainted materials.
- The cost of selecting a facility is correlated with the capacity such that the highest capacity has the highest selecting cost.
- The cost of inspection is correlated to the percentage improvement, which is the difference between q_l and r_l i.e., for higher improvement the inspection cost is higher.
- Parameters in Table 1 are drawn from a discrete uniform distribution with the specified intervals:

1. Calculate the expected available capacity of every facility.
2. Evaluate the total cost of selecting every facility individually. These costs are computed as

$$f_i + \frac{\sum_{c=1}^{|C|} (\lambda_{ic} + o_{ic} + \gamma_{ic})}{|C|} \text{ for all facilities.}$$
3. Sort the facilities in increasing order of total costs (using an $O(n \log n)$ sort such as HeapSort).
4. Let $i = 1$.
5. If $\sum_{j=1}^i E(\bar{\kappa}_{[j]}) \geq \sum_{c \in C} b_c$, open facilities 1 to $[i]$ and go to 6, else $i = i+1$ and go to 5.
6. **end**

Fig. 4. Pseudocode of the Selective Greedy Heuristic (SGH).

1. Calculate the expected available capacity as presented in step 1 of SGH.
2. Sort the calculated values in step 1 in a decreasing order (using an $O(n \log n)$ sort such as HeapSort).
3. Let $i = 1$.
4. If $\sum_{j=1}^i E(\bar{\kappa}_{[j]}) \geq \sum_{c \in C} b_c$, open facilities 1 to $[i]$ and go to 6, else $i = i+1$ and go to 5.
5. **end**

Fig. 5. Pseudocode of the Capacity-Based Greedy Heuristic (CBGH).

1. Calculate the maximum reduction in the fraction tainted materials as $m_i = q_i - r_i$ for each facility.
2. Sort the calculated values in step 1 in a decreasing order (using an $O(n \log n)$ sort such as HeapSort).
3. Let $s = 1$
4. Let $i = 1$
5. If $\sum_{j=1}^i r_{[j]s} x_{[j]} \kappa_{[j]} > (1 - \Delta) \sum_{c \in C} b_c$, implement inspection in selected facilities $[1]$ to $[i]$ by setting $z_{[j]s} = 1 \forall [j]: x_{[j]} = 1$ and go to 6, else $i = i+1$ and go to 5.
6. If $s < |S|$, let $s = s+1$ and go to 3, else go to 7.
7. **end**

Fig. 6. Pseudocode of the Greedy Inspection Heuristic (GIH).

Table 1
Test data intervals.

Parameter	Interval
b_c	[100, 300]
f_i	[\$1,000,000, \$2,000,000]
n_i	[\$50,000, \$100,000]
λ_{ic}	[\$500, \$1000]
o_{ic}	[\$10,000, \$25,000]

5. Computational experiments

In this section, we perform computational experiments to assess the effectiveness of the algorithms and to gain insight into the behavior of the supply chain when faced with the risk of delivering tainted materials. We also present some managerial insights.

```

for  $s = 1$  to  $|S|$ 
  1. Calculate the amount of saving  $\phi_{ls}$  for facility  $l$  in scenario  $s$ :

$$\phi_{ls} = \frac{(q_{ls} - r_{ls}) \sum_{c=1}^{|C|} o_{lc}}{n_l + (q_{ls} - r_{ls}) \sum_{c=1}^{|C|} \gamma_{lc}}, \forall l \in L.$$

  Let the vector  $\Phi_s$  be the set of calculated savings where

$$|\Phi_s| = |X|.$$

  2. Normalize vector  $\Phi_s : \hat{\Phi}_s = \frac{\Phi_s}{\|\Phi_s\|}.$ 
  3. Let  $i = 1$ 
  4. Generate  $r \sim Unif(0, 1)$ . If  $r \leq \phi_{ls}$  and  $x_i = 1$ , then  $z_{ls} = 1$ .
  5. If  $i = |L|$  then end, else  $i = i + 1$  and go to 4.
end for

```

Fig. 7. Pseudocode of the Random Greedy Inspection Heuristic (RGIH).

In Section 3.1, we presented three heuristics (BGH, SGH, and CBGH) to determine set of selected facilities, X , and also three heuristics (FSIH, GIH, and RGIH) to determine the set of inspections to conduct, z . In total, we construct nine different heuristic for determining x and z . Finally, we employ the greedy heuristic presented in Fig. 8 to solve SCD-Sub. All the algorithms and SA were implemented and executed in MATLAB 7.9 and tested on a single core of a Dell OptiPlex 980 computer running the Windows 7 Enterprise 64 bit operating system with an Intel(R) Core(TM) i7 CPU860@ 2.80 GHz, and 16 GB RAM.

We consider 12 sets of problems with 10 data instances in each. Hence, we solve in total 120 instances of varying sizes as illustrated in Table 2. The second, third and the fourth columns represent the size of the problems under consideration. We also report the average optimal value and average solution time of the SCD model for each set achieved by Gurobi. Finally, the last column represents the total number of optimal solutions obtained from 10 data instances by Gurobi within the time limit imposed.

