JOGO manuscript No.
(will be inserted by the editor)

A Facility Location Model with Safety Stock Costs: Analysis of the Cost of Single-Sourcing Requirements

Semra Ağralı * · **Joseph Geunes** · **Z. Caner Taşkın**

Received: date / Accepted: date

Abstract We consider a supply chain setting where multiple uncapacitated facilities serve a set of customers with a single product. The majority of literature on such problems requires assigning all of any given customer's demand to a single facility. While this single-sourcing strategy is optimal under linear (or concave) cost structures, it will often be suboptimal under the nonlinear costs that arise in the presence of safety stock costs. Our primary goal is to characterize the incremental costs that result from a single-sourcing strategy. We propose a general model that uses a cardinality constraint on the number of supply facilities that may serve a customer. The result is a complex mixed-integer nonlinear programming problem. We provide a generalized Benders decomposition algorithm for the case in which a customer's demand may be split among an arbitrary number of supply facilities. The Benders subproblem takes the form of an uncapacitated, nonlinear transportation problem, a relevant and interesting problem in its own right. We provide analysis and insight on this subproblem, which allows us to devise a hybrid algorithm based on an outer approximation of this subproblem to accelerate the generalized Benders decomposition algorithm. We also provide computational results for the general model that permit characterizing the costs that arise from a single-sourcing strategy.

Keywords Mixed-integer nonlinear programming · stochastic location-inventory problem · generalized Benders decomposition · outer approximation

S. Ağralı
Department of Industrial Engineering, Bahçeşehir University, Beşiktaş, İstanbul, Turkey
Tel.: +90-212-3810887
Fax: +90-212-3810550
E-mail: semra.agrali@bahcesehir.edu.tr

J. Geunes
Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL, USA
E-mail: geunes@ise.ufl.edu

Z. C. Taşkın
Department of Industrial Engineering, Boğaziçi University, Bebek, İstanbul, Turkey
E-mail: caner.taskin@boun.edu.tr

1 Introduction and Motivation

Classical transportation problems determine the assignment of customers to supply facilities in order to minimize total transportation cost while obeying supply limits and meeting (deterministic) customer demands. In the absence of supply capacities, an optimal solution exists for the transportation problem such that each customer's demand is assigned entirely to a single supply facility. The classical uncapacitated facility location problem (UFLP) contains an embedded transportation subproblem, with the addition of fixed costs for open supply facilities (see, e.g., [1, 10]). Because this problem has a concave cost objective function (such that an extreme point optimal solution exists), we again find that an optimal solution for the UFLP exists such that a given customer's demand is entirely assigned to a single supply facility. More recent work considers practical generalizations of this class of problems that account not only for fixed operating and variable assignment costs, but also for inventory-related costs at facilities. In particular, when we consider contexts with uncertain demands, it is important to consider the impacts of safety stock costs.

Chopra and Meindl [2] discuss general trends in supply chain costs as a function of the number of facilities. For example, it is clear that an increase in the number of facilities in a supply chain network results in a corresponding increase in facility costs. Reducing the number of facilities, however, tends to increase outbound transportation costs, which must be balanced against facility and inventory costs. Similarly, Chopra and Meindl [2] note that an increase in the number of facilities tends to increase total supply chain inventory costs due to the need to increase total system-wide safety stock in order to meet customer service level expectations. Conversely, a reduction in the number of facilities that hold safety stock permits a reduction in total safety stock cost as a result of the risk-pooling benefits from aggregating safety stock in fewer locations. As our results later illustrate, aggregation of safety stock at fewer location is not necessarily required to gain risk-pooling benefits. That is, it is possible to increase the number of supply facilities without a corresponding increase in safety stock cost, even while maintaining a prescribed cycle service level at each facility. As we later discuss, these risk-pooling benefits arise from splitting customer demands among facilities and mixing multiple customer demands within a facility.

Because safety stock costs represent a non-trivial component of overall facility-related costs, recent literature has recognized the need to account for safety stock costs when making facility location decisions (e.g., [32]). The majority of this work, however, continues to enforce single-sourcing restrictions, which are optimal for the UFLP and uncapacitated transportation problems embedded in these larger inventory-location problems. Unfortunately, safety stock costs cannot be represented, in general, as a linear or concave function of the assignment decision variables. Thus, imposing single-sourcing requirements on such inventory-location problems may be suboptimal when compared to the problem in the absence of this requirement. Our primary goal in this paper is, therefore, to improve our understanding of the degree of loss that may result from enforcing a single-sourcing requirement.

Clearly there are some benefits to enforcing single-source requirements, although these benefits are typically difficult to quantify. From a practical standpoint, customers often prefer having a single point of contact for delivery and problem resolution. Similarly, suppliers face lower coordination complexity under a single-sourcing arrangement. Algorithmically, heuristic solution approaches are often easier to construct because of the combinatorial nature of solutions to problems that use single-sourcing requirements.

In contrast, in the absence of single sourcing, a customer has a built-in backup plan when their demand is split among multiple sources, and one of the sources is unable to deliver. With our goal of understanding the costs of single-sourcing in mind, we address the following problem: Given a set of supply facilities, each with some fixed location cost, and a set of customers, each with uncertain demand, determine which supply facilities to open, which customers to assign to which supply facilities and what level of inventory to hold in order to minimize total location, assignment and safety stock costs, while achieving specified service levels and obeying a pre-determined limit on the number of facilities that can supply any given customer.

Note that when the limit on the number of facilities that can supply any given customer equals one, we have the single-sourcing constraint. When this limit equals N (where N is the number of facilities), we effectively have no limit on the number of suppliers that can serve a customer. This problem falls in the class of mixed-integer nonlinear programming problems and is NP-hard (by virtue of generalizing the UFLP). Shen et al. [32] consider a similar joint location-inventory problem with a single-sourcing requirement that minimizes the cost of facility location, transportation, and holding working process inventory and safety stock. Their model is similar to ours, except that we do not require single sourcing and our model includes a cardinality constraint on the number of sources that can supply a customer. Interestingly, when single-sourcing is required and customer demands are normally distributed, the expression typically used for safety stock cost is concave in the assignment decision variables (when we consider the continuous relaxation of these assignment variables). When single sourcing is not required, however, this expression is instead convex, destroying the concavity of the objective function. Thus, the problem studied by Shen et al. [32] contains structural properties that are lost when the single-sourcing requirement is dropped. França and Luna [12] also study a similar problem where demand splitting is allowed (i.e., when a customer's demand may be split among multiple supply facilities). Instead of considering inventory-related costs at the supplier echelon, however, they consider inventory holding and shortage costs at the customer stage, and provide a generalized Benders decomposition algorithm to solve the problem.

In this paper, we first define and formulate a general model for assigning customers to supply facilities when supplier safety stock costs are considered, demand splitting is permitted, and customer demand distributions are approximated by a normal distribution (as in [32]). We analyze the special case with zero fixed facility costs, which results in an interesting and practically relevant transportation problem with safety stock costs. We demonstrate important properties of optimal solutions for special cases of this class of transportation problems that, in some cases, lead to closed-form solutions. Moreover, these optimal solution properties provide insight on effective ways to manage risk due to uncertain demand in supply chains. We provide a generalized Benders decomposition algorithm and an acceleration strategy to solve the general problem with fixed supply-facility operating costs. We then discuss the results of an empirical study intended to characterize the cost of single-sourcing requirements.

The rest of this paper is organized as follows. Section 2 next reviews related literature on location-inventory problems. We define the general problem and model formulation in Section 3, and discuss solution methods for special cases in which no fixed cost component exists. Then we present the generalized Benders decomposition algorithm in Section 4 and propose a hybrid algorithm that significantly accelerates the generalized Benders approach. Section 5 discusses the results of our computational study. Finally, concluding remarks are provided in Section 6.

