

Research Article

Designing a Supply Chain Network under the Risk of Disruptions

**Armin Jabbarzadeh,¹ Seyed Gholamreza Jalali Naini,¹
Hamid Davoudpour,² and Nader Azad²**

¹ Department of Industrial Engineering, Iran University of Science and Technology,
16846113114 Tehran, Iran

² Department of Industrial Engineering, Amirkabir University of Technology,
158754413 Tehran, Iran

Correspondence should be addressed to Armin Jabbarzadeh, arminj@iust.ac.ir

Received 23 August 2011; Accepted 19 December 2011

Academic Editor: Andrzej Swierniak

Copyright © 2012 Armin Jabbarzadeh et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper studies a supply chain design problem with the risk of disruptions at facilities. At any point of time, the facilities are subject to various types of disruptions caused by natural disasters, man-made defections, and equipment breakdowns. We formulate the problem as a mixed-integer nonlinear program which maximizes the total profit for the whole system. The model simultaneously determines the number and location of facilities, the subset of customers to serve, the assignment of customers to facilities, and the cycle-order quantities at facilities. In order to obtain near-optimal solutions with reasonable computational requirements for large problem instances, two solution methods based on Lagrangian relaxation and genetic algorithm are developed. The effectiveness of the proposed solution approaches is shown using numerical experiments. The computational results, in addition, demonstrate that the benefits of considering disruptions in the supply chain design model can be significant.

1. Introduction

Under today's highly competitive business environment, supply chain network design is a critical and difficult decision. Many of supply chain design decisions such as facility location are strategic in nature and very expensive to change. In particular, supply chain design involves both strategic decisions of facility location and tactical decisions of inventory. Traditional supply chain design models in the literature treat location and inventory decisions separately. However, ignoring interaction between long-term decisions of location and short-term decisions of inventory can lead to suboptimality [1–3]. Thus, integrated supply chain design models incorporating location and inventory decisions have emerged in recent years [4, 5].

Daskin et al. [1] and Shen et al. [6] develop a basic integrated location inventory model that explicitly considers expected inventory costs when making facility location decisions. The model simultaneously determines facility location and demand assignment decisions in order to minimize the total cost including location, inventory, and shipment costs. This basic integrated location-inventory model is extended by researchers [3, 7–12]. For instance, Snyder et al. [7] propose a stochastic version of integrated location-inventory that handles uncertainty by describing discrete scenarios. Max Shen and Qi [3], also, add routing decisions to joint location-inventory model. That is, they examine an integrated model which determines location, inventory, and routing decisions. Ozsen et al. [8] consider an integrated location inventory when capacities of facilities are limited. Yao et al. [11] consider a joint location-inventory problem in which a company allows its retailers to be sourced by more than one warehouse. Shen [4] and Melo et al. [5] present comprehensive literature reviews on integrated supply chain design models.

A limitation of the most existing studies on the integrated supply chain design is that they implicitly assume that facilities are always available and perfectly reliable. However, facilities in real world are always vulnerable to partial or complete disruptions. In fact, different factors, such as natural disasters, labor strikes, power outages, parts shortages, quality rejections, poor communications of customer requirements, transportation damages, and machine breakdowns, can lead to unreliable performance of facilities and huge losses. For instance, many companies like Intel, Wal-Mart, Ford, Isuzu Motors, and Suzuki had to stop production because of cut-off of electricity and water supply at their suppliers in 2008 [13]. Disruption at a Philips Semiconductor plant in 2001 caused shortages of cell phone components for Ericsson which resulted in losing a substantial portion of Ericsson's market to its rival, Nokia [14]. Furthermore, smaller-scale disruptions at facilities take place much more frequently. For example, Wal-Mart's Emergency Operations Center receives a call almost every day from a store with some sort of crisis [15]. These examples show that disruptions can significantly impact firm's operations and highlight the need to account for disruptions during design of supply chain network.

Facility location models with disruptions consideration have gained much attention recently [16–21]. Snyder and Daskin [16] present two facility location models based on P-median problem and incapacitated fixed location problem, in which facilities are disrupted with the same probabilities. In both models, the objective functions include terms of expected operational cost and expected failure cost to model a trade-off between these two costs. Their models rely on the assumption that the disruption probabilities of the facilities are equal. This assumption is relaxed by [17–19].

Berman et al. [17] study a P-median problem where each facility fails independently with a certain probability. The objective minimizes the expected P-median cost plus penalty cost for not being able to serve a customer. Due to intractability of the model, the authors suggest heuristic to solve the problem. Lim et al. [18] also focus on a facility location problem in presence of disruptions with the option of hardening selected facilities. Their model is formulated based on the assumption that disruption probabilities are independent. Cui et al. [19] propose a continuum approximation model to study incapacitated fixed charge location problem in which facilities are disrupted with site-dependent probabilities. Their model seeks to minimize initial setup costs and expected transportation costs in normal and failure scenarios. Snyder et al. [22] and Snyder and Daskin [23] review the broad range of facility location models under presence of disruptions. Among the above works in the area of facility location with disruptions, no model considers inventory costs.

Aryanezhad et al. [24] and Chen et al. [25] propose location-inventory models when facilities are subject to disruptions. They propose integer programming models that minimize the sum of facility construction costs, expected inventory costs, and expected customer costs under normal and failure cases. Their models are based on the restrictive assumption that facilities fail independently with an equal probability. Also, they do not consider partial disruptions in their models. Qi and Shen [26] concentrate on integrated supply chain design under yield uncertainty; however, they do not consider disruptions at facilities. Qi et al. [27] formulate a joint location-inventory model in which facilities can be disrupted. They assume that all inventory at a facility is destroyed if the facility is disrupted. In fact, none of the above models consider both partial and complete disruptions at facilities.

This paper investigates an integrated supply chain design problem with multiple distribution centers subject to various types of disruptions. The problem is formulated as a nonlinear integer programming model which determines the location of distribution centers and the assignments of customers to distribution centers. The proposed model in this study builds upon recent developments of integrated supply chain design models that simultaneously consider location, inventory, and shipment decisions in the same model. In order to obtain near-optimal solutions with reasonable computational requirements, two solution methods based on Lagrangian relaxation and genetic algorithm are developed. Numerical experiments are conducted to test the performance of solution approaches and draw managerial insights on the benefits of considering disruptions during design of supply chain networks.

The proposed model in this paper differs from the earlier works in literature of integrated supply chain design. Unlike most of joint location-inventory models in the literature, the model takes into account the different disruptions scenarios at facilities during decision making. Also, the profit-maximizing model in this study does not restrictively assume that every potential demand has to be satisfied. Typically, cost-minimizing models in the literature require every potential demand must be met. However, for a profit-maximizing business, it may not always be optimal to serve all potential customers, especially if the additional cost is higher than the additional revenue associated with satisfying some demands [26, 28]. This article is also different from the literature on facility location problems with disruptions. First, the model does not ignore nonlinear inventory costs. In addition, it takes into account the possibility of both partial and complete facility disruptions in the same model.

The remainder of the paper is organized as follows. In Section 2, the problem is described and the formulation model for the problem is presented. Section 3 proposes two solution approaches based on Lagrangian relaxation and genetic algorithm. In Section 4, numerical experiments are conducted to evaluate the performance of solution methods and to study the benefits of considering disruptions in the supply chain design model. Finally, Section 5 concludes the paper along with directions for future research.

2. Model Formulation

2.1. Problem Description and Assumptions

This paper addresses a supply chain design network problem under random facility disruptions. As typically considered in the supply chain design literature, a three-tiered supply chain consisting of a single supplier, distribution centers (DCs), and customers is



Figure 1: Strategy of the present article for determining the supply chain design decisions.

studied. The supplier ships one type of product to a set of customers in order to satisfy their demands. The supply chain is flexible in determining which customers to serve. In other words, if the cost of serving some customers is prohibitive, they are not served at all. DCs function as the direct intermediary between the plant and customers for shipment of the product. That is, DCs combine the orders from different customers and then order to the supplier. Similar to [17–19, 21, 25, 27], we assume that the lead time for order delivery is negligible and the demand rate is fixed. Also, the capacities of DCs are assumed to be infinite as assumed in [1–3, 6, 7, 17–19, 26–28].