As observed from Table 2, increasing the number of facilities implies an increase in the number of scenarios and the size of the problem has a significant impact on the solution time. For instance, for the case of 10 consumers or 20 consumers and 10 facilities, Gurobi did not return any optimal solutions within the prescribed time limit of 14,400 s. In order to calculate the relative optimality gap, we use the objective function value that is provided by Gurobi when the prescribed time limit is reached. To assess each heuristic, we consider solution quality and solution (computational) time. For the solution quality, we consider a quality criterion which is the gap between the result of heuristic/SA and the optimal/best solution obtained from Gurobi. This gap is defined according to the following equation:

$$\%gap = \frac{(SA(or\ Heuristics)Solution - Best\ Found(or\ Optimal\ Solution))}{Best\ Found(or\ Optimal\ Solution)}.$$

Furthermore, given the randomness characteristic of GIH, RGIH, HC, VNS, and SA, the corresponding objective values and solution times are the average across thirty independent replications. Table 6 reports the result for 2 facilities and 2, 5, 10, and 20 consumers. Note that bold-faced values indicate achievement of the best solutions among constructive heuristics and improvement heuristics/SA, respectively.

The results in Table 6 show that, regardless of the number of consumers, the heuristic algorithms always provide solutions within 3% of the solution found by Gurobi. Heuristic algorithms are fast and their solution time generally does not vary with the number of consumers. We see that SGH&FSIH and CBGH&FSIH algorithms provide better solution quality while HC does not provide any improvement to the solution of the constructive heuristics. The VNS procedure provides better quality solutions than the HC though this improvement comes with an increase in the solution time. Another observation from Table 6 is that even though the solution time for SA algorithm is notably higher than the other algorithms, its performance are not as good as VNS when we limit the problem instances to 2 facilities.

We now evaluate the effectiveness of our algorithms for five facilities. The results are presented in Table 7. The performance of BGH&FSIH, BGH&GIH, and BGH&RGIH is not good. The reason is that in these three heuristics we use BGH to select all the facilities while the total demand can be satisfied by selecting fewer facilities. SGH&FSIH, SGH&GIH, and SGH&RGIH provide solutions on average within 8% of the best found solution with a remarkably fast solution time in comparison to the optimal solution time. Both HC and VNS are capable of improving the solution quality even for a larger number of consumers resulting in an average solution gap within 5% of the optimal solution. In particular, SA clearly provides the best overall solution cost for the range of problems tested and requires only a moderate amount of additional computational time

Input: $c \in C$ set of consumers, $l \in L$ set of facilities, $s \in S$ set of scenarios, f_l , κ_l , n_l , b_c , ρ_s , λ_{lc} , o_{lc} , q_{ls} and r_{ls} .

Output: x_l , z_{ls} , p_{lcs} , k_{lcs} , d_{lcs} and the total cost.

Stage One: Determine x by using *BGH*, *SGH* or *CBGH*.

Stage Two: Determine z_{ls} by using *FSIH*, *GIH*, or *RGIH*.

Stage Three: Solve SCD-Sub

Phase 1

```

1.   $\forall s \in S$ 
2.  Sort  $\lambda_{lc}$  in increasing order.  $a_c \leftarrow b_c$ ;  $g_l \leftarrow \kappa_l$ .
3.   $\forall c \in C$ 
4.    do until  $a_c = 0$ 
5.       $l = \arg \min \{\lambda_{lc}\} \forall l \in L, c \in C$ 
6.      if  $g_l > a_c$  then
7.         $p_{lcs} \leftarrow a_c$ ,  $g_l \leftarrow g_l - a_c$ ,  $a_c \leftarrow 0$ 
8.      else if  $g_l > 0$  then
9.         $p_{lcs} \leftarrow g_l$ ,  $a_c \leftarrow a_c - g_l$ ,  $g_l \leftarrow 0$ 
10.     end if
11.    loop
12.  end for
13. end for

```

Phase 2

```

14.  $\forall s \in S, \forall l \in L$ 
15.  if  $z_{ls} = 0$  then
16.     $\forall c \in C$ 
17.       $d_{lcs} \leftarrow 0$ 
18.       $k_{lcs} \leftarrow q_{ls} p_{lcs}$ 
19.    end for
20.  else if  $z_{ls} = 1$  then
21.     $\forall c \in C$ 
22.       $d_{lcs} \leftarrow (q_{ls} - r_{ls}) p_{lcs}$ 
23.       $k_{lcs} \leftarrow r_{ls} p_{lcs}$ 
24.    end for
25.  end if
26. end for
27. end

```

Fig. 8. Pseudocode to Solve Problem SCD-Sub.

compared to other algorithms. SA achieves solutions which are on average within 3% of the best solutions found. For 5 facilities and 10 consumers as shown in Table 2, we found 9 optimal solutions in 10 data instances. Therefore, in Table 7 we show the gap between the optimal and non-optimal solutions (or best solutions found) separately.