2 Literature Review

Since this paper addresses a location-inventory model, the literature on both facility location and inventory theory is relevant to our work. We thus consider past work in both of these areas, as well as at the intersection of these areas. In the classical facility location problem, the aim is to determine locations of facilities and assignments of retailers to these facilities that minimize the fixed facility location and transportation costs. Thus, inventory related costs are not addressed. We refer the reader to [4, 6, 19, 25, 34] for a comprehensive review of facility location problems. On the other hand, the inventory theory literature typically assumes that location decisions have been made beforehand, and, based on this assumption, it evaluates inventory related decisions. The aim is therefore to find the policy that minimizes inventory related costs while meeting appropriate service levels at distribution centers or retailers (for examples, please see [39]).

Joint location-inventory models have gained increased attention recently (see [22, 23, 24, 26, 30, 31, 32, 37]). The problem analyzed by Shen et al. [32] is closely related to our work. In particular, Shen et al. [32] consider a joint location-inventory problem, where multiple retailers−each with stochastic demand−are assigned to distribution centers (DCs). Because of uncertain demand, some amount of safety stock must be carried at distribution centers. In their model, they enforce a single-sourcing requirement, i.e., each customer's demand must be assigned to a single DC. Shu et al. [33] study a similar problem with one supplier and multiple retailers, where each retailer can serve as a distribution center in order to achieve risk pooling benefits.

The solution methods applied to these location-inventory models typically depend on the form of the objective function. The form of the objective function, in turn, depends on the decision variable restrictions. For instance, if we have binary assignment variables and an objective function that uses the squared values of these binary variables, then these squared terms can be linearized by simply replacing them with their original binary values (since $x = x^2$ for binary variables). This affects the convexity of the safety stock cost component of the objective function and, therefore, the solution techniques that can be successfully applied. We model our problem as a mixed-integer nonlinear programming problem with continuous assignment variables. We, therefore, need to consider solution techniques relevant to mixed-integer nonlinear programming problems in general, and location-inventory problems in particular.

Recently, Ozsen et al. [27] studied a logistics system with a single plant and a set of capacitated warehouses that serve as intermediaries between the plant and a set of retailers, each of whom faces stochastic demand, which is Poisson distributed. They assume that warehouses order a product from a single plant and carry safety stock in order to meet appropriate service levels, and do not require single sourcing. Their model assumes that each unit of retailer demand is randomly assigned to one of a number of warehouses permitted to serve the associated retailer. The resulting model is a mixed integer nonlinear program (MINLP) with an objective function that is neither convex nor concave, and they propose a Lagrangian relaxation solution algorithm for solving the model. We also relax the single-sourcing requirement by allowing a customer's demand to be split among supply facilities if it is economical to do so. However, while Ozsen et al. [27] use a policy that randomly assigns each unit of retailer demand to one of its supply facilities, our model assigns a predetermined fraction of each period's demand to a supply facility. Moreover, their model has an additional supply stage from which DCs order replenishment batches, and they propose a Lagrangian relax-

ation algorithm, while we propose an exact algorithm that uses generalized Benders decomposition. Our modeling approach also departs from theirs in our demand distribution assumptions, i.e., we assume normality of retailer demands, while they assume Poisson. Their Poisson assumption leads to an important and nontrivial difference in the functional form of the safety stock cost at each facility with respect to our model, which is closely related to our discussion in the previous paragraph. In particular, their approach leads to a safety stock term that is concave in the assignment variables, while our safety stock function is convex in these variables. Thus, in addition to differences in practical operational assumptions, our model differs from theirs in the mathematical structure of the resulting optimization problem.

Lagrangian relaxation based algorithms have been widely used in the location-inventory literature for problems that require single sourcing. Daskin et al. [5] consider a problem similar to the one addressed in Shen et al. [32], where they account for both working inventory and safety stock cost terms. They model this problem as a nonlinear integer programming problem with binary assignment variables, and propose a Lagrangian relaxation solution algorithm. Similarly, Sourirajan et al. [36] apply Lagrangian relaxation to a problem in which a production facility replenishes a single product at multiple retailers. Their model determines the DC locations that minimize total location and inventory costs. Snyder et al. [35], Ozsen et al. [26] and Miranda and Garrido [20] also propose solution methods based on Lagrangian relaxation for mixed-integer nonlinear models. However, each of these papers assumes that single sourcing is required. Moreover, Lagrangian relaxation based solution methods do not provide strictly better solutions than the continuous relaxation for several important special cases of the problem we define in this paper (because of the so-called integrality property; see [14]).

Several heuristic solution methods have also been proposed in the literature for location-inventory problems. Erlebacher and Meller [9] consider a problem where products are distributed from plants to DCs and from DCs to retailers. Their aim is to minimize the sum of the fixed operating costs of open DCs, inventory holding costs at DCs, total transportation costs from plants to DCs, and transportation costs from DCs to customers. DCs and customers are located on a grid, and each customer must be assigned to a single DC; thus demand splitting is not allowed. They propose a location-allocation heuristic that uses the better solution obtained using two different approaches. The first approach assigns each customer to its closest DC and then reduces the number of DCs by greedily reassigning customers to other DCs, until reaching a predetermined number of open DCs. The second approach starts by assigning one customer to each open DC (where the number of open DCs equals a predetermined number), and then adds the remaining (unassigned) customers to DCs until all customers are assigned.

As we have noted, our solution method uses generalized Benders decomposition (see [13]), which has been used effectively for certain classes of mixed-integer nonlinear programming problems [15, 18, 21]. For example, Hoc [16] considered a transportation and computer communication network design problem with a budget constraint. Hoc [16] formulated this problem as a mixed-integer nonlinear programming model and proposed an approach using generalized Benders decomposition. França and Luna [12] also proposed a similar algorithm for a location-inventory problem that is closely related to our work. In their model, they allow backordering with an associated penalty function. Their model considers inventory holding cost at the retail level, whereas our

model takes the supplier's point of view, considering inventory costs at the supplier level.

Benders decomposition is a powerful solution algorithm that has been applied to many cases; however, as noted by [28] and [17], the straigthforward application of the classical Benders decomposition algorithm leads to slow convergence in some cases. Most acceleration methods are related to the generation of cuts and their properties. Magnanti and Wong [17] propose an acceleration method that is based on selecting the best optimal solution out of alternative optimal solutions of the subproblem, if any, such that the generated Benders cut is pareto optimal. Saharidis et al. [29] observe that the cuts produced by classical Benders algorithm are usually low-density cuts, meaning that the number of decision variables of the master problem used in these cuts are small, which have limited effect on stregthening the master problem. They propose a new strategy in which multiple low-density cuts are produced instead of a single cut at every iteration of the algorithm, which improves the efficiency of the algorithm significantly. Generation of multiple cuts is also proposed by Saharidis and Ierapetritou [28]. Their strategy is effective in cases where the number of feasibility cuts produced is more than the optimality cuts.

Solving the master and subproblem can also be time consuming in some algorithms. Cote and Laughton [3] propose an acceleration strategy in which, instead of solving an integer program at every step, they relax the integrality constraint of the master problem and solve its LP relaxation. Then, they apply a heuristic to determine when to force integrality constraints in the master problem in order to guarantee convergence. Zakeri et al. [38] suggest an algorithm that can be used to accelerate Benders decomposition when solving the subproblem is the main issue related to the convergence speed of the algorithm. They propose an inexact cut algorithm, in which cuts are not obtained from an ectreme point solution of the subproblem, but instead they used primal-dual-interior point algorithm to obtain a feasible dual solution, which will yield a valid cut. As we will explain in Section 4, we use a similar idea to accelerate the generalized Benders algorithm proposed in this paper. The next section formally defines our problem, provides the mathematical model and analyzes two special cases.