The key problem is that DCs may face different amounts of disruptions from time to time. If a customer is assigned to a DC but the DC is disrupted and cannot meet all of the customer's demands, the unmet demands are lost. In this case, the system incurs a lost-sales cost for each unit of lost demand. The percentage of disruptions at each DC is uncertain. In other words, each DC may experience different amounts of disruptions with different probabilities. DCs are not required to be the same in the problem; as a result, the probabilities of disruptions for each DC can be different from those of others. To formulate the disruptions at the DCs, a scenario-based modeling approach is used, in which each scenario specifies the percentage of disruptions for each DC. For instance, if a DC experiences complete disruptions in a scenario, the percentage of the disruptions is considered hundred for that DC. Note that the scenario-based modeling framework is flexible enough to consider both complete and partial disruptions. Also, it allows us to model the complex situation in which the probability failures of DCs are dependent.

The problem lies in simultaneously determining (1) how many DCs are opened, and where to locate them; (2) which subset of customers is served; (3) which DCs are assigned to which customers; (4) how much and how often to order at each DC. Figure 1 demonstrates the structure of the paper for solving this problem. The problem is formulated as a mixed-integer nonlinear program which maximizes the expected total profit. That is, the objective is to maximize the expected difference between total revenue and total cost. The total cost includes three main components: (1) the fixed cost to locate DCs, (2) the working inventory cost (including order costs, shipment costs from supplier to DCs, holding costs, and lost-sales costs) at the located DCs, and (3) shipment cost from located DCs to customers.

2.2. Notation

To develop the integrated model, the following notations are used throughout the paper. Additional notations will be given out when required.

Parameters

- (i) I : set of customers indexed by i ;
- (ii) J : set of candidate DC locations indexed by j ;

- (iii) S : set of disruption scenarios indexed by s ;
- (iv) λ_i : demand rate at customer i , for each $i \in I$;
- (v) R_i : selling price at customer i , per unit of demand for each $i \in I$;
- (vi) f_j : fixed cost of locating a DC at j , for each $j \in J$;
- (vii) a_j : fixed cost of placing an order at j , for each $j \in J$;
- (viii) b_j : fixed cost per shipment from the supplier to DC at j , for each $j \in J$;
- (ix) c_j : per-unit shipment cost from the supplier to DC at j , for each $j \in J$;
- (x) h : inventory holding cost per unit of product per year;
- (xi) d_{ij} : per-unit cost to ship from distribution center j to customer i , for each $i \in I$ and for each $j \in J$;
- (xii) β : weight factor associated with the shipment cost;
- (xiii) θ : weight factor associated with the inventory cost;
- (xiv) r_{sj} : percentage of supply at distribution center j which is disrupted in scenario s , for each $s \in S$ and for each $j \in J$;
- (xv) π_j : penalty cost for not being able to satisfy customers' demands due to disruptions at j , per unit of demand, for each $j \in J$ (it may be a lost-sales cost during disruptions, or the cost of filling the supply by purchasing product from a competitor on an emergency basis to compensate the disrupted supply at j);
- (xvi) q_s : probability that scenario s occurs, for each $s \in S$.

Decision Variables

- (i) $X_j = 1$, if j is selected as a DC location, and 0, otherwise, for each $j \in J$;
- (ii) $Y_{ij} = 1$, if customer i is assigned to a DC based at j , and 0 otherwise, for each $i \in I$ and $j \in J$;
- (iii) $Z_i = 1$, if customer i is not selected to be served, and 0 otherwise, for each $i \in I$.

The total demand which is assigned to a distribution center is unknown in advance. However, the assigned demand to each distribution center j can be obtained by the assignment decisions (Y) as follows:

$$D_j = \sum_{i \in I} \lambda_i Y_{ij}, \quad (2.1)$$

where D_j denotes the total demand that is assigned to the DC at j .

2.3. Working Inventory Cost

This subsection formulates the working inventory cost including costs of ordering, shipment from supplier to DCs, holding, and lost-sales. For the moment, let D_j denote the unknown total demand that is assigned to the DC at j (it is obvious that $D_j = \sum_{i \in I} \lambda_i Y_{ij}$). Also, let n be the unknown number of orders per year.

Then, the expected shipment size per shipment from the supplier to DC at j is equal to D_j/n and the working inventory cost at distribution center j can be obtained by

$$a_j n + \beta \left(b_j + \frac{c_j D_j}{n} \right) n + \sum_s q_s \left(\frac{(1-r_{sj})D_j}{2n} \theta h \right) + \sum_s q_s \left(\frac{r_{sj} D_j}{2n} \pi_j \right). \quad (2.2)$$

The first term of (2.2) is the fixed cost of placing n orders. The second term indicates the cost of shipping n orders of size D_j/n , assuming the shipment cost from the supplier to the distribution center j has a fixed cost b_j and volume-dependent cost c_j . The third term represents the cost of holding average of $\sum_s q_s ((1-r_{sj})D_j/2n)$ units. To explain this amount of average inventory, we note that if the DC at j was not subject to any disruptions, the average units to hold would be $D_j/2n$ [29]. Also, recall that if scenario s was occurred with certainty, r_{sj} % of supply at distribution center j would be disrupted and the average inventory would be $(1-r_{sj})D_j/2n$. Thus, knowing that each scenario s is occurred with probability of q_s , the average units to hold will be $\sum_s q_s ((1-r_{sj})D_j/2n)$. Also, the average disrupted supply at distribution center j will be $\sum_s q_s (r_{sj} D_j/2n)$. Therefore, the last term in (2.2) indicates the lost-sales cost due to disruptions at distribution center j , where the penalty cost for losing a unit of supply due to disruptions is denoted by π_j . In order to determine the optimal number of orders, we take derivative of (2.2) with respect to n and set the derivative to zero:

$$a_j + \beta b_j - \sum_s q_s \left(\frac{(1-r_{sj})D_j}{(2n)^2} \theta h \right) - \sum_s q_s \left(\frac{r_{sj} D_j}{(2n)^2} \pi_j \right) = 0. \quad (2.3)$$

Solving (2.3) for n , we obtain $n = \sqrt{\sum_s q_s (r_{sj} \pi_j + (1-r_{sj}) \theta h) D_j / 2(a_j + \beta b_j)}$. Plugging this into (2.2), working inventory cost at distribution center j can be calculated as follows:

$$\sqrt{2(a_j + \beta b_j) \sum_s q_s (r_{sj} \pi_j + (1-r_{sj}) \theta h) D_j} + \beta c_j D_j. \quad (2.4)$$

Since $D_j = \sum_{i \in I} \lambda_i Y_{ij}$, (2.4) can be rewritten as follows:

$$\sqrt{2(a_j + \beta b_j) \sum_s q_s (r_{sj} \pi_j + (1-r_{sj}) \theta h) \sum_{i \in I} \lambda_i Y_{ij} + \beta c_j \sum_{i \in I} \lambda_i Y_{ij}}. \quad (2.5)$$

2.4. Integrated Model

The problem is formulated as follows:

$$\begin{aligned} \text{Max } & \sum_{i \in I} R_i \lambda_i (1 - Z_i) - \left(\sum_{j \in J} f_j X_j \right) - \beta \sum_{j \in J} \sum_{i \in I} d_{ij} \lambda_i Y_{ij} \\ & - \sum_{j \in J} \left(\sqrt{2(a_j + \beta b_j) \sum_s q_s (r_{sj} \pi_j + (1-r_{sj}) \theta h) \sum_{i \in I} \lambda_i Y_{ij} + \beta c_j \sum_{i \in I} \lambda_i Y_{ij}} \right) \end{aligned} \quad (2.6)$$

subject to

$$\sum_{j \in J} Y_{ij} + Z_i = 1 \quad \forall i \in I, \quad (2.7)$$

$$Y_{ij} \leq X_j \quad \forall i \in I, \forall j \in J, \quad (2.8)$$

$$X_j \in \{0, 1\} \quad \forall j \in J, \quad (2.9)$$

$$Y_{ij} \in \{0, 1\} \quad \forall i \in I, \forall j \in J. \quad (2.10)$$

The objective function (2.6) is composed of four components. The first component indicates the sales revenue which is gained by serving the customers. The second component represents the fixed cost of locating DCs. The third component indicates the expected shipment cost from the DCs to customers. Finally, the fourth component represents the working inventory cost. Constraints (2.7) require each customer to be assigned to exactly one DC, or not to be served at all. Constraints (2.8) state that customers can only be assigned to candidate sites that are selected as DCs. Constraints (2.9) and (2.10) are binary constraints.