In order to exercise the model more fully, we examine a set of problems with 10 facilities. In Table 2 we show that for sets 9 and 12 we were not able to find the optimal solution for any of the 10 instances in 14,400 s. In addition, in sets 3 and 5 only 50% and 10% of the data instances were solved to optimality, respectively. This indicates how increasing the number of

1. **for** each $l \in L$ **do**,
2. **if** $x_l = 1$ then $x_l \leftarrow 0$ **else** $x_l \leftarrow 1$. Let ψ be the new set of selected facilities
3. Determine z_{ls} by using FSIH, GIH, or RGIH.
4. Solve Stage Three to determine $SCD(\psi)$.
5. Compute saving as $\sigma_l = SCD(x) - SCD(\psi)$
6. **If** $\sigma_l < 0$ then $x_l \leftarrow \psi_l$ and go to 1. Else, $l = l + 1$ and go to 2.
7. **end for**

Fig. 9. Pseudocode of HC.

Consumer facility	1	2	3	4
1	50	60 +	85	85 −
2	21	45	29	29
3	81	36* −	73	73 +
4	62	78	91	20

Fig. 10. Neighborhood strategy in VNS.

1. Set $k \leftarrow 1$
2. **for** all $s \in S$
3. **while** $k < K_{max}$
- Shaking:**
4. Set $u \leftarrow p$
5. Randomly select two values of l and c to identify which flows will be changed. Call them l_1, l_2, c_1, c_2 .
6. Generate a random value $0 \leq \hat{R} \leq \min(p_{l_1 c_1}, p_{l_1 c_2}, p_{l_2 c_1}, p_{l_2 c_2})$
7. Subtract \hat{R} from $p_{l_1 c_1}$ and $p_{l_2 c_2}$ and add \hat{R} to $p_{l_1 c_2}$ and $p_{l_2 c_1}$ or vice versa (with probability $\frac{1}{2}$)
- Improve or not:**
7. Calculate the cost for u from equation (20)
8. **If** $cost(u) < cost(p)$ **then** $p \leftarrow u$ **else** $k \leftarrow k + 1$
9. **end while**
10. **end for**

Fig. 11. VNS for improving the transportation cost.

facilities and correspondingly the number of scenarios has a significant impact on the solution time. We present the results of algorithms for the tested problems in Table 8. Negative values in Table 8 indicate that the heuristics or SA achieved a better solution than the best solution found by Gurobi. For 10 facilities and 10 or 20 consumers, SGH&FSIH, SGH&GIH, and SGH&RGIH perform well based on the average solution gap. For the case of 10 facilities and two or five consumers CBGH&FSIH, CBGH&GIH, and CBGH&RGIH achieved a better performance. However, the SA solutions outperform those found by all the other algorithms, even though SA requires more computational time. Also, the minimum and maximum gap is

1. Initialize the parameters of the annealing schedule (Initial temperature, final temperature and total number of iterations)
2. Generate an initial solution by determining vector $x^0, z^0, p_{lcs}^0, k_{lcs}^0, d_{lcs}^0$ by the represented constructive or improvement heuristics and define relevant total cost $f(x^0, z^0, p_{lcs}^0, k_{lcs}^0, d_{lcs}^0)$
3. $iter \leftarrow 1$; Temperature \leftarrow Initial Temperature
4. **while** Temperature > Final Temperature or $iter <$ total number of iterations **do**
5. **while** done=false
6. aZeroElem \leftarrow Number of zero elements in vector x^{iter} and z^{iter}
7. aOneElem \leftarrow Number of one elements in vector x^{iter} and z^{iter}
8. aRand \leftarrow Generate a Random Number
9. **if** $0 \leq aRand < \frac{1}{4}$
10. create a new solution using SA-swap method and return x^{iter} and z^{iter}
11. done \leftarrow true
12. **else if** $\frac{1}{4} \leq aRand < \frac{1}{2}$ **and** $aZeroElem \geq 1$
13. create a new solution using SA-add method and return x^{iter} and z^{iter}
14. done \leftarrow true
15. **else if** $\frac{1}{2} \leq aRand < \frac{3}{4}$ **and** $aOneElem > 1$
16. create a new solution using SA-remove method and return x^{iter} and z^{iter}
17. done \leftarrow true
18. **else if** $\frac{3}{4} \leq aRand < 1$ **and** $aOneElem \geq 2$
19. create a new solution using SA-2swap method and return x^{iter} and z^{iter}
20. done \leftarrow true
21. **end while**
22. Obtain the values of $p_{lcs}^{iter}, k_{lcs}^{iter}$, and d_{lcs}^{iter} by using SCD-Sub.
23. **if** $f(x^{iter}) - f(x^0) \leq 0$ **then** $f(x^0) = f(x^{iter}), x^0 = x^{iter}$
24. Update Temperature
25. **end while**

Fig. 12. Pseudo code of the SA algorithm.

usually somewhat better for the SA. Hence, for large size problems we recommend using the SA algorithm, although reasonable results can still be achieved by some of the algorithms. For 10 facilities and five consumers, we found only 1 optimal solution in 10 data instances. Hence, we separate the result for this data instance from the others and display the gap between the optimal solution and the algorithms in the corresponding row of Table 8.