3 Problem Definition and Mathematical Model

We consider a set $J = \{1, \ldots, N\}$ of potential supply facility locations, indexed by j, such that opening a supply facility at location j results in a fixed cost of F_j for all $j \in J$. We wish to satisfy the demand of a set $I = \{1, \ldots, M\}$ of customers, indexed by i, using some subset of the open facilities. Each customer has a random demand of d_i per time period, and we assume that successive demands in different time increments are independent and identically distributed with mean μ_i and variance σ_i^2. Each supply facility requires achieving a prespecified service level which is supply-facility-dependent. Because customer demands are random, each supply facility carries some amount of safety stock to achieve this service level. The parameters and the decision variables used in the model are as follows.

Parameters

I	set of customers, i.e., $I = \{1, 2, \ldots, M\}$, indexed by i
J	set of supply facilities, i.e., $J = \{1, 2, \ldots, N\}$, indexed by j
c_{ij}	cost of assigning customer i to facility j
$\hat{c}_{ij}$	cost per unit of flow from facility j to customer i
h_j	annual cost of holding a unit of inventory at supply facility j
d_i	random variable for customer i demand per year
μ_i	expected value of d_i
σ_i	standard deviation of d_i
D_j	$\sum_{i \in I} d_i x_{ij}$, i.e., total demand allocated to supply facility j per year
F_j	annualized fixed cost of opening supply facility j
S_j	stock level at supply facility j at the beginning of a year (we assume zero supply lead time)
N_i	maximum number of supply facilities that may serve customer i.

Decision Variables

x_{ij}	proportion of customer i demand allocated to supply facility j
t_{ij}	1 if any supply is sent to customer i from supply facility j; 0 otherwise
y_j	1 if supply facility j is opened; 0 otherwise

If we assign the fraction x_{ij} of customer i's demand to supply facility j, then the expected assignment cost equals $c_{ij}x_{ij}$, where $c_{ij} = \hat{c}_{ij}\mu_i$. We assume that all customer demands are independent and normally distributed. Note that the demand seen by supply facility j in a time period is normally distributed with mean $\mu(j) = \sum_{i \in I} \mu_i x_{ij}$ and variance $\sigma^2(j) = \sum_{i \in I} \sigma_i^2 x_{ij}^2$, i.e., $D_j \sim N(\mu(j), \sigma^2(j))$.

We assume that supply facility j follows a base stock policy, and orders up to a stock level S_j at the beginning of every period, such that $\Pr\{D_j \leq S_j\} = \delta_j$; let $z_j^\delta = \frac{S_j - \mu(j)}{\sigma(j)}$ denote the corresponding z value, i.e., $\Phi(z_j^\delta) = \delta_j$. The expected annual safety stock cost at supply facility j is then given by $h_j z_j^\delta \sqrt{\sum_{i \in I} \sigma_i^2 x_{ij}^2}$. This set of assumptions is consistent with situations in which the supply facility corresponds to a distribution center that receives regular periodic shipments (e.g., weekly) from external suppliers, and is required to meet prespecified service level targets. Observe that although the demands seen by different facilities in a period may be correlated, because we assume that facilities do not share inventory, this correlation does not affect the stock level set at a facility. That is, each facility independently manages stock at its own facility based on the distribution of demand it observes, and there is not mechanism to centrally use information regarding demand correlation at different facilities within a period to better manage stock levels.

We wish to decide which supply facilities to open and how to allocate the demand of each customer i to at most N_i of these open supply facilities in order to minimize the total expected cost. We formulate this location-inventory problem (LIP) as follows:

$$\text{(LIP)} \qquad Z = \text{Minimize} \qquad \sum_{j\in J} F_j y_j + \sum_{i\in I}\sum_{j\in J} c_{ij}x_{ij} + \sum_{j\in J} h_j z_j^{\delta}\sqrt{\sum_{i\in I}\sigma_i^2 x_{ij}^2} \tag{1}$$

$$\text{Subject to} \qquad \sum_{j\in J} x_{ij} \geq 1, \qquad \forall i \in I, \tag{2}$$

$$\sum_{j\in J} t_{ij} \leq N_i, \qquad \forall i \in I, \tag{3}$$

$$0 \leq x_{ij} \leq t_{ij} \leq y_j, \qquad \forall i \in I, j \in J, \tag{4}$$

$$y_j, t_{ij} \in \{0,1\}, \qquad \forall i \in I, j \in J. \tag{5}$$

The objective function (1) minimizes the sum of the fixed cost of locating supply facilities, the assignment and variable cost from supply facilities to customers, and the safety stock costs. Constraint set (2) ensures that each customer's demand is fully assigned to supply facilities. Note that this constraint will be satisfied at equality in an optimal solution. Constraint set (3) limits the number of supply facilities that can serve customer i to at most N_i. Constraint set (4) permits assigning customer demand only to open supply facilities, forces y_j to 1 if $t_{ij} = 1$ and t_{ij} to 1 if $x_{ij} > 0$. This constraint also ensures nonnegativity of the assignment proportion variables (x_{ij}'s). Constraint set (5) reflects the integrality requirements.

Letting $\phi(x) = \sum_{i\in I}\sum_{j\in J} c_{ij}x_{ij} + \sum_{j\in J} h_j z_j^{\delta}\sqrt{\sum_{i\in I}\sigma_i^2 x_{ij}^2}$, the following lemma helps in characterizing the structure of the objective function of (LIP).

Lemma 1 *$\phi(x)$ is convex in x.*

Proof: Please see the Appendix. □

Lemma 1 implies that (LIP) becomes a convex program for given y_j and t_{ij} variables. We will use this fact later when constructing a Benders decomposition algorithm. Before discussing a solution technique for the general model, we would like to analyze two special cases of (LIP). Both of these special cases assume that locations are fixed, or equivalently, a fixed value of the vector of y_j variables, which we denote by $\tilde{\mathbf{y}}$ (note that this is equivalent to the assumption of zero fixed costs). These special cases also assume that $N_i = N$ for all $i = 1, \ldots, N$, which permits dropping constraint set (3) and the t-variables from the formulation. The resulting problem is an uncapacitated transportation problem with safety stock costs which, to the best of our knowledge, has not been considered in the literature. While the resulting problem is a convex program (and is therefore readily solved using commercial optimization packages), the analysis of particular special cases of this problem class leads to some interesting structural properties of optimal solutions, and provides insight on managing the tradeoffs between transportation and safety stock costs.

3.1 Nonlinear transportation problem

This section analyzes two special cases of the uncapacitated nonlinear transportation problem (where all locations decisions are fixed and no cardinality constraint exists on the number of suppliers that can serve a customer). The first special case assumes

identical customer variances and supply-facility-invariant costs, while the second special case considers a two-by-two problem with specially structured assignment and holding costs that lead to a simple closed-form optimal solution.

3.1.1 Identical supply costs and customer variances

We first consider a special case with identical supply facilities (in terms of supply facility costs) where customer demand variances are identical. For this special case and the one discussed in the following subsection, we assume that locations are fixed, which results in an uncapacitated transportation problem with safety stock costs.

By Lemma 1 we know that the objective function of this special case is a convex function of x. Since all of the constraints of (LIP) are linear in x, the problem with zero fixed costs for facilities is a convex programming problem such that the KKT conditions are necessary and sufficient for optimality for this special case (note that any feasible solution such that $\sum_{i\in I} x_{ij} = 0$ violates the differentiability assumption required for application of the KKT conditions at the associated point; however, we are able to consider such solutions separately in our analysis).