3. Solution Methods

In order to solve the model formulated in Section 2, two solution methods are developed. The first solution approach is based on Lagrangian relaxation which has been largely applied in the literature of integrated supply chain design models [1, 3, 7–9, 18, 19, 25, 27]. Also, a solution method based on genetic algorithm is developed, to obtain near-optimal solutions in relatively short time. Genetic algorithm has been successfully used to solve various supply chain design problems and has proven to be a very effective heuristic procedure to solve these problems, particularly problems of large scale [21, 30–34]. It can overcome computational complexity caused by the nonlinear and stochastic objective functions to solve the model [34, 35].

3.1. Lagrangian Relaxation

Here, a Lagrangian relaxation is developed to solve the model due to its proven effectiveness for solving integrated supply chain design models [1, 3, 7–9, 18, 19, 25, 27]. Lagrangian relaxation approach provides both upper and lower bounds on the optimal value of the objective function. In other words, it allows the decision maker to know how far from the optimality the best found feasible solution is [36]. Figure 2 outlines the steps of the developed Lagrangian relaxation. In the following subsections, the steps of the proposed Lagrangian relaxation are explained with more details.

3.1.1. Finding a Lower Bound

First, the model formulated in Section 2 is converted into a standard model for which an effective Lagrangian relaxation approach exists in the literature [1, 27]. Replacing Z_i in (2.6)

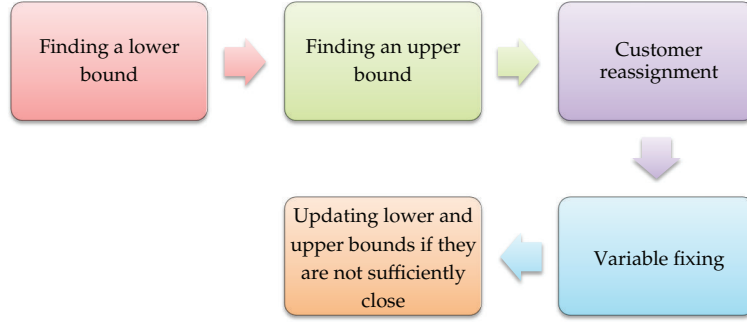


Figure 2: Lagrangian relaxation procedure.

with $1 - \sum_{j \in J} Y_{ij}$ according to (2.7), the original model formulated in Section 2 can be written as follows:

$$\begin{aligned} \text{Max } & \sum_{j \in J} \sum_{i \in I} R_i \lambda_i Y_{ij} - \left(\sum_{j \in J} f_j X_j \right) - \beta \sum_{j \in J} \sum_{i \in I} d_{ij} \lambda_i Y_{ij} \\ & - \sum_{j \in J} \left(\sqrt{2(a_j + \beta b_j) \sum_s q_s (r_{sj} \pi_j + (1 - r_{sj}) \theta h) \sum_{i \in I} \lambda_i Y_{ij} + \beta c_j \sum_{i \in I} \lambda_i Y_{ij}} \right) \end{aligned} \quad (3.1)$$

subject to

$$\sum_{j \in J} Y_{ij} \leq 1 \quad \forall i \in I, \quad (3.2)$$

$$Y_{ij} \leq X_j \quad \forall i \in I, \forall j \in J, \quad (3.3)$$

$$X_j \in \{0, 1\} \quad \forall j \in J, \quad (3.4)$$

$$Y_{ij} \in \{0, 1\} \quad \forall i \in I, \forall j \in J. \quad (3.5)$$

By changing the sign of (3.1) and rearranging the terms, the problem can be converted into a minimizing model with the following objective:

$$\begin{aligned} \text{Min } & \sum_{j \in J} f_j X_j + \beta \sum_{j \in J} \sum_{i \in I} d_{ij} \lambda_i Y_{ij} \\ & + \sum_{j \in J} \left(\sqrt{2(a_j + \beta b_j) \sum_s q_s (r_{sj} \pi_j + (1 - r_{sj}) \theta h) \sum_{i \in I} \lambda_i Y_{ij} + \beta c_j \sum_{i \in I} \lambda_i Y_{ij}} \right) - \sum_{j \in J} \sum_{i \in I} R_i \lambda_i Y_{ij} \end{aligned}$$

$$\begin{aligned}
&= \sum_{j \in J} \left(f_j X_j + \sum_{i \in I} (\beta d_{ij} + \beta c_j - R_i) \lambda_i Y_{ij} + \sqrt{2(a_j + \beta b_j) \sum_s q_s (r_{sj} \pi_j + (1 - r_{sj}) \theta h) \sum_{i \in I} \lambda_i Y_{ij}} \right) \\
&= \sum_{j \in J} \left(f_j X_j + \sum_{i \in I} u_i Y_{ij} + \sqrt{\sum_{i \in I} v_i Y_{ij}} \right),
\end{aligned} \tag{3.6}$$

where

$$\begin{aligned}
u_{ij} &= (\beta d_{ij} + \beta c_j - R_i) \lambda_i, \\
v_{ij} &= 2(a_j + \beta b_j) \lambda_i \sum_s q_s (r_{sj} \pi_j + (1 - r_{sj}) \theta h).
\end{aligned} \tag{3.7}$$

Obviously the upper and lower bounds for (3.6) can easily be used as the lower and upper bounds for (3.5) by changing their signs.

Relaxing constraints (3.2) with Lagrange multipliers, ω_i , leads to the following Lagrangian dual problem:

$$\begin{aligned}
&\text{Max}_{\omega} \text{Min}_{X, Y} \sum_{j \in J} \left\{ f_j X_j + \sum_{i \in I} u_{ij} Y_{ij} + \sqrt{\sum_{i \in I} v_{ij} Y_{ij}} \right\} + \sum_{i \in I} \omega_i \left(1 - \sum_{j \in J} Y_{ij} \right) \\
&= \sum_{j \in J} \left\{ f_j X_j + \sum_{i \in I} (u_{ij} - \omega_i) Y_{ij} + \sqrt{\sum_{i \in I} v_{ij} Y_{ij}} \right\} + \sum_{i \in I} \omega_i
\end{aligned} \tag{3.8}$$

subject to

$$Y_{ij} \leq X_j \quad \forall i \in I, \forall j \in J, \tag{3.9}$$

$$X_j \in \{0, 1\} \quad \forall j \in J, \tag{3.10}$$

$$Y_{ij} \in \{0, 1\} \quad \forall i \in I, \forall j \in J. \tag{3.11}$$

For given values of the Lagrange multipliers, ω_i , the objective is to minimize (3.8) over the decision variables X_j and Y_{ij} . This problem is decomposed by j ; as a result, we need to solve the following sub-problem for each candidate location $j \in J$:

$$\text{SP}_j : \tilde{V}_j = \text{Min} \sum_{i \in I} (u_{ij} - \omega_i) P_i + \sqrt{\sum_{i \in I} v_{ij} P_i} \tag{3.12}$$

subject to

$$P_i \in \{0, 1\} \quad \forall i \in I. \tag{3.13}$$

In (3.12)-(3.13), the assignment variables Y_{ij} have been replaced by P_i to simplify the notation, as SP_j is specific to distribution j . Note that the facility location cost, f_j , is not included in \tilde{V}_j and will be added later. Subproblem SP_j can be solved using the exact algorithm developed by Shen et al. [6]. Modified to our problem, their algorithm is given as follows.