It is observable from the result that the number of facilities and consequently the number of scenarios has a significant impact on the computational time in our model. However, the results indicate the effectiveness of the SA algorithm we proposed, particularly for larger sized problems. For problems in practice (that can have even larger sizes), our SA heuristic shows promising results.

5.1. Sensitivity analysis and insights

In this section, we analyze the SCD outcomes for various settings of some of the parameters in order to provide insights that can assist decision-makers. We consider ten data instances for a supply chain network consisting of five facilities and five consumers. Note that all 10 data instances are solved to optimality and the average optimal value is presented in Table 2. We compare the results of our sensitivity analysis to the results obtained in our computational analysis, which we refer to as the “base case.” We investigate the effect of each of the following:

Table 2

Test problems' sizes and the corresponding optimum solution values and times.

Set no.	No. of consumers	No. of facilities	No. of scenarios	Avg. optimal value/ best values found	Avg. optimum time (s)	No. of optimal solutions in 10 instances
1	2	2	4	3,132,033	0.014	10
2	2	5	32	6,031,316	17.6	10
3	2	10	1024	12,190,855 ^a	11267.6	5
4	5	2	4	3,857,839	0.02	10
5	5	5	32	6,587,944	62.9	10
6	5	10	1024	12,632,921 ^a	13102.7	1
7	10	2	4	4,498,383	0.022	10
8	10	5	32	7,248,655	150.5	9
9	10	10	1024	13,207,650 ^a	^b	0
10	20	2	4	5,526,660	0.03	10
11	20	5	32	8,503,834	325.7	10
12	20	10	1024	15,095,012 ^a	^b	0

^a Average of best objective values found.^b The CPU time exceeded the prescribed time limit of 14,400 s.

- Fixed cost of selecting a facility (f_i).
- Fixed cost of implementing an inspection (n_i).
- Cost of shipping tainted materials (o_{ic}).

5.1.1. Effect of varying fixed cost of selecting a facility (f_i)

In this section, we numerically examine the impact of the fixed cost on the optimal decisions. We first assume that our facilities are small-size facilities where the fixed cost of selecting a facility is drawn from a discrete uniform distribution between \$300,000 and \$500,000. We also consider larger size facilities where the fixed cost of selecting a facility is drawn

Table 3

Impacts of the fixed cost on optimal solution.

	$f_i \in [300k, 500k]$	$f_i \in [1M, 2M]$	$f_i \in [2M, 3M]$
Avg. expected total cost	2,538,062	6,587,944	10,514,286
Avg. fixed cost	1,642,266	5,210,752	8,994,122
Avg. expected untainted delivered cost	587,447	592,174	597,569
Avg. expected tainted penalty cost	232,563(−70%)	755,792	907,643(20%)
Avg. expected inspection cost	68,523(162%)	26,185	13,203(−50%)
Avg. expected discard cost	7262(138%)	3041	1748(−42%)
Avg. no. of selected facilities	4.1	3.4	3.5

Table 4

Impacts of the inspection cost on optimal solution.

	$n_i \in [25k, 50k]$	$n_i \in [50k, 100k]$	$n_i \in [75k, 150k]$
Avg. expected total cost	6,569,842	6,587,944	6,620,171
Avg. fixed cost	5,210,752	5,210,752	5,210,752
Avg. expected untainted delivered cost	592,205	592,174	592,152
Avg. expected tainted penalty cost	751,566(−0.6%)	755,792	775,485(3%)
Avg. expected inspection cost	12,265(−52%)	26,185	38,865(48%)
Avg. expected discard cost	3054(0.4%)	3041	2942(−4%)
Avg. no. of selected facilities	3.4	3.4	3.4

Table 5

Impacts of the cost shipping tainted materials on optimal solution.

	$o_{ic} \in [5k, 10k]$	$o_{ic} \in [10k, 25k]$	$o_{ic} \in [15k, 30k]$
Avg. expected total cost	6,000,543	6,587,944	6,605,522
Avg. fixed cost	5,068,814	5,210,752	5,373,692
Avg. expected untainted delivered cost	582,557	592,174	485,849
Avg. expected tainted penalty cost	340,375(−55%)	755,792	698,080(−8%)
Avg. expected inspection cost	7971(−70%)	26,185	43,276(65%)
Avg. expected discard cost	826(−72%)	3041	4625(52%)
Avg. no. of selected facilities	3.3	3.4	3.7

Table 6

Comparison of algorithms results for 2 facilities.