For this special case, we assume the assignment cost is customer-specific and equal to c_i for customer i, i.e., $c_{ij} = c_i$ for all $j \in J$ and for each customer i. We will refer to cases in which transportation costs are facility invariant as cases with symmetric transportation costs. We also assume that the supply facility unit holding costs and required cycle service levels are identical for all supply facilities, and that all customer demand variances are equal, i.e., $h_j = h$ and $z_j^\delta = z^\delta$ for all $j \in J$ and $\sigma_i^2 = \sigma^2$ for all $i \in I$. Letting μ and β denote the vectors of KKT multipliers for the assignment constraints (2) and nonnegativity constraints on the x_{ij} variables, we next analyze the KKT conditions for this special case, which can be written as follows.

$$c_i + hz^\delta \sigma \frac{x_{ij}}{\sqrt{\sum_{i\in I} x_{ij}^2}} - \mu_i - \beta_{ij} = 0, \qquad \forall i \in I, j \in J, \tag{6}$$

$$\mu_i \left(1 - \sum\nolimits_{j\in J} x_{ij}\right) = 0, \qquad \forall i \in I, \tag{7}$$

$$\beta_{ij} x_{ij} = 0, \qquad \forall i \in I, j \in J, \tag{8}$$

$$\sum\nolimits_{j\in J} x_{ij} \geq 1, \qquad \forall i \in I, \tag{9}$$

$$\mu_i, \beta_{ij}, x_{ij} \geq 0, \qquad \forall i \in I, j \in J. \tag{10}$$

Given a solution and any supply facility j, let $I(j)$ denote the set of customers such that $x_{ij} > 0$. Similarly, denote $J(i)$ as the set of facilities such that $x_{ij} > 0$. The following theorem characterizes the structure of optimal solutions for this special case.

Theorem 1 *Any feasible solution such that*

1. $x_{ij} = \frac{1}{\omega_j}$ *for some finite* $\omega_j \geq 1$ $\forall j \in J$, $i \in I(j)$ *(with* $x_{ij} = 0$ $\forall i \notin I(j)$*); and*
2. $\sum_{j\in J(i)} \frac{1}{\omega_j} = 1$ *for all* $i \in I$

satisfies the KKT conditions, and is therefore optimal for the special case we have described.

Proof: Please see the Appendix. □

Theorem 1 implies that any *balanced* solution is optimal under identical supply costs and identical customer variance values. That is, provided that all customers assigned to a supply facility have an equal fraction of their expected demand allocated to the supply facility, the solution is optimal. Thus, for example, an optimal solution exists such that all customers are assigned to a single supply facility, which is consistent with the well known use of inventory aggregation to obtain safety stock risk pooling benefits. Theorem 1 illustrates that we can obtain the same degree of risk pooling benefits in a number of different ways, without requiring inventory aggregation. That is, given a problem with N facilities and N customers, for example, a solution such that all N facilities are open, and $\frac{1}{N}$ of each customer's demand is allocated to each open facility achieves the same degree of risk pooling benefits of aggregating all inventory at a single facility (for this special case). This illustrates the fact that one can achieve risk pooling benefits without physical aggregation by splitting customers' demands among different facilities, and mixing the demands of multiple customers within a facility. Clearly, when accounting for fixed costs of identical facilities, the solution that aggregates all customers at one facility (a single-sourcing solution) is preferred when all customers and facility costs are identical. When neither facilities nor customers are identical, however, solutions that require single sourcing are often suboptimal, as we later show in our computational results section, and as the special case discussed in the following subsection illustrates.

3.1.2 Specially structured assignment and holding costs

We next consider a specially structured case with two suppliers and two customers. For this special case, we assume that facility holding costs and service levels are equal, as are customer variances, i.e., $h_j = h$ and $z_j^\delta = z^\delta$ for $j = 1, 2$ and $\sigma_i^2 = \sigma^2$ for $i = 1, 2$. Then, letting $H = hz^\delta\sigma$, we assume the following assignment cost relationship holds for some α between 0 and 1:

$$c_{11} = c_{12} + Hg(\alpha), \tag{11}$$

$$c_{22} = c_{21} + Hg(\alpha), \tag{12}$$

where $g(\alpha) = \frac{(1-2\alpha)}{\sqrt{\alpha^2+(1-\alpha)^2}}$. Note that this permits values of $c_{11} \in [c_{12} - H, c_{12} + H]$ and $c_{22} \in [c_{21} - H, c_{21} + H]$. Observe that when $\alpha = \frac{1}{2}$ we have a symmetric transportation cost instance with $c_{11} = c_{12}$ and $c_{22} = c_{21}$, which results in the special case in which assignment costs are facility independent (as in the special case discussed in the previous subsection). For the two-by-two special case in which facility holding costs and customer variances are equal, and assignment costs obey (11) and (12), we have the following proposition.

Proposition 1 *For a two-supplier, two-customer problem instance with identical supplier holding costs, service levels, and customer demand variances, when the assignment costs obey (11) and (12), an optimal solution exists such that* $x_{11} = x_{22} = \alpha$ *and* $x_{12} = x_{21} = 1 - \alpha$*, with minimum cost* $c_{11} + c_{22} + 2H\frac{\alpha}{\sqrt{\alpha^2+(1-\alpha)^2}} = c_{12} + c_{21} + 2H\frac{(1-\alpha)}{\sqrt{\alpha^2+(1-\alpha)^2}}$.

Proof: Please see the Appendix.

Observe that when $\alpha = \frac{1}{2}$, the symmetric cost case, the optimal cost equals $c_{12} + c_{22} + \sqrt{2}H = c_{11} + c_{21} + \sqrt{2}H = c_{11} + c_{22} + \sqrt{2}H = c_{12} + c_{21} + \sqrt{2}H$. In this case, any one of the following solutions is optimal: $(x_{11}, x_{12}, x_{21}, x_{22}) = \left(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right)$; $(x_{11}, x_{12}, x_{21}, x_{22}) = (0, 1, 0, 1)$; $(x_{11}, x_{12}, x_{21}, x_{22}) = (1, 0, 1, 0)$. This case is consistent with the special case covered in the previous section, where an optimal solution exists that allocates $\frac{1}{\omega_j}$ of each customer's demand to each active facility, where ω_j is the number of customers assigned to facility j. When $\alpha = 1$ ($\alpha = 0$), an optimal solution sets $(x_{11}, x_{12}, x_{21}, x_{22}) = (1, 0, 0, 1)$ ($(x_{11}, x_{12}, x_{21}, x_{22}) = (0, 1, 1, 0)$) with an optimal cost of $c_{11} + c_{22} + 2H$ ($c_{12} + c_{21} + 2H$). In this case, the difference in transportation cost does not offset any benefits from risk pooling. While in each of the cases with $\alpha \in \left\{0, \frac{1}{2}, 1\right\}$ an optimal single-sourcing solution exists, the following corollary shows that this is not the case for the remaining values of α on the interval $[0, 1]$.

Corollary 1 *For the two-supplier, two-customer problem class described, the difference between the objective function value of the minimum-cost single-sourcing solution and that of the minimum-cost solution with demand splitting equals $H \times \rho(\alpha)$, where* $\rho(\alpha) = \min\left\{2\left(1 - \frac{\max\{\alpha, 1-\alpha\}}{\sqrt{\alpha^2 + (1-\alpha)^2}}\right); \sqrt{2} - \frac{1}{\sqrt{\alpha^2 + (1-\alpha)^2}}\right\}$.

Proof: Please see the Appendix.