- (1) Define $I^0 = \{i : (u_{ij} - \omega_i) < 0 \text{ and } v_{ij} = 0\}$ and $I^- = \{i : (u_{ij} - \omega_i) < 0 \text{ and } v_{ij} > 0\}$.
- (2) Sort the elements of I^- in increasing order of $(u_{ij} - \omega_i)/v_{ij}$ and represent the elements by $1^-, 2^-, \dots, n^-$, respectively, where $n = |I^-|$.
- (3) Find the value of m that minimizes

$$\sum_{i \in I^0} (u_{ij} - \omega_i)P_i + \sqrt{\sum_{i \in I^0} v_{ij}P_i} + \sum_{i=1, i \in I^-}^m (u_{ij} - \omega_i)P_i + \sqrt{\sum_{i=1, i \in I^-}^m v_{ij}P_i}. \quad (3.14)$$

- (4) The optimal solution to subproblem SP_j is obtained by $P_i = 1$ for $i \in I^0$, $P_{1^-} = P_{2^-} = \dots = P_{m^-} = 1$ for $i \in I^-$ and $P_i = 0$ for all other $i \in I$.

After solving subproblem SP_j for each j , f_j is added to the optimal objective value of \tilde{V}_j . If $\tilde{V}_j + f_j < 0$, then we select the candidate distribution center j and set $X_j = 1$; otherwise, we set $X_j = 0$. For each selected distribution center j (those for which $X_j = 1$), the assignment variables Y_{ij} are the same as the optimal P_i values in subproblem SP_j ; for each unselected distribution center j (those for which $X_j = 0$), $Y_{ij} = 0, \forall i \in I$.

Having solved the Lagrangian problem, the optimal Lagrange multipliers are found using a standard subgradient optimization procedure [37, 38]. The optimal objective value of the Lagrangian dual problem (3.8) provides a lower bound on the optimal objective value of (3.6).

3.1.2. Finding an Upper Bound

The lower bound solution obtained in each iteration of Lagrangian relaxation can violate constraints (3.2). In other words, some customers can be assigned to more than one DC in the obtained lower bound solution. Thus, in each iteration of Lagrangian relaxation, the obtained lower bound solution is converted into feasible upper bound solution using the procedure adapted from Shen et al. [6], as follows.

- (1) If a customer is assigned to more than one DC, we find the DC which assigning it to the customer leads to the least objective (3.6). If the resulted objective (3.6) is less than the case that the customer is not served at all, we assign the customer to that DC. Otherwise, the customer is not assigned to any DC and is not served at all.
- (2) The unnecessary DCs, those which no longer serve any customers after performing step 1, are closed.

If the obtained feasible solution results in a less value for (3.6) than the best known upper bound, it will be taken as the new upper bound solution. Also, it will be improved using customer reassignment algorithm.

3.1.3. Customer Reassignment

Each obtained upper bound is improved using following algorithm.

- (1) For each customer, we check whether the objective value (3.6) is improved if the customer is assigned, instead to another located DC, or if it is not served at all. The best improving swap is performed.
- (2) The unnecessary DCs, those which no longer serve any customers after performing step 1, are closed.

3.1.4. Variable Fixing

At the end of Lagrangian procedure, a variable fixing technique is employed. This method uses two following rules whose detailed proofs of the validity are presented by Shu et al. [2].

- (1) Let LB and UB denote the current lower and upper bounds on the solution, respectively. If no DC is located at candidate site location $j \in J$ in the Lagrangian solution (i.e., $X_j = 0$ in the optimal solution to (3.8) at some iteration), and if $LB + f_j + \tilde{V}_j > UB$, then no DC will be located at j in any optimal solution for (3.6). Thus, we fix $X_j = 0$.
- (2) If a DC is located at candidate site location $j \in J$ in the Lagrangian solution (i.e., $X_j = 1$ in the optimal solution to (3.8) at some iteration), and if $LB - (f_j + \tilde{V}_j) > UB$, then a DC will be located at j in any optimal solution for (3.6). Thus, we fix $X_j = 1$.

3.2. Genetic Algorithm

In order to find near-optimal solutions for the model in relatively short time, here a solution approach based on genetic algorithm (GA) is developed. GA is a stochastic search and heuristic optimization technique based on the mechanism of natural genetics which has proven to be a very effective heuristic procedure to solve supply chain design problems, particularly problems of large scale [21, 30–34].

GA starts with an initial set of random solution called population. Each solution in the population is called chromosome and each component of chromosome is designated by gene. The chromosomes evolve through successive iterations, called generations. During each generation, the chromosomes are evaluated, using some measures of fitness. To create next generation, new chromosomes (called offspring) are formed by crossover or mutation operators. Crossover operator combines two chromosomes from current generation, while mutation operator modifies a chromosome to form offspring.

A new generation is created by selecting some of current chromosomes (called parents) and offspring based on the fitness values. Also, some chromosomes are rejected so as to keep the population size constant. Fitter chromosomes have higher probabilities of being selected. After several generations, the algorithms converge to the best chromosome, which may represent the optimum or suboptimal solution to the problem [33, 39, 40]. Figure 3 summarizes the steps of the proposed GA. In the following subsections, the developed GA is explained with more details.

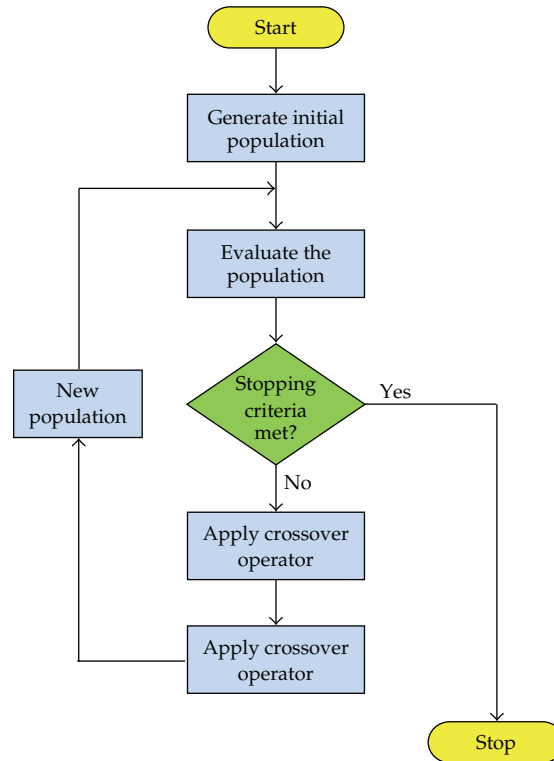


Figure 3: Genetic algorithm procedure.

X_1	X_2	X_3	X_4	Y_1	Y_2	Y_3	Y_4
0	1	0	1	4	2	0	4

Figure 4: Chromosome structure.

3.2.1. Chromosome Representation

In this GA-based approach, each chromosome is indicated as a single-dimensional array. Let n be the number of candidate DCs and m be the number of customers. Then, each chromosome C can be indicated by: $C = (X_j, Y_i) = (X_1, X_2, \dots, X_n, Y_1, Y_2, \dots, Y_m)$, where X_j correspond to the location genes and Y_i correspond to the assignment genes. Location genes indicate where DCs are located and assignment genes represent how customers are assigned to the located DCs, respectively. In particular, if $X_j = 1$, it means that candidate site j is selected as a DC location, while if $X_j = 0$, candidate location j is not chosen as a DC site. Also, $Y_i = j$ represents that customer i is assigned to distribution center j . If customer i is not served at all, the corresponding assignment gene takes the value of 0 (i.e., $Y_i = 0$). For example, in Figure 4, distribution centers are located at 2 and 4. It follows that customers 1 and 4 are assigned to the DC at 4 and customer 2 is allocated to the DC at 2. Also, customer 3 is not served at all.