	Constructive heuristics									Improve heuristic and metaheuristic		
	BGH&FSIH	BGH&GIH	BGH&RGH	SGH&FSIH	SGH&GIH	SGH&RGH	CBGH&FSIH	CBGH&GIH	CBGH&RGH	HC	VNS	SA
<i>2 Consumers</i>												
Min gap (%)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.08	0.00	0.00	0.00	0.00
Avg. gap (%)	0.40	0.47	0.22	0.21	0.64	0.36	0.36	0.56	0.25	0.21	0.06	0.25
Max gap (%)	0.89	0.77	1.59	0.76	2.49	0.76	0.56	1.59	0.59	0.56	0.09	0.89
Avg. time (s)	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.008	0.021	0.047
<i>5 Consumers</i>												
Min gap (%)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Avg. gap (%)	1.34	1.59	1.86	0.97	1.36	1.33	0.84	1.08	1.23	0.84	0.25	0.55
Max gap (%)	3.75	3.62	3.83	3.83	3.83	3.83	1.47	2.53	2.33	1.47	1.37	1.47
Avg. time (s)	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.006	0.044	0.047
<i>10 Consumers</i>												
Min gap (%)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Avg. gap (%)	0.96	2.12	0.68	0.48	2.19	1.08	0.48	0.81	1.08	0.48	0.05	0.50
Max gap (%)	4.77	7.02	1.84	1.84	7.02	4.55	1.84	2.16	2.84	1.84	0.40	1.84
Avg. time (s)	0.003	0.003	0.003	0.003	0.003	0.003	0.003	0.003	0.003	0.006	0.03	0.14
<i>20 Consumers</i>												
Min gap (%)	0.00	0.00	0.43	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Avg. gap (%)	1.58	1.58	2.31	1.58	1.58	1.97	1.58	1.90	1.90	1.58	1.29	1.58
Max gap (%)	5.37	5.37	5.37	5.37	5.37	5.37	5.37	5.37	5.37	5.37	4.49	5.37
Avg. time (s)	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.005	0.023	0.117

Bold-faced values indicate achievement of the best solutions found.

from a discrete uniform distribution between \$2,000,000 and \$3,000,000. We keep all other parameters constant. The results are presented in Table 3.

From Table 3 we observe a considerable reduction in both the average expected penalty cost of shipping tainted materials and the average expected cost of discarding tainted materials. This reduction is a consequence of an increase in the average number of selected facilities as well as the implementation of more inspections at these facilities. The reduction in the average number of selected facilities causes the decrease of the available capacity. Hence, the inspection at the facilities is refused and the tainted materials are not discarded in order to provide enough capacity to be able to satisfy the total demand of consumers. Furthermore, higher fixed costs of selecting a facility results in a reduction in implementation of inspection at the facilities, and subsequently, increased quantities of tainted materials reaching consumers. On the other hand, reducing the fixed cost of selecting a facility or adding more small size facilities results in selecting more facilities and providing extra capacity which leads to implementation of more inspection at facilities and discarding more tainted materials.

5.1.2. Effect of varying fixed cost of implementing an inspection (n_i)

In this section, we examine the impact of inspection cost on the optimal decision by considering two ranges for our sensitivity analysis. In the first range, we will have a 50% reduction in the cost of inspection (i.e., inspection cost drawn from U[\$25,000,\$50,000]), and in the second range, we consider a 50% increase in the cost of inspection (i.e., inspection cost drawn from U[\$75,000,\$150,000]). The results are presented in Table 4.

From Table 4, we note that while the inspection cost decreases, the average discard cost increases and the average expected penalty cost of shipping tainted materials decreases. We also observe higher average expected cost of discarding tainted materials, which indicates the reduction in the cost of implementing inspection and results in more inspections being performed, as would be expected. We observe that the average fixed cost, the average expected penalty cost of shipping tainted materials, and the average number of selected facilities are all insensitive to the change.

The results in Table 4 also show a reduction of inspection implementation when the inspection cost increases, while the average number of selected facilities and the average fixed cost remain unchanged. However, the average discard cost decreases and the average expected penalty cost of shipping tainted materials increases. The obtained result indicates that managers and decision-makers should maintain inspection cost at the lowest possible value.

5.1.3. Effect of varying cost of shipping tainted materials (o_{ic})

This section presents relative differences in the optimal cost expectations and the average number of selected facilities with respect to changes in the penalty cost of shipping tainted materials. We consider two ranges for our analysis. In the first range, the cost of shipping tainted materials is drawn from a discrete uniform distribution between \$5000 and

Table 7

Comparison of algorithms results for 5 facilities.