Figure 1 illustrates the value of $\rho(\alpha)$ for $\alpha \in [0, 1]$. We can show that the peak values occur at the values of α such that the terms in the minimum operator given in the corollary are equal. This occurs at $\alpha = 0.2725$ and $\alpha = 0.7275$, where $\rho(\alpha) = 12.7\%$. At either of these values of α the minimum cost single-sourcing solution exceeds the minimum possible cost by $0.127H$, while the actual percentage cost increase associated with single sourcing depends on the transportation and holding cost parameter values. This analysis illustrates the fact that single-sourcing solutions are either optimal or close-to-optimal when transportation costs are symmetric (as is the case when $\alpha = \frac{1}{2}$) or severely asymmetric (as is the case when $\alpha = 0$ or 1). In the former case, multiple optimal solutions exist (using either one or two facilities) while in the latter case, a single optimal solution exists that uses the dominant facility (in terms of lower transportation costs). For intermediate cases, however (when transportation costs are neither symmetric nor grossly asymmetric), the cost of a single-sourcing strategy can exceed that under a demand splitting strategy by a non-trivial amount. Our computational tests on the general model with location decisions (and associated costs), presented later in Section 5, illustrate this phenomenon further, by showing cost increases associated with single sourcing on the order of $2 - 7\%$.

4 Solution Algorithms for (LIP)

This section returns to the general (LIP) model and provides an effective solution approach for this problem class. We present our solution algorithm in two sections. We start by reformulating problem (LIP) in a form that is amenable to applying generalized Benders decomposition algorithm. Then, we use an outer approximation approach to solve the subproblem that results in our hybrid Benders decomposition/outer approximation approach.

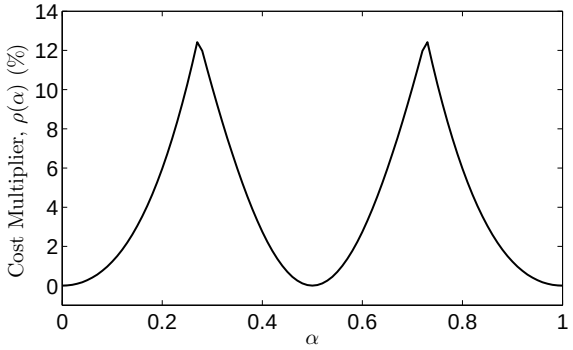

Fig. 1 Cost increase multiplier for single-sourcing as a function of α.

4.1 A Generalized Benders Decomposition Approach for (LIP)

This subsection provides the generalized Benders decomposition algorithm that we propose to solve (LIP). Recall that for a fixed location vector $\tilde{\mathbf{y}}$ and a feasible binary assignment vector $\tilde{\mathbf{t}}$, from Lemma 1, we know that the remaining problem is a convex program. Let us temporarily fix the location vector at $\tilde{\mathbf{y}}$ and the binary assignment vector at $\tilde{\mathbf{t}}$, such that constraints (3), (4) and (5) admit a feasible solution in the x_{ij} variables. Then the associated restricted problem becomes

$$
\begin{aligned}
(\mathrm{LIP}(\tilde{\mathbf{t}},\tilde{\mathbf{y}})) \qquad \text{Minimize} \quad & \sum_{j\in J} F_j\tilde{y}_j + \sum_{i\in I}\sum_{j\in J} c_{ij}x_{ij} + \sum_{j\in J} h_j z_j^{\delta}\sqrt{\sum_{i\in I}\sigma_i^2 x_{ij}^2} \\
\text{Subject to} \quad & \sum_{j\in J} x_{ij} \geq 1, \qquad \forall i \in I, \\
& 0 \leq x_{ij} \leq \tilde{t}_{ij}, \qquad \forall i \in I, j \in J. \qquad (13)
\end{aligned}
$$

Note that the fixed-charge component, $\sum_{j\in J} F_j\tilde{y}_j$, in the objective function is a constant for a given vector $\tilde{\mathbf{y}}$. Similarly, the right-hand-side value of each constraint in set (13) is either 0 or 1, depending on the value of $\tilde{t}_{ij}$. We also note that (LIP($\tilde{\mathbf{t}}$, $\tilde{\mathbf{y}}$)) is feasible if and only if $\sum_{j\in J}\tilde{t}_{ij} \geq 1$ for all $i \in I$.

We can then write our original problem (LIP) in the space of the vector of t_{ij} and y_j variables as

$$
\begin{aligned}
(\mathrm{LIP}') \qquad \text{Minimize} \quad & \sum_{j\in J} F_j y_j + v(t) \\
\text{Subject to} \quad & \sum_{j\in J} t_{ij} \leq N_i, \qquad \forall i \in I, \\
& \sum_{j\in J} t_{ij} \geq 1, \qquad \forall i \in I, \qquad (14) \\
& t_{ij} \leq y_j, \qquad \forall i \in I, j \in J, \\
& t_{ij}, y_j \in \{0,1\}, \qquad \forall i \in I, j \in J,
\end{aligned}
$$

where, for any given vector $\mathbf{t}$, the value $v(\mathbf{t})$ is determined by the following subproblem (LISP):

$$\begin{aligned}
\text{(LISP)} \qquad v(\mathbf{t}) = \text{Minimize} \quad & \sum_{i\in I}\sum_{j\in J} c_{ij}x_{ij} + \sum_{j\in J} h_j z_j^{\delta}\sqrt{\sum_{i\in I}\sigma_i^2 x_{ij}^2} \\
\text{Subject to} \quad & \sum_{j\in J} x_{ij} \geq 1, \qquad \forall i \in I, \\
& 0 \leq x_{ij} \leq t_{ij}, \qquad \forall i \in I, j \in J. \qquad (15)
\end{aligned}$$

(Note that constraint set (14) in LIP$'$ ensures feasibility of the subproblem LISP.) Since (LISP) is a convex program with linear constraints for a fixed $\mathbf{t}$ vector, its KKT conditions are necessary and sufficient for optimality (note that since the square root function is not differentiable at zero, the KKT conditions do not apply at this single point; if, however, we consider the approximate problem with each square root term replaced by $\sqrt{\epsilon + \sum_{i\in I}\sigma_i^2 x_{ij}^2}$, for arbitrarily small $\epsilon > 0$, then the KKT conditions are necessary and sufficient for this approximate problem). Problem (LISP) is therefore amenable to dualization techniques, and its optimal dual objective function value equals the optimal primal objective function value. Define the vectors of dual variables $\mu = (\mu_1, \ldots, \mu_m) \geq 0$ and $\lambda = (\lambda_{11}, \ldots, \lambda_{mn}) \geq 0$ corresponding to the two constraint sets in (LISP). Then we can write the Lagrangian dual as

$$v(\mathbf{t}) = \max_{\mu\geq 0, \lambda \geq 0} \left[\min_{x \geq 0} \left[\phi(x) + \sum_{i\in I}\mu_i(1 - \sum_{j\in J} x_{ij}) + \sum_{i\in I}\sum_{j\in J}\lambda_{ij}(x_{ij} - t_{ij}) \right] \right], \quad (16)$$

where $\phi(x) = \sum_{i\in I}\sum_{j\in J} c_{ij}x_{ij} + \sum_{j\in J} h_j z_j^{\delta}\sqrt{\sum_{i\in I}\sigma_i^2 x_{ij}^2}$.