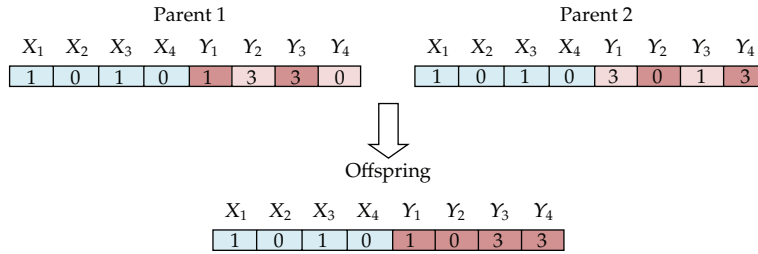


Figure 5: Sample of crossover.

3.2.2. Chromosomes Fitness

The rank-based evaluation function is defined as the objective function (2.6) for the chromosomes. In fact, the objective (2.6) is calculated for each of the chromosomes. Obviously, chromosomes with better values for objective (2.6) will have the better rank.

3.2.3. Crossover Operator

Crossover operator generates offspring by merging parent chromosomes. Let C_k denote the chromosomes of the population for $k = 1, 2, \dots, pop\text{-size}$. In order to determine which of the chromosomes are selected as parents for crossover operation, the following procedure is repeated from $k = 1$ to $pop\text{-size}$: generating a random number r from the interval $[0, 1]$, the chromosome C_k will be selected as a parent provided that $r < P_C$, where the parameter P_C is the probability of crossover. Then, randomly we group the selected parents C'_1, C'_2, C'_3, \dots to the pairs $(C'_1, C'_2), (C'_3, C'_4), \dots$. Without loss of generality, let us explain the crossover operator on each pair by (C'_1, C'_2) .

Crossover operator assigns each customer i in offspring chromosome either to the DC which is allocated to customer i in parent chromosome C'_1 , or to the DC which is assigned to customer i in parent chromosome C'_2 . This occurs randomly and with probability of 0.5. If a customer is allocated to an unselected candidate DC site, a DC is located in that candidate location. A sample of crossover operator is shown in Figure 5.

3.2.4. Mutation Operator

Mutation operator modifies a chromosome to form an offspring. In order to decide which of chromosomes C_k undergo mutation, the following practice is repeated from $k = 1$ to $Pop\text{-size}$: generating a random number r from the interval $[0, 1]$, the chromosome C_k will be selected as a parent provided that $r < P_M$, where the parameter P_M is the probability of mutation. A selected chromosome is modified by one of the two following types of mutation for several times. Each type of mutation is occurred with probability 0.5.

The first type of mutation generates offspring by modifying the assignment genes of parent chromosome. That is, in the first type of mutation, two located DCs are selected randomly; let s and t denote them. Then, if any customer in the parent chromosome is assigned to s , that customer will be assigned to t , and if any customer is assigned to t , it will be allocated to s . The second type of mutation modifies location genes of parent chromosome to form an offspring. In other words, the second type of mutation randomly selects a location

X_1	X_2	X_3	X_4	Y_1	Y_2	Y_3	Y_4
1	0	1	0	1	5	3	1

↓

X_1	X_2	X_3	X_4	Y_1	Y_2	Y_3	Y_4
1	0	1	0	3	5	1	3

Figure 6: Sample of mutation type 1.

X_1	X_2	X_3	X_4	Y_1	Y_2	Y_3	Y_4
1	0	1	0	1	5	3	1

↓

X_1	X_2	X_3	X_4	Y_1	Y_2	Y_3	Y_4
0	1	1	0	2	5	3	2

Figure 7: Sample of mutation type 2.

in which no DC is located; let t denotes it. Next, a DC is selected randomly from the located DCs and is named s . This type of mutation closes distribution center s and locates a DC at t instead. Then, all the customers assigned to distribution center s are allocated to distribution center t . The samples of mutation type 1 and mutation type 2 are illustrated in Figures 6 and 7, respectively.

4. Computational Results

This section summarizes the computational experience with the solution approaches outlined in the previous section. Two sets of experiments were designed. The objective of the first set of experiments was to evaluate the performance of the proposed solution methods in terms of the solution quality and time. The second set of experiments was designed to study the benefits of considering facility disruptions during the supply chain design phase. The developed solution approaches were coded in Visual Basic.Net and executed on Pentium 5 computer with 1.00 GB RAM and 2.00 GHz CPU.

4.1. Experimental Design

The computational experiments were conducted on the 49-node, 88-node, and 150-node datasets described in Daskin [41]. These datasets have been very popular in the literature and have been used in lots of studies to validate the new solution methods [1, 3, 6–8, 24, 27, 28, 30, 34]. The 49-node dataset represents the capitals of the lower 48 United States plus Washington, DC; the 88-node data set represents the 50 largest cities in the 1990 U.S. census along with the 49-node dataset minus duplicates; the 150-node dataset contains the 150 largest cities in the 1990 U.S. census.

For all three data sets, the mean of demand was obtained by dividing the population data given in Daskin [41] by 1000. Fixed costs of locating DCs (f_j) were gained by dividing the fixed cost in Daskin [41] by 10 for the 49-node problem and by 100 for 88-node problem. For the 150-node problem, fixed locating costs were set to 10,000 for all the candidate DC

Table 1: Parameters for Lagrangian relaxation.

Parameter	Value
Initial value of the scalar used to define step size	2
Minimum value of the scalar	10^{-12}
Maximum number of iterations before halving the scalar	12
Maximum number of iterations	400
Initial Lagrange multiplier value	0

Table 2: Parameters for the genetic algorithm.

Parameter	Value
Population size	100
Probability of crossover	0.95
Probability of mutation	0.01
Number of generations	400

locations. We set the per-unit cost to ship from distribution center j to customer i , d_{ij} , to the great-circle distance between these locations. Similar to Daskin et al. [1], the fixed ordering a_j and shipping costs b_j were set to 10 and the variable shipping cost c_j was set to 5 for all DCs. Also, we set the inventory holding cost per unit of product to 1 and set the selling price at each customer to 500. The disruption scenarios were generated randomly. In particular, for each $s \in S$ and for each $j \in J$, r_{sj} randomly was set to 0 with probability of ninety percent, or to a random number from $(0,1]$ with probability of ten percent. The probability of occurrence associated with each scenario was drawn uniformly from $(0,1]$ and then normalized such that the total probability of all the scenarios is equal to 1. Table 1 shows the parameters for the Lagrangian relaxation approach in the computational experiments. These parameters were set similar to Qi et al. [27]. The parameters for the genetic algorithm were set based on the optimal values suggested by Sourirajan et al. [34] and Grefenstette [42]. These parameters are given in Table 2.

4.2. Performance of the Algorithms

Tables 3, 4, and 5 present the computational results with Lagrangian relaxation on 49-node, 88-node, and 150-node problems, respectively. In each table, the first column gives the number of scenarios in the problem. The columns labeled β and θ give the values of β and θ . The columns marked LB and UB give the lower and upper bounds obtained for the profit maximizing model. The last column in each table indicates the percentage gap between the obtained upper and lower bounds and is gained by $((UB-LB)/LB) \times 100$. In all of the cases, the gap was always less than one percent representing that bounds provided by the Lagrangian relaxation approach are very tight and can be relied upon to produce good feasible solutions.

Also, the computational results with genetic algorithm are presented in Tables 6–8. Similarly, in each table, the first column gives the number of scenarios in the problem and the columns labeled β and θ give the values of β and θ . The column marked GA represents the objective value obtained by genetic algorithm. The column labeled UB indicates the upper bound obtained by Lagrangian relaxation for the integrated model. The percentage gap between this upper bound and the objective value obtained by genetic algorithm is

Table 3: Computational results with Lagrangian relaxation for 49-node problem.