	Constructive heuristics									Improve heuristic and metaheuristic		
	BGH&FSIH	BGH&GIH	BGH&RGIH	SGH&FSIH	SGH&GIH	SGH&RGIH	CBGH&FSIH	CBGH&GIH	CBGH&RGIH	HC	VNS	SA
<i>2 Consumers</i>												
Min gap ^a (%)	7.09	7.09	7.09	0.52	1.34	1.34	0.29	0.29	0.30	0.29	0.26	0.14
Avg. gap ^a (%)	16.05	15.68	15.74	3.83	3.99	3.98	2.37	2.61	2.49	2.16	1.92	0.89
Max gap ^a (%)	30.08	30.08	28.65	12.36	12.37	12.32	7.46	7.53	7.70	6.40	2.02	1.08
Avg. time (s)	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.01	0.066	0.146
<i>5 Consumers</i>												
Min gap ^a (%)	13.10	14.38	13.74	1.00	0.98	1.45	1.44	2.38	1.47	0.58	0.00	0.04
Avg. gap ^a (%)	18.37	18.35	18.30	4.13	4.20	4.11	4.52	4.48	4.38	1.99	1.23	1.18
Max gap ^a (%)	25.37	25.33	25.64	7.78	7.60	7.54	11.14	11.11	11.27	3.50	2.83	2.66
Avg. time (s)	0.002	0.002	0.002	0.002	0.002	0.003	0.002	0.002	0.002	0.009	0.173	0.208
<i>10 Consumers</i>												
Min gap ^a (%)	8.37	9.70	9.09	1.50	1.28	3.24	3.59	3.69	3.48	1.44	0.98	0.43
Avg. gap ^a (%)	14.24	14.51	14.90	4.02	4.41	4.72	7.81	7.72	7.37	2.71	2.21	1.49
Max gap ^a (%)	19.20	18.13	19.08	6.08	6.09	6.60	12.09	11.44	10.25	3.59	3.58	2.21
Avg. gap w/non-opt sol. ^b (%)	9.63	9.70	9.83	2.17	3.74	3.86	8.31	8.70	8.15	1.03	0.03	-0.11
Avg. time (s)	0.002	0.002	0.002	0.002	0.002	0.003	0.002	0.002	0.002	0.014	0.318	0.709
<i>20 Consumers</i>												
Min gap ^a (%)	2.01	2.01	3.30	3.12	3.11	4.02	6.86	6.87	6.94	1.97	1.89	1.75
Avg. gap ^a (%)	9.70	9.70	10.91	6.04	6.02	7.94	12.75	12.83	13.78	4.65	3.53	3.17
Max gap ^a (%)	15.06	15.06	18.32	12.69	13.32	13.91	15.77	15.77	17.68	9.41	5.67	5.37
Avg. time (s)	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.029	0.635	1.049

Bold-faced values indicate achievement of the best solutions found.

Italicized indicate a better solution than the best solution found by Gurobi within the time limit.

^a Values indicate the average gap with optimal solutions found.^b Values indicate the average gap with the best solution found.

Table 8

Comparison of algorithms results for 10 facilities.

	Constructive heuristics									Improve heuristic and metaheuristic		
	BGH&FSIH	BGH&GIH	BGH&RGH	SGH&FSIH	SGH&GIH	SGH&RGH	CBGH&FSIH	CBGH&GIH	CBGH&RGH	HC	VNS	SA
<i>2 Consumers</i>												
Min gap ^a (%)	20.01	21.25	20.52	4.24	4.17	2.52	0.46	0.46	0.43	0.13	0.09	0.09
Avg. gap ^a (%)	24.10	24.02	22.75	7.04	6.86	6.57	3.93	3.89	3.56	1.14	0.54	0.12
Max gap ^a (%)	29.06	28.11	28.00	10.63	8.72	8.78	9.65	9.98	9.29	2.33	1.24	0.35
Avg. gap w/non-opt. sol. ^b (%)	21.17	21.26	21.20	6.72	6.69	6.44	3.21	3.45	3.14	1.09	0.47	-0.02
Avg. time (s)	0.005	0.005	0.005	0.005	0.005	0.006	0.005	0.005	0.005	0.218	3.838	4.075
<i>5 Consumers</i>												
Avg. gap ^a (%)	20.25	21.42	22.63	8.41	7.14	5.94	0.89	0.89	0.90	0.89	0.87	0.85
Min gap w/non-opt. sol. ^b (%)	10.37	10.31	0.70	4.92	4.32	3.72	0.89	0.89	0.90	0.87	-0.31	-0.36
Avg. gap w/non-opt. sol. ^b (%)	15.63	15.69	16.96	5.18	5.67	5.62	4.63	4.62	4.60	1.93	0.40	0.38
Max gap w/non-opt. sol. ^{**} (%)	25.23	24.98	24.70	15.40	9.52	8.89	8.67	8.83	9.78	4.22	1.40	1.73
Avg. time (s)	0.007	0.007	0.007	0.007	0.008	0.008	0.007	0.007	0.008	0.196	9.068	16.907
<i>10 Consumers</i>												
Min gap w/non-opt. sol. ^b (%)	11.14	11.08	11.30	-3.89	-3.64	-2.58	-4.34	-4.43	-4.29	-4.49	-6.28	-6.52
Avg. gap w/non-opt. sol. ^{**} (%)	18.25	18.36	17.75	3.38	3.38	4.05	5.20	5.21	5.59	0.95	-0.37	-0.88
Max gap w/non-opt. sol. ^b (%)	25.68	25.58	23.73	8.54	9.54	10.95	14.42	12.95	12.98	5.87	4.02	1.73
Avg. time (s)	0.011	0.011	0.011	0.011	0.011	0.011	0.011	0.011	0.011	0.825	15.839	19.971
<i>20 Consumers</i>												
Min gap w/non-opt. sol. ^b (%)	-0.89	-0.83	-0.15	-12.02	-11.99	-10.70	-7.63	-7.59	-5.52	-13.33	-13.33	-14.28
Avg. gap w/non-opt. sol. ^b (%)	11.26	11.30	11.77	-1.85	-1.74	0.58	4.59	4.62	5.37	-2.34	-2.76	-3.77
Max gap w/non-opt. sol. ^b (%)	17.78	17.80	18.06	2.70	2.70	5.45	17.03	17.05	17.03	2.42	2.37	0.59
Avg. time (s)	0.018	0.018	0.018	0.018	0.018	0.018	0.018	0.018	0.018	2.266	35.945	44.989

Bold-faced values indicate achievement of best solutions found.