Problem (LIP) is therefore equivalent to the following Master Problem (MP):

$$\begin{aligned}
\text{(MP)} \qquad \text{Minimize} \quad & \sum_{j\in J} F_j y_j + \theta \\
\text{Subject to} \quad & \theta \geq \min_{x\geq 0}[\phi(x) + \sum_{i\in I}\mu_i(1 - \sum_{j\in J} x_{ij}) + \sum_{i\in I}\sum_{j\in J}\lambda_{ij}(x_{ij} - t_{ij})], \\
& \qquad\qquad \forall \mu \geq 0, \lambda \geq 0, \qquad (17) \\
& \sum_{j\in J} t_{ij} \leq N_i, \qquad \forall i \in I, \qquad (18) \\
& \sum_{j\in J} t_{ij} \geq 1, \qquad \forall i \in I, \\
& t_{ij} \leq y_j, \qquad \forall i \in I, j \in J, \\
& t_{ij}, y_j \in \{0,1\}, \qquad \forall i \in I, j \in J, \\
& \theta \geq 0.
\end{aligned}$$

Clearly we cannot write the above formulation with a constraint of the form (17) for all possible values of μ and λ. We therefore generate valid cuts successively that correspond to specific values of the vectors μ and λ and add them to the formulation in an iterative fashion (such cuts are generally referred to as Benders cuts). Given a particular binary

vector $\mathbf{t}^k$ we can solve the convex program (LISP) and recover corresponding optimal dual multiplier vectors μ^k and λ^k. We can then write

$$v(\mathbf{t}^k) = \min_{x \geq 0} \left[\sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} + \sum_{j \in J} h_j z_j^{\delta} \sqrt{\sum_{i \in I} \sigma_i^2 x_{ij}^2} + \sum_{i \in I} \mu_i^k (1 - \sum_{j \in J} x_{ij}) + \sum_{i \in I} \sum_{j \in J} \lambda_{ij}^k (x_{ij} - t_{ij}^k) \right]$$
$$= \sum_{i \in I} \mu_i^k - \sum_{i \in I} \sum_{j \in J} \lambda_{ij}^k t_{ij}^k + \min_{x \geq 0} \left[\sum_{i \in I} \sum_{j \in J} (c_{ij} + \lambda_{ij}^k - \mu_i^k) x_{ij} + \sum_{j \in J} h_j z_j^{\delta} \sqrt{\sum_{i \in I} \sigma_i^2 x_{ij}^2} \right]. \tag{19}$$

We therefore have that $\min_{x \geq 0} \left[\sum_{i \in I} \sum_{j \in J} (c_{ij} + \lambda_{ij}^k - \mu_i^k) x_{ij} + \sum_{j \in J} h_j z_j^{\delta} \sqrt{\sum_{i \in I} \sigma_i^2 x_{ij}^2} \right] = v(\mathbf{t}^k) - \sum_{i \in I} \mu_i^k + \sum_{i \in I} \sum_{j \in J} \lambda_{ij}^k t_{ij}^k$. Substituting this in (17) provides the following Benders cut for (MP) corresponding to the dual multipliers μ^k and λ^k

$$\theta \geq v(\mathbf{t}^k) - \sum_{i \in I} \sum_{j \in J} \lambda_{ij}^k (t_{ij} - t_{ij}^k). \tag{20}$$

Our Relaxed Master Problem (RMP) then becomes

(RMP) Minimize $\sum_{j \in J} F_j y_j + \theta$

Subject to

$$\theta \geq v(\mathbf{t}^k) - \sum_{i \in I} \sum_{j \in J} \lambda_{ij}^k (t_{ij} - t_{ij}^k), \qquad \forall k = 1, \ldots, K,$$
$$\sum_{j \in J} t_{ij} \leq N_i, \qquad \forall i \in I,$$
$$\sum_{j \in J} t_{ij} \geq 1, \qquad \forall i \in I,$$
$$t_{ij} \leq y_j, \qquad \forall i \in I, j \in J,$$
$$t_{ij}, y_j \in \{0, 1\}, \qquad \forall i \in I, j \in J,$$
$$\theta \geq 0,$$

where K denotes the number of Benders cuts we have generated. For a given $\mathbf{t}^k$ vector, the above Benders cut implicitly accounts for all constraints of the form of (17) (for all possible μ and λ), because λ^k and μ^k maximize $v(\mathbf{t}^k)$ over all μ and λ. Note that the (RMP) formulation is a 0-1 integer program plus a single continuous variable θ. At each iteration, we solve the (RMP) to obtain a (possibly) new $\mathbf{t}^k$ vector. Given this $\mathbf{t}^k$ vector, we then solve the subproblem (LISP) to determine the corresponding optimal dual (KKT) multiplier values. We then add the new constraint (20) to the (RMP) formulation. If the value of θ at the previous iteration does not violate this new cut at the previous $\mathbf{t}^k$, then the current solution is optimal. Otherwise we re-solve (RMP) and repeat this procedure until the same $\mathbf{t}^k$ vector is optimal in successive iterations. In the worst case, if we were to generate a constraint of the form of (20) for all possible $\mathbf{t}$ vectors, the resulting (RMP) formulation would be equivalent to (MP). In practice, however, a relatively small number of such cuts are needed to find an optimal solution. We next formalize the algorithm as follows.

Step 1: Choose an initial pair of vectors $\mathbf{y}^0$ and $\mathbf{t}^0$ that ensure a feasible solution for (LISP) and select an optimality tolerance ϵ. Solve (LISP) at $\mathbf{t} = \mathbf{t}^0$, obtaining $\mathbf{x}^0$ and corresponding optimal μ^0 and λ^0 vectors. Set $UB = \sum_{j\in J} F_j y_j^0 + v(\mathbf{t}^0)$ and let $(\bar{\mathbf{x}}, \bar{\mathbf{t}}, \bar{\mathbf{y}}) = (\mathbf{x}^0, \mathbf{t}^0, \mathbf{y}^0)$ denote the initial incumbent solution.

Step 2: Solve the (RMP) with all previously generated cuts. Let $(\theta^*, \mathbf{t}^*)$ denote an optimal solution to (RMP), and let $LB = \theta^* + \sum_{j\in J} F_j y_j^*$. If $UB - LB < \epsilon$, stop.

Step 3: Solve (LISP) at $\mathbf{t} = \mathbf{t}^*$, denoting $\mathbf{x}^*$ as the optimal solution vector and $v(\mathbf{t}^*)$ as the optimal solution value. If $\sum_{j\in J} F_j y_j^* + v(\mathbf{t}^*) < UB$, set $UB = \sum_{j\in J} F_j y_j^* + v(\mathbf{t}^*)$ and update the incumbent solution, i.e., let $(\bar{\mathbf{x}}, \bar{\mathbf{t}}, \bar{\mathbf{y}}) = (\mathbf{x}^*, \mathbf{t}^*, \mathbf{y}^*)$. If $UB - LB < \epsilon$, stop; $(\bar{\mathbf{x}}, \bar{\mathbf{y}})$ is a ϵ-optimal solution. Otherwise, recover the optimal dual multiplier vectors μ^* and λ^*, add the corresponding cut (20) to the (RMP) formulation and return to Step 2.