	Number of scenarios	β	θ	LB	UB	GAP
(1)	20	0.001	0.1	1085370	1085649	0.026
(2)	20	0.005	0.1	931632	932200.3	0.061
(3)	20	0.005	0.5	922288	922881	0.064
(4)	20	0.005	1	913837	914104.9	0.029
(5)	20	0.005	5	736483	737237.7	0.102
(6)	40	0.001	0.1	1083956	1084224	0.025
(7)	40	0.005	0.1	931431	932001.2	0.061
(8)	40	0.005	0.5	922758	923345.3	0.064
(9)	40	0.005	1	914907	915496.2	0.064
(10)	40	0.005	5	741487	742397.6	0.123
(11)	60	0.001	0.1	1079756	1080051	0.027
(12)	60	0.005	0.1	918411	918992.6	0.063
(13)	60	0.005	0.5	909053	909665	0.067
(14)	60	0.005	1	900592	901211.6	0.069
(15)	60	0.005	5	722305	723255.6	0.131

Table 4: Computational results with Lagrangian relaxation for 88-node problem.

	Number of scenarios	β	θ	LB	UB	GAP
(1)	20	0.001	0.1	2216709	2216907	0.009
(2)	20	0.005	0.1	2186046	2191504	0.249
(3)	20	0.005	0.5	2181345	2182632	0.059
(4)	20	0.005	1	2177170	2183424	0.286
(5)	20	0.005	5	2103086	2109739	0.315
(6)	40	0.001	0.1	2217520	2219700	0.098
(7)	40	0.005	0.1	2187909	2193168	0.240
(8)	40	0.005	0.5	2183342	2189198	0.268
(9)	40	0.005	1	2179227	2185382	0.282
(10)	40	0.005	5	2105254	2111011	0.273
(11)	60	0.001	0.1	2216361	2218739	0.107
(12)	60	0.005	0.1	2184004	2186680	0.122
(13)	60	0.005	0.5	2179289	2185444	0.282
(14)	60	0.005	1	2175124	2181677	0.300
(15)	60	0.005	5	2099659	2109405	0.462

given in the column marked GAP. Tables 6–8 show that the gap does not exceed one percent suggesting that the obtained solutions by genetic algorithm are close to optimal values.

Table 9 summarizes the computational results and compares the performance of Lagrangian relaxation approach with that of genetic algorithm in terms of the solution quality and time. In this table, GAP and CPU time (per second) values are averaged and reported for different values of β and θ . It follows from Table 9 that typically Lagrangian relaxation results in lower values of GAP than genetic algorithm. Thus, the performance of Lagrangian relaxation is better than genetic algorithm from the quality point of view. However, the computational time of genetic algorithm is consistently lower than that of Lagrangian relaxation. This suggests that genetic algorithm is faster than Lagrangian relaxation at solving the model.

Table 5: Computational results with Lagrangian relaxation for 150-node problem.

	Number of scenarios	β	θ	LB	UB	GAP
(1)	20	0.001	0.1	2894910	2901856	0.239
(2)	20	0.005	0.1	2886064	2888441	0.082
(3)	20	0.005	0.5	2876835	2879014	0.076
(4)	20	0.005	1	2868984	2872850	0.135
(5)	20	0.005	5	2754046	2766775	0.460
(6)	40	0.001	0.1	2894672	2896058	0.048
(7)	40	0.005	0.1	2885945	2888322	0.082
(8)	40	0.005	0.5	2876750	2880020	0.114
(9)	40	0.005	1	2868843	2872907	0.141
(10)	40	0.005	5	2752319	2765049	0.460
(11)	60	0.001	0.1	2893617	2935389	1.423
(12)	60	0.005	0.1	2885666	2888043	0.082
(13)	60	0.005	0.5	2876102	2879471	0.117
(14)	60	0.005	1	2867909	2872072	0.145
(15)	60	0.005	5	2748015	2775004	0.973

Table 6: Computational results with genetic algorithm for 49-node problem.

	Number of scenarios	β	θ	GA	UB	GAP
(1)	20	0.001	0.1	1083809	1085649	0.170
(2)	20	0.005	0.1	925985.3	932200.3	0.667
(3)	20	0.005	0.5	919960.8	922881	0.316
(4)	20	0.005	1	913925.2	914104.9	0.020
(5)	20	0.005	5	730852.1	737237.7	0.866
(6)	40	0.001	0.1	1082130	1084224	0.193
(7)	40	0.005	0.1	931904.4	932001.2	0.010
(8)	40	0.005	0.5	922998.5	923345.3	0.038
(9)	40	0.005	1	915301.9	915496.2	0.021
(10)	40	0.005	5	739498.9	742397.6	0.390
(11)	60	0.001	0.1	1078547	1080051	0.139
(12)	60	0.005	0.1	918801.3	918992.6	0.021
(13)	60	0.005	0.5	906391.2	909665	0.360
(14)	60	0.005	1	901080.2	901211.6	0.015
(15)	60	0.005	5	716398.5	723255.6	0.948

4.3. The Benefits of Considering Disruptions in the Supply Chain Design Model

This section compares the performance of two different approaches for designing the supply chain networks. The first approach considers disruptions scenarios during making supply chain design decisions including location and allocation decisions, as we do in this article. That is, the first approach uses the presented model in this study to determine the supply chain design decisions. The second approach, however, makes supply chain design decisions without taking into consideration the disruptions scenarios. In order to determine the supply chain design decisions under the second method, the proposed model by Daskin et al. [1] is

Table 7: Computational results with genetic algorithm for 88-node problem.

	Number of scenarios	β	θ	GA	UB	GAP
(1)	20	0.001	0.1	2207253	2216907	0.435
(2)	20	0.005	0.1	2174020	2191504	0.798
(3)	20	0.005	0.5	2181933	2182632	0.032
(4)	20	0.005	1	2168216	2183424	0.697
(5)	20	0.005	5	2091540	2109739	0.863
(6)	40	0.001	0.1	2219469	2219700	0.010
(7)	40	0.005	0.1	2171824	2193168	0.973
(8)	40	0.005	0.5	2182029	2189198	0.327
(9)	40	0.005	1	2184849	2185382	0.024
(10)	40	0.005	5	2102215	2111011	0.417
(11)	60	0.001	0.1	2199708	2218739	0.858
(12)	60	0.005	0.1	2167173	2186680	0.892
(13)	60	0.005	0.5	2170677	2185444	0.676
(14)	60	0.005	1	2162680	2181677	0.871
(15)	60	0.005	5	2098644	2109405	0.510

Table 8: Computational results with genetic algorithm for 150-node problem.

	Number of scenarios	β	θ	GA	UB	GAP
(1)	20	0.001	0.1	2877662	2901856	0.834
(2)	20	0.005	0.1	2860587	2888441	0.964
(3)	20	0.005	0.5	2876819	2879014	0.076
(4)	20	0.005	1	2861319	2872850	0.401
(5)	20	0.005	5	2741150	2766775	0.926
(6)	40	0.001	0.1	2872745	2896058	0.805
(7)	40	0.005	0.1	2870917	2888322	0.603
(8)	40	0.005	0.5	2878679	2880020	0.047
(9)	40	0.005	1	2859725	2872907	0.459
(10)	40	0.005	5	2747475	2765049	0.636
(11)	60	0.001	0.1	2901035	2935389	1.170
(12)	60	0.005	0.1	2865515	2888043	0.780
(13)	60	0.005	0.5	2862338	2879471	0.595
(14)	60	0.005	1	2850407	2872072	0.754
(15)	60	0.005	5	2747790	2775004	0.981

used. By comparing the total profits under these two approaches, the benefits of considering disruptions in the supply chain design model are implied.

Tables 10, 11, and 12 demonstrate the total profits of the two approaches for the same datasets in Section 4.1. The first column marked the number of scenarios, β and θ give the number of scenarios and the values of β and θ for each instance, respectively. The total profits under the first and second approaches are denoted by TP_1 and TP_2 , respectively. The last column in each table represents the percentage difference between the total profit of the first and that of the second approach. In other words, this column indicates the amount of increase in total profit by following the first approach instead of the second approach for each instance.