Italicized indicate a better solution than the best solution found by Gurobi within the time limit.

^a Values indicate the average gap with optimal solutions found.^b Values indicate the average gap with the best solution found.

\$10,000. For the second range, the cost of shipping tainted materials is drawn from a discrete uniform distribution between \$15,000 and \$30,000. The results are presented in Table 5.

For $o_{ic} \in U[5000, 10,000]$, we observe that the average number of selected facilities is insensitive to the change. We also notice a remarkable reduction in the inspection cost and the average expected discard cost. This reduction implies the implementation of fewer inspections at the facilities. These observations indicate that the use of a low penalty cost for shipping tainted materials results in decisions that do not encourage the detection of tainted materials or the selection of enough facilities to protect against shipping tainted materials. For $o_{ic} \in U[15,000, 30,000]$, we observe a noticeable increase in the number of selected facilities. We also notice an increase in the inspection cost and the average expected discard cost.

The results imply that lower penalty costs for shipping tainted materials will lead to fewer facilities being selected and also fewer inspections conducted at facilities. If the decision maker wants to protect against the delivery of tainted materials, then either the penalty cost must be increased or the maximum allowable amount of tainted product delivery must be reduced when designing a supply chain network model.

Another parameter that is fixed (as an assumption) is that inspection yields are known a priori. This is another area in which the decision maker can carry out sensitivity analyses (based on inspection yield values that they feel are appropriate) in order to gain the insight into which facilities to open or close.

6. Conclusions and future research

This research addresses a supply chain network design to hedge against the risk of supply disruptions and sending tainted materials to consumers. The aim of the model consists of the facility selection, actual capacity allocation among the consumers, and determination of an inspection policy with the objective of minimizing the total cost. The impact of supply/capacity uncertainty is explicitly modeled in all our models in order to design a reliable supply chain network.

Experience from solving the problem using commercial software indicated that the number of facilities, and consequently the number of scenarios, has a significant impact on the computational time. As a result, we developed several heuristic methods and a metaheuristic approach to effectively solve the presented model. Based on our results, the SA approach is not efficient in terms of solution quality and solution time for the small sized problems or small number of scenarios. However, some of the heuristics, in particular SGH&FSIH, SGH&GIH, SGH&RGIH and CBGH&FSIH, achieved good solution qualities in a more reasonable time when compared to the optimal or best found solution. HC and VNS were able to improve the solutions obtained from constructive heuristics. Therefore, constructive and improvement heuristics are preferable on small sized problems. However, for practical sized problems, i.e. ten facilities and more, SA outperforms constructive and improvement heuristics, even though it requires higher computational time.

Our sensitivity analysis revealed some managerial insights. Our analysis showed that the fixed cost of selecting a facility or adding smaller-sized facilities results in the implementation of more inspections at facilities and discarding more tainted products. Furthermore, maintaining inspection costs at the lowest possible value leads to shipping less tainted products. Finally, our analysis indicated that considering high penalty costs of shipping tainted products when designing a supply chain network model results in discarding more tainted products and consequently shipping less tainted products.

The contributions of the paper to the literature are threefold. First, we develop and present a mathematical model in order to design a supply chain infrastructure that is resilient to supply quality disruption and shipping tainted materials to consumers. Second, we introduce a producer-implemented inspection policy in order to reduce the volume of tainted materials. The costs related to inspection and disposal of discovered tainted materials are also incurred in the mathematical model. Finally, we develop several heuristics and metaheuristics to obtain acceptable solutions to the models in a reasonable time.

There are several interesting future research directions. We assumed a deterministic demand in our model whereas in real world this may not be a valid assumption. Moreover, we assumed an inspection and discard policy but in some industries like automotive and electronics industry this can be considered as inspection and fixed policy where items defected after detecting can be repaired. As another future extension, we can consider other solution methodologies, such as Bender's Decomposition, or Lagrangian relaxation, to solve larger-sized problems. Other metaheuristic techniques such as Genetic Algorithm or Tabu Search can be developed to compare their effectiveness with SA algorithm.

Acknowledgements

This work was partially supported by a Center Directed Project of the Center for Excellence in Logistics and Distribution (CELDi), an Industry/University Cooperative Research Center (I/UCRC) sponsored by the National Science Foundation – United States under grant number IIP 0641414.

Appendix A

This appendix contains all the detailed tables of sensitivity analysis (see Figs. 4–12 and Table 2–8).