Remark 1 If $N_i = N$ for all $i \in I$, then the t-variables are not needed and can be removed from the master problem, thus significantly reducing the number of binary variables. In this case RMP can be simplified as:

$$
\begin{aligned}
\text{(RMPN)} \qquad \text{Minimize} \quad & \sum_{j\in J} F_j y_j + \theta \\
\text{Subject to} \quad & \theta \geq v(\mathbf{t}^k) - \sum_{i\in I}\sum_{j\in J} \lambda_{ij}^k (y_j - y_j^k), \qquad \forall k = 1, \ldots, K, && (21) \\
& \sum_{j\in J} y_j \geq 1, && (22) \\
& y_j \in \{0,1\}, \qquad \forall j \in J, \\
& \theta \geq 0,
\end{aligned}
$$

where y^k represents the y-vector generated at iteration k. The Benders cut can be written in terms of the y-variables as (21). Finally, (22) ensures feasibility of LISP, which needs to be updated so that Constraints (15) are replaced by

$$0 \leq x_{ij} \leq y_j, \qquad \forall i \in I, j \in J.$$

4.2 Hybrid Benders Decomposition/Outer Approximation Algorithm

Our preliminary computational results revealed that solving the subproblem actually serves as a bottleneck for our generalized Benders decomposition approach. We note that the subproblem should be solved to optimality at each iteration to ensure validity of the generated Benders cuts. However, in our initial computational tests, we observed that repeatedly solving the subproblem, which is a nonlinear programming problem, to optimality is computationally expensive. Observe that the subproblems solved at successive iterations of our decomposition algorithm are closely related. However, this similarity cannot be exploited without a practical warm-start capability, and the subproblem therefore must be solved from scratch at each iteration. In this section, we develop an algorithm for the subproblem to remedy these difficulties.

Recall that our subproblem has a convex objective function and linear constraints. Since the t_{ij} variables only appear in constraints (15), only the upper bounds on the

x-variables are modified between iterations, and the objective function does not depend on $\mathbf{t}$. Our main observation is that if we can reformulate the objective function as a set of linear constraints, then our subproblem can be solved as a linear program. Since linear programs can be solved to optimality efficiently and can also be re-optimized efficiently after changing variable bounds, we expect such a reformulation to yield a computationally attractive solution algorithm for our subproblem.

These observations inspired us to design an algorithm that can use information from previous iterations to solve the subproblem, and that eliminates the need to re-solve the subproblem to optimality at each iteration. This section provides an outer approximation method that can be used to solve (LISP). Outer approximation was proposed by Duran and Grossmann [7, 8] for a class of MINLP problems containing continuous variables whose feasible set is nonempty, compact, and convex, and such that functions of these continuous variables are continuous and differentiable. Our subproblem (LISP) possesses these properties, and, therefore, outer approximation can be employed for its solution.

The idea behind outer approximation (linearization) is similar to that applied in generalized Benders decomposition: at each iteration we generate upper and lower bounds on the problem's optimal solution. By using the variable values obtained using a linear approximation to the problem, we can compute these upper and lower bounds. The lower bound is the objective function value of the approximate problem, and the upper bound results from inserting the resulting x variable values into the original objective function. As this algorithm proceeds, the lower and upper bounds become closer, and they converge within ϵ in a finite number of iterations (see Floudas [11]).

Recall the subproblem for fixed assignment variables t^k (henceforth we will refer to this subproblem as the primal problem of the linearization):

$$\begin{aligned} \eta(\mathbf{t}) = \text{Minimize} \quad & \phi(x) \\ \text{Subject to} \quad & \sum_{j\in J} x_{ij} \geq 1, \qquad \forall i \in I, \\ & 0 \leq x_{ij} \leq t^k_{ij}, \qquad \forall i \in I, j \in J. \end{aligned}$$

where $\phi(x) = \sum_{i\in I}\sum_{j\in J} c_{ij}x_{ij} + \sum_{j\in J} h_j z_j^{\delta} \sqrt{\sum_{i\in I} \sigma_i^2 x_{ij}^2}$. (Recall that we overcome any infeasibility issues by adding a constraint to the master problem that requires assigning customers to at least one facility, i.e. $\sum_{j\in J} t_{ij} \geq 1$, for all $i \in I$.)

Our linearization of $\eta(t)$ will be defined in terms of an infinite set of supporting functions, which corresponds to a linearization of $\phi(x)$ at all feasible x^k points. We also note that $\phi(x)$ contains the linear term, $\sum_{i\in I}\sum_{j\in J} c_{ij}x_{ij}$, and we need not, therefore, linearize $\phi(x)$. Instead, we will only linearize the nonlinear term, $\sqrt{\sum_{i\in I} \sigma_i^2 x_{ij}^2}$, for each facility. Then, the primal problem for the linearized subproblem (LISP) can be written as

$$\begin{aligned} \text{(PPOA)} \qquad \pi(\mathbf{t}) = \text{Minimize} \quad & \sum_{i\in I}\sum_{j\in J} c_{ij}x_{ij} + \sum_{j\in J} h_j z_j^{\delta} \kappa_j(x) \\ \text{Subject to} \quad & \sum_{j\in J} x_{ij} \geq 1, \qquad \forall i \in I, \\ & 0 \leq x_{ij} \leq t^k_{ij}, \qquad \forall i \in I, j \in J, \end{aligned} \tag{23}$$

where $\kappa_j(x) = \sqrt{\sum_{i\in I} \sigma_i^2 x_{ij}^2}$. Because of the convexity and continuous differentiability of $\kappa_j(x)$ (except at 0), the following condition is satisfied for all feasible x^k and all facilities $j \in J$:

$$\kappa_j(x) \geq \kappa_j(x^k) + \nabla\kappa_j(x^k)(x - x^k) \tag{24}$$

Then, we can write the master problem for our linearization approach as

$$\begin{aligned}
\text{(MPOA)} \quad \chi(t) = \text{Minimize} \quad & \sum_{i\in I}\sum_{j\in J} c_{ij}x_{ij} + \sum_{j\in J} h_j z_j^{\delta} \xi_j^{OA} \\
\text{Subject to} \quad & \xi_j^{OA} \geq \kappa_j(x^k) + \nabla\kappa_j(x^k)(x - x^k), \quad \forall j \in J, k \in F \\
& \sum_{j\in J} x_{ij} \geq 1, \qquad \forall i \in I, \\
& 0 \leq x_{ij} \leq t_{ij}^k, \qquad \forall i \in I, j \in J \\
& \kappa_j(x^k) + \nabla\kappa_j(x^k)(x - x^k) \geq 0, \quad \forall j \in J,
\end{aligned}$$

where $F = \{k : x^k \text{ is a feasible solution to the primal problem (PPOA)}\}$. Then, the formal algorithm for the linearization approach can be given as follows:

Algorithm OA

Step 0: Input: Feasible $\mathbf{t}$, an optimality tolerance ϵ and a counter k.

Step 1: Solve the master problem of the outer approximation (MPOA) for $\mathbf{t}$ with all previously added constraints, and obtain an optimal solution x^k and corresponding optimal dual λ vector. Set the lower bound $LB_{OA} = \chi(t)$.

Step 2: Solve (PPOA) to calculate the value of $\pi(t^k)$ by using the optimal x^k values obtained in the previous step. Set $UB_{OA} = \pi(t^k)$. If $UB_{OA} - LB_{OA} \leq \epsilon$, go to Step 3. Otherwise, calculate $\nabla\kappa_j(x^k)$ for each $j \in J$, obtain a new constraint (24), add it to (MPOA), set $k = k + 1$ and return to Step 2.

Step 3: Output: Optimal x^k and λ^k vectors, and $\chi(t^k)$.

Note that at each iteration, by adding supporting vectors to (MPOA), we obtain a better approximation of the subproblem. We can use these valid cuts at each successive iteration of the Benders algorithm because at each iteration we add a new constraint that provides a better approximation of the objective function and is independent of the vector $\mathbf{t}$.

The new Benders cut that we obtain from the linearization approach is

$$\theta \geq \chi(t^k) - \sum_{i\in I}\sum_{j\in J} \lambda_{ij}^k (t_{ij} - t_{ij}^k) \tag{25}$$

where λ^k denotes the vector of dual variables associated with constraints (23) at iteration k. This Benders cut is valid for any ϵ that is selected to solve the linearization because $\chi(t)$ gives us a lower bound on the optimal subproblem solution at any iteration of the linearization algorithm. Since we do not need to solve the subproblem to optimality to obtain valid cuts, and because we can use the previously generated supporting vectors for the subproblem at any iteration, we have designed a hybrid algorithm that uses Benders decomposition with an embedded linearization algorithm.