It follows from Tables 10–12 that considering facility disruptions during the supply chain design phase can lead to increase in total profit. Particularly, the results show that

Table 9: Performance Comparison between Lagrangian relaxation and genetic algorithm.

Data set	β	θ	Lagrangian relaxation		genetic algorithm	
			Average GAP	Average time (sec.)	Average GAP	Average time (sec.)
49-node	0.001	0.1	0.025904	5.8	0.167293	1.1
	0.005	0.1	0.061807	2.9	0.232634	1.6
	0.005	0.5	0.061807	3.1	0.23796	1.2
	0.005	1	0.05414	2.6	0.018489	1.4
	0.005	5	0.118819	3.2	0.734896	1.3
88-node	0.001	0.1	0.07145	12.6	0.434526	3.4
	0.005	0.1	0.203758	10.9	0.887688	3.8
	0.005	0.5	0.202721	14.3	0.345074	3.1
	0.005	1	0.289501	18.7	0.530557	3.2
	0.005	5	0.350059	8.6	0.596479	3.9
150-node	0.001	0.1	0.570109	58.5	0.93636	5.3
	0.005	0.1	0.082331	41.1	0.782338	5.6
	0.005	0.5	0.102101	42.9	0.239287	5.8
	0.005	1	0.140347	39.2	0.538185	5.9
	0.005	5	0.631033	52.5	0.847475	5.2

Table 10: The benefits of considering disruptions in the supply chain design model for 49-node problem.

	Number of scenarios	β	θ	TP ₁	TP ₂	Profit difference (%)
(1)	20	0.001	0.1	1085370	1017859.99	6.22
(2)	20	0.005	0.1	931632	877317.854	5.83
(3)	20	0.005	0.5	922288	868887.525	5.79
(4)	20	0.005	1	913837	864855.337	5.36
(5)	20	0.005	5	736483	714093.917	3.04
(6)	40	0.001	0.1	1083956	975235.213	10.03
(7)	40	0.005	0.1	931431	848347.355	8.92
(8)	40	0.005	0.5	922758	841001.641	8.86
(9)	40	0.005	1	914907	836682.452	8.55
(10)	40	0.005	5	741487	703819.46	5.08
(11)	60	0.001	0.1	1079756	891230.602	17.46
(12)	60	0.005	0.1	918411	778169.64	15.27
(13)	60	0.005	0.5	909053	773513.198	14.91
(14)	60	0.005	1	900592	776400.363	13.79
(15)	60	0.005	5	722305	645162.826	10.68

the benefit of considering disruptions scenarios in the supply chain design model can be significant up to twenty-eight percent. There are some properties for the presented model in this study which can explain why considering disruptions in the supply chain design phase results in higher profits. First, in the optimal solution DCs are more likely to be opened at locations with lower risks of disruptions to reduce the lost-sales costs due to disruptions. Likewise, when a DC is more likely to be disrupted, fewer customers will be assigned to this DC in the optimal solution and lower lost-sales costs will be incurred. Also, when the possible disruptions are considered during the design of supply chain, fewer customers are selected to be served. That is, the optimal solution will involve more customers not served by any DC, in order to reduce the risk of incurring considerable lost-sales costs.

Table 11: The benefits of considering disruptions in the supply chain design model for 88-node problem.

	Number of scenarios	β	θ	TP ₁	TP ₂	Profit difference (%)
(1)	20	0.001	0.1	2216709	1996589.8	9.93
(2)	20	0.005	0.1	2186046	1998483.25	8.58
(3)	20	0.005	0.5	2181345	1998984.56	8.36
(4)	20	0.005	1	2177170	2001254.66	8.08
(5)	20	0.005	5	2103086	1992253.37	5.27
(6)	40	0.001	0.1	2217520	1907732.46	13.97
(7)	40	0.005	0.1	2187909	1929298.16	11.82
(8)	40	0.005	0.5	2183342	1931384.33	11.54
(9)	40	0.005	1	2179227	1936025.27	11.16
(10)	40	0.005	5	2105254	1929465.29	8.35
(11)	60	0.001	0.1	2216361	1715241.78	22.61
(12)	60	0.005	0.1	2184004	1735846.38	20.52
(13)	60	0.005	0.5	2179289	1722074.17	20.98
(14)	60	0.005	1	2175124	1730093.63	20.46
(15)	60	0.005	5	2099659	1737047.89	17.27

Table 12: The benefits of considering disruptions in the supply chain design model for 150-node problem.

	Number of scenarios	β	θ	TP ₁	TP ₂	Profit difference (%)
(1)	20	0.001	0.1	2216709	1939177.03	12.52
(2)	20	0.005	0.1	2186046	1940553.03	11.23
(3)	20	0.005	0.5	2181345	1945541.61	10.81
(4)	20	0.005	1	2177170	1946825.41	10.58
(5)	20	0.005	5	2103086	1914649.49	8.96
(6)	40	0.001	0.1	2217520	1805726.54	18.57
(7)	40	0.005	0.1	2187909	1808963.16	17.32
(8)	40	0.005	0.5	2183342	1811082.19	17.05
(9)	40	0.005	1	2179227	1815078.17	16.71
(10)	40	0.005	5	2105254	1809044.76	14.07
(11)	60	0.001	0.1	2216361	1586471.2	28.42
(12)	60	0.005	0.1	2184004	1589299.71	27.23
(13)	60	0.005	0.5	2179289	1598072.62	26.67
(14)	60	0.005	1	2175124	1606981.61	26.12
(15)	60	0.005	5	2099659	1617787.26	22.95

It can be concluded that not only the facility location costs affect the location decisions, but also the rates of disruptions at facilities are important factors in determining where to locate DCs. For instance, despite low facility location cost, it is possible that a DC is not opened in a candidate site because of high rates of disruptions. Similarly, the number of customers to be served and the assignments of customers to DCs can be dependent to the rates of disruptions at the DCs. Therefore, significant increase in total profit can be realized, if the facility disruptions are considered in the supply chain design model.

5. Conclusion

This paper has addressed a supply chain design problem where distribution centers are subject to partial and complete disruptions. The problem has been formulated as a nonlinear mixed-integer programming which maximizes the total profit for the whole supply chain. The integrated model simultaneously determines the optimal number and location of DCs, the subset of customers to serve, the assignment of customers to DCs, and the cycle-order quantities at DCs. In order to obtain near-optimal solutions with reasonable computational requirements, two heuristics based on Lagrangian relaxation and genetic algorithm have been presented. Computational results for different data sets have revealed that the proposed solution approaches are effective. Also, it has been demonstrated that the benefits of considering disruptions during supply chain design phase can be significant.

In future, it would be interesting to formulate the problem when suppliers are unreliable. Also, the model can be extended to consider constraints on the maximum capacity of DCs, or on the maximum supply that can be filled by suppliers. Finally, considering routing decisions in the model makes it more useful.

Acknowledgment

The authors would like to acknowledge Mr. Pooramini whose comments improved the paper significantly.