References

- Agrawal, V., Seshadri, S., 2000. Impact of uncertainty and risk aversion on price and order quantity in the newsvendor problem. *Manuf. Serv. Oper. Manage.* 2, 410.
- Alonso-Ayuso, A., Escudero, L., Garin, A., Ortuno, M., Pérez, G., 2003. An approach for strategic supply chain planning under uncertainty based on stochastic 0–1 programming. *J. Global Optim.* 26, 97–124.
- Benson, P., 2012. Generic-drug user fees mark a step in the right direction. *Pharm. Technol.* 36, 98.
- Brandeau, M.L., Chiu, S.S., 1989. An overview of representative problems in location research. *Manage. Sci.* 35, 645–674.
- Caserta, M., Rico, E.Q., 2008. A cross entropy-based metaheuristic algorithm for large-scale capacitated facility location problems. *J. Oper. Res. Soc.* 60, 1439–1448.
- Chopra, S., Sodhi, M.M.S., 2004. Managing risk to avoid supply-chain breakdown. *MIT Sloan Manage. Rev.* 46, 53–61.
- Cornuejols, G., Sridharan, R., Thizy, J., 1991. A comparison of heuristics and relaxations for the capacitated plant location problem. *Eur. J. Oper. Res.* 50, 280–297.
- Daskin, M.S., Coullard, C.R., Shen, Z.M., 2002. An inventory-location model: formulation, solution algorithm and computational results. *Ann. Oper. Res.* 110, 83–106.
- Demirtas, E.A., Üstün, Ö., 2008. An integrated multiobjective decision making process for supplier selection and order allocation. *Omega* 36, 76–90.
- Doerner, K.F., Gendreau, M., Greistorfer, P., 2007. *Metaheuristics: Progress in Complex Systems Optimization*. Springer Verlag.
- Drezner, Z., Hamacher, H.W., 2004. *Facility Location: Applications and Theory*. Springer Verlag.
- Everett, L., Baker, M.M., 2004. LaRouche: reverse the policy that created the flu crisis. *Executive Intell. Rev.* 31, 4–13.
- Geoffrion, A., Graves, G., 1974. Multicommodity distribution system design by benders decomposition. *Manage. Sci.* 20, 822.
- Glover, F., Kochenberger, G.A., 2003. *Handbook of Metaheuristics*. Springer.
- Green, R.L., Brown, C.D., 2009. Moving toward 100% raw material inspection with a handheld Raman spectrometer. *Eur. J. Parenteral Sci. Pharm. Sci.* 14, 111.
- Hinojosa, Y., Kalcsics, J., Nickel, S., Puerto, J., Velten, S., 2008. Dynamic supply chain design with inventory. *Comput. Oper. Res.* 35, 373–391.
- Kirkpatrick, S., Gelatt Jr., C.D., Vecchi, M.P., 1983. Optimization by simulated annealing. *Science* 220, 671–680.
- Laporte, G., Louveaux, F.V., van Hamme, L., 1994. Exact solution to a location problem with stochastic demands. *Transp. Sci.* 28, 95–103.
- Lyles, M.A., Flynn, B.B., Frohlich, M.T., 2008. All supply chains don't flow through: Understanding supply chain issues in product recalls. *Manage. Org. Rev.* 4, 167–182.
- Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E., 1953. Equation of state calculations by fast computing machines. *J. Chem. Phys.* 21, 1087.
- MirHassani, S.A., Lucas, C., Mitra, G., Messina, E., Poojari, C.A., 2000. Computational solution of capacity planning models under uncertainty. *Parallel Comput.* 26, 511–538.
- Owen, S.H., Daskin, M.S., 1998. Strategic facility location: a review. *Eur. J. Oper. Res.* 111, 423–447.
- Qi, L., Shen, Z.M., 2007. A supply chain design model with unreliable supply. *Naval Res. Logis.* 54, 829–844.
- Santos, T., Ahmed, S., Goetschalckx, M., Shapiro, A., 2005. A stochastic programming approach for supply chain network design under uncertainty. *Eur. J. Oper. Res.* 167, 96–115.
- Schoder, D., 2010. Melamine milk powder and infant formula sold in East Africa. *J. Food Prot.* 73, 1709–1714.
- Shapiro, A., 2008. Stochastic programming approach to optimization under uncertainty. *Math. Program.* 112, 183–220.
- Shen, Z.J.M., Zhan, R.L., Zhang, J., 2008. The reliable facility location problem: formulations, heuristics, and approximation algorithms. *INFORMS J. Comput.*
- Snyder, L.V., Daskin, M.S., 2005. Reliability models for facility location: the expected failure cost case. *Transp. Sci.* 39, 400–416.
- Sridharan, R., 1995. The capacitated plant location problem. *Eur. J. Oper. Res.* 87, 203–213.
- Suman, B., Kumar, P., 2005. A survey of simulated annealing as a tool for single and multiobjective optimization. *J. Oper. Res. Soc.* 57, 1143–1160.
- Talbi, E., 2009. *Metaheuristics: From Design to Implementation* (Wiley Series on Parallel and Distributed Computing). Wiley.
- Tsiakis, P., Shah, N., Pantelides, C.C., 2001. Design of multi-echelon supply chain networks under demand uncertainty. *Ind. Eng. Chem. Res.* 40, 3585–3604.
- Usdin, S., 2009. The heparin story. *Int. J. Risk Saf. Med.* 21, 93–103.
- Vidal, C.J., Goetschalckx, M., 1997. Strategic production–distribution models: a critical review with emphasis on global supply chain models. *Eur. J. Oper. Res.* 98, 1–18.