References

- [1] M. S. Daskin, C. R. Coullard, and Z.-J. M. Shen, "An inventory-location model: formulation, solution algorithm and computational results," *Annals of Operations Research*, vol. 110, no. 1–4, pp. 83–106, 2002.
- [2] J. Shu, C.-P. Teo, and Z.-J. M. Shen, "Stochastic transportation-inventory network design problem," *Operations Research*, vol. 53, no. 1, pp. 48–60, 2005.
- [3] Z. J. Max Shen and L. Qi, "Incorporating inventory and routing costs in strategic location models," *European Journal of Operational Research*, vol. 179, no. 2, pp. 372–389, 2007.
- [4] Z.-J. M. Shen, "Integrated supply chain design models: a survey and future research directions," *Journal of Industrial and Management Optimization*, vol. 3, no. 1, pp. 1–27, 2007.
- [5] M. T. Melo, S. Nickel, and F. Saldanha-da-Gama, "Facility location and supply chain management—a review," *European Journal of Operational Research*, vol. 196, no. 2, pp. 401–412, 2009.
- [6] Z. J. M. Shen, C. Coullard, and M. S. Daskin, "A joint location-inventory model," *Transportation Science*, vol. 37, no. 1, pp. 40–55, 2003.
- [7] L. V. Snyder, M. S. Daskin, and C. P. Teo, "The stochastic location model with risk pooling," *European Journal of Operational Research*, vol. 179, no. 3, pp. 1221–1238, 2007.
- [8] L. Ozsen, C. R. Coullard, and M. S. Daskin, "Capacitated warehouse location model with risk pooling," *Naval Research Logistics*, vol. 55, no. 4, pp. 295–312, 2008.
- [9] L. Ozsen, M. S. Daskin, and C. R. Coullard, "Facility location modeling and inventory management with multisourcing," *Transportation Science*, vol. 43, no. 4, pp. 455–472, 2009.
- [10] P. A. Miranda and R. A. Garrido, "Inventory service-level optimization within distribution network design problem," *International Journal of Production Economics*, vol. 122, no. 1, pp. 276–285, 2009.
- [11] Z. Yao, L. H. Lee, W. Jaruphongsa, V. Tan, and C. F. Hui, "Multi-source facility location-allocation and inventory problem," *European Journal of Operational Research*, vol. 207, no. 2, pp. 750–762, 2010.
- [12] S. Park, T. E. Lee, and C. S. Sung, "A three-level supply chain network design model with risk-pooling and lead times," *Transportation Research Part E*, vol. 46, no. 5, pp. 563–581, 2010.
- [13] K. Sweet, "China earthquake hits home for u.s. companies," *Fox Business*, May 2008.

- [14] A. Latour, "Trial by fire: a blaze in Albuquerque sets off major crisis for cell-phone giants-Nokia handles supply chain shock with aplomb as Ericsson of Sweden gets burned- Was Sisu the difference?" *Wall Street Journal*, January 2001.
- [15] D. Leonard, "The only lifeline was the wal-mart," *Fortune*, vol. 152, no. 7, pp. 74–80, 2005.
- [16] L. V. Snyder and M. S. Daskin, "Reliability models for facility location: the expected failure cost case," *Transportation Science*, vol. 39, no. 3, pp. 400–416, 2005.
- [17] O. Berman, D. Krass, and M. B. C. Menezes, "Facility reliability issues in network p -median problems: strategic centralization and co-location effects," *Operations Research*, vol. 55, no. 2, pp. 332–350, 2007.
- [18] M. Lim, M. S. Daskin, A. Bassamboo, and S. Chopra, "A facility reliability problem: formulation, properties, and algorithm," *Naval Research Logistics*, vol. 57, no. 1, pp. 58–70, 2010.
- [19] T. Cui, Y. Ouyang, and Z.-J. M. Shen, "Reliable facility location design under the risk of disruptions," *Operations Research*, vol. 58, no. 4, part 1, pp. 998–1011, 2010.
- [20] F. Liberatore, M. P. Scaparra, and M. S. Daskin, "Analysis of facility protection strategies against an uncertain number of attacks: the stochastic R-interdiction median problem with fortification," *Computers & Operations Research*, vol. 38, no. 1, pp. 357–366, 2011.
- [21] P. Peng, L. V. Snyder, A. Lim, and Z. Liu, "Reliable logistics networks design with facility disruptions," *Transportation Research Part B*, vol. 45, no. 8, pp. 1190–1211, 2011.
- [22] L. V. Snyder, M. P. Scaparra, M. S. Daskin, and R. L. Church, "Planning for disruptions in supply chain networks," in *TutORials in Operations Research*, P. Johnson, B. Norman, and N. Secomandi, Eds., pp. 234–257, INFORMS, Baltimore, Md, USA, 2006.
- [23] L. V. Snyder and M. S. Daskin, "Models for reliable supply chain network design," in *Critical Infrastructure: Reliability and Vulnerability*, A. T. Murray and T. H. Grubescic, Eds., pp. 257–289, Springer, Berlin, Germany, 2007.
- [24] M. B. Aryanezhad, S. G. Jalali, and A. Jabbarzadeh, "An integrated supply chain design model with random disruptions consideration," *African Journal of Business Management*, vol. 4, no. 12, pp. 2393–2401, 2010.
- [25] Q. Chen, X. Li, and Y. Ouyang, "Joint inventory-location problem under the risk of probabilistic facility disruptions," *Transportation Research Part B*, vol. 45, no. 7, pp. 991–1003, 2011.
- [26] L. Qi and Z.-J. M. Shen, "A supply chain design model with unreliable supply," *Naval Research Logistics*, vol. 54, no. 8, pp. 829–844, 2007.
- [27] L. Qi, Z. J. M. Shen, and L. V. Snyder, "The effect of supply disruptions on supply chain design decisions," *Transportation Science*, vol. 44, no. 2, pp. 274–289, 2010.
- [28] Z.-J. M. Shen, "A profit-maximizing supply chain network design model with demand choice flexibility," *Operations Research Letters*, vol. 34, no. 6, pp. 673–682, 2006.
- [29] P. Zipkin, *Foundations of Inventory Management*, McGraw-Hill, Irwin, Calif, USA, 2000.
- [30] J. H. Jaramillo, J. Bhadury, and R. Batta, "On the use of genetic algorithms to solve location problems," *Computers & Operations Research*, vol. 29, no. 6, pp. 761–779, 2002.
- [31] O. Alp, E. Erkut, and Z. Drezner, "An efficient genetic algorithm for the p -median problem," *Annals of Operations Research*, vol. 122, pp. 21–42, 2003.
- [32] T. Drezner, Z. Drezner, and S. Salhi, "Solving the multiple competitive facilities location problem," *European Journal of Operational Research*, vol. 142, no. 1, pp. 138–151, 2002.
- [33] M. B. Aryanezhad, S. G. Jalali, and A. Jabbarzadeh, "An integrated location inventory model for designing a supply chain network under uncertainty," *Life Science Journal*, vol. 8, no. 4, pp. 670–679, 2011.
- [34] K. Sourirajan, L. Ozsen, and R. Uzsoy, "A genetic algorithm for a single product network design model with lead time and safety stock considerations," *European Journal of Operational Research*, vol. 197, no. 2, pp. 599–608, 2009.
- [35] H. Min, H. J. Ko, and C. S. Ko, "A genetic algorithm approach to developing the multi-echelon reverse logistics network for product returns," *Omega*, vol. 34, no. 1, pp. 56–69, 2006.
- [36] J. Current, M. Daskin, and D. Schilling, "Discrete network location models," in *Facility Location: Applications and Theory*, Z. Drezner and H. W. Hamacher, Eds., pp. 83–120, Springer, Berlin, Germany, 2001.
- [37] M. L. Fisher, "The Lagrangian relaxation method for solving integer programming problems," *Management Science*, vol. 27, no. 1, pp. 1–18, 1981.
- [38] M. L. Fisher, "An applications oriented guide to Lagrangian relaxation," *Interfaces*, vol. 15, no. 2, pp. 10–21, 1985.
- [39] M. Gen and R. Cheng, *Genetic Algorithms and Engineering Design*, Wiley, New York, NY, USA, 1996.
- [40] M. Gen and R. Cheng, *Genetic Algorithms and Engineering Optimization*, Wiley, New York, NY, USA, 2000.

- [41] M. S. Daskin, *Network and Discrete Location: Models, Algorithms, and Applications*, Wiley-Interscience Series in Discrete Mathematics and Optimization, John Wiley & Sons, New York, NY, USA, 1995.
- [42] J. J. Grefenstette, "Optimization of control parameters for genetic algorithms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 16, no. 1, pp. 122–128, 1986.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