Hybrid Algorithm. In this hybrid algorithm, we use the linearization algorithm to solve the subproblems of the Benders decomposition algorithm. We start with feasible $\mathbf{y}$ and $\mathbf{t}$ variables and solve the subproblem with a higher value of optimality tolerance, ϵ'. The reason for not solving the subproblem to optimality, i.e., with a smaller value of an optimality tolerance, is because we do not want to spend a lot of time solving the subproblem for values of $\mathbf{y}$ and $\mathbf{t}$ that may not serve as a good choice with respect to the Benders master problem. We first solve the subproblem approximately, i.e., with a higher optimality tolerance, obtain a new Benders cut that we add to the master problem, and then obtain new $\mathbf{y}$ and $\mathbf{t}$-variables. We continue solving the subproblem approximately until we obtain $\mathbf{y}$ and $\mathbf{t}$ variables that have previously been investigated. When we encounter previously investigated variables, we solve the subproblem to optimality, i.e., with a smaller optimality tolerance, ϵ'', where $\epsilon'' << \epsilon'$. We then obtain a new Benders cut and continue the procedure until the Benders decomposition algorithm terminates within an optimality tolerance of ϵ. This approach saves an important amount of CPU time, and therefore permitted solving much larger problem instances.

The formal algorithm can be given as follows:

Step 1: Choose an initial pair of vectors $\mathbf{y}^0$ and $\mathbf{t}^0$ that ensure a feasible solution for (LISP). Add $(\mathbf{y}^0, \mathbf{t}^0)$ to the set S. Select an optimality tolerance ϵ for the Benders algorithm, and set the counter $l = 0$ and $UB_B = \infty$.

Step 2: Solve the subproblem using **Algorithm OA** with input $\mathbf{t}^l$ and ϵ' to obtain a χ^l value and x^l and λ^l vectors.

Step 3: Calculate the current upper bound $\bar{UB}_B = \sum_{j \in J} F_j y_j^l + \pi(t^k)$. If $\bar{UB}_B < UB_B$, set $UB_B = \bar{UB}_B$ and update the incumbent solution, i.e., let $(\bar{\mathbf{x}}, \bar{\mathbf{t}}, \bar{\mathbf{y}}) = (\mathbf{x}^*, \mathbf{t}^*, \mathbf{y}^*)$.

Step 4: Add a Benders cut (25) to (RMP). Solve (RMP) with all previously generated cuts. Let $(\theta^*, \mathbf{y}^*, \mathbf{t}^*)$ denote an optimal solution to (RMP), and let $LB_B = \theta^* + \sum_{j \in J} F_j y_j^*$.

Step 5: If $UB_B - LB_B < \epsilon$, stop. Otherwise, let $(\mathbf{y}^l, \mathbf{t}^l) = (\mathbf{y}^*, \mathbf{t}^*)$. If $(\mathbf{y}^*, \mathbf{t}^*) \in S$, go to Step 2 and solve the subproblem to within an optimality tolerance of ϵ''. Otherwise, add $(\mathbf{y}^l, \mathbf{t}^l)$ to the set S, set $l = l + 1$, and go to Step 2 using an optimality tolerance of ϵ'.

Remark 2 Our hybrid algorithm is similar to Zakeri et al. [38]'s method for accelerating Benders decomposition for problems where the subproblem is a large-scale linear programming problem. In particular, Zakeri et al. [38] propose solving the linear subproblem using a primal-dual interior point method, terminating the solution process before optimality is reached and generating "inexact Benders cuts." Recall that our subproblem is a convex nonlinear programming problem, whose repeated solution to optimality requires a significant amount of CPU time. We solve the subproblem approximately using an outer approximation approach and generate inexact Benders cuts before optimality is reached. Furthermore, similar to Zakeri et al. [38], we modify the optimality tolerance dynamically to ensure that the subproblem is initially solved to a coarse approximation and the approximation is refined when necessary. Therefore, our approach can be viewed as an acceleration method for generalized Benders decomposition similar to Zakeri et al. [38]'s acceleration method for the classical Benders decomposition.

5 Computational Results

In this section we first compare the efficacy of our hybrid algorithm with the direct solution of the problem (LIP) using a commercial solver and the implementation of the classical generalized Benders decomposition algorithm on small and medium problem instances. We then present computational results on our hybrid algorithm for large problem instances. Finally, we provide a computational characterization of the incremental costs that result from a single-sourcing strategy.

5.1 Comparison of Our Hybrid Algorithm with GAMS/BARON and Generalized Benders Algorithm

We first attempted to solve the problem (LIP) using GAMS/BARON, a commercial mixed-integer nonlinear solver. We use six different data sets that have different parameter settings, as shown in Table 1. Our test instances contain five supply facilities, each of which has a fixed cost that is uniformly distributed between 400 and 500, i.e., $F_j \sim U[400, 500]$, and five customers, each of which has demand that has an average that is uniformly distributed between 4000 and 6000, i.e. $\mu_i \sim U[4000, 6000]$. We take two different coefficient of variation values for demand as shown in the third column of Table 1. We take holding cost of items at each facility as 1, and vary the unit flow cost as shown in the second column of Table 1.

Table 1 Data parameter settings for algorithm comparison.

Data Set #	c_{ij}	**CoV** (σ/μ)
1	$U[0.05, 0.55]$	0.3
2	$U[0.05, 0.95]$	0.3
3	$U[0.05, 1.35]$	0.3
4	$U[0.05, 0.55]$	0.4
5	$U[0.05, 0.95]$	0.4
6	$U[0.05, 1.35]$	0.4

For each data set given in Table 1, we generated 10 random instances, resulting in 60 test instances in total. In all these instances, we assume that $N_i = 5$, which means practically there is no limit on the number of supply facilites to which a customer can be assigned. We limit the CPU time to 1200 seconds for each problem instance while using GAMS/BARON to solve these instances. We implemented the problem on GAMS 23.6 and performed all tests on a Windows XP PC with a 3.4 GHz CPU and 2 GB RAM.

GAMS/BARON was able to solve only 24 instances out of 60 to optimality within the given time limit. We provide the number of instances that are solved to optimality in the column labeled "# of Solved" in Table 2. For those instances that are solved to optimality we calculate the average CPU times that GAMS/BARON spent and provide them in the column labeled "Avg CPU" under GAMS/BARON column. As it can seen, the average CPU time varies between 273.74 sec. and 1304.33 sec. There are 36 instances that GAMS/BARON failed to solve within 1200 seconds. We report the average gap for these instances that were reported at the end of the time limit and present them in the column labeled "Avg Gap." These averages vary between 13.11% and 31.51%.

Table 2 Comparison of the three solution approaches.

	GAMS/ BARON		Generalized Benders	Hybrid Algorithm	
Data Set	**# of Solved**	**Avg CPU (sec)**	**Avg Gap (%)**	**Avg CPU (sec)**	**Avg CPU (sec)**	**Avg Density (%)**
S1	3	447.43	13.11	1.98	0.098	94.2
S2	4	646.81	13.75	1.91	0.092	91.5
S3	6	594.27	14.38	1.88	0.080	91.9
S4	1	1304.33	23.01	2.09	0.090	93.5
S5	5	273.74	31.51	1.74	0.092	93.5
S6	5	324.11	23.71	1.89	0.073	93.6

In order to solve problems of reasonable size we developed code for the generalized Benders algorithm and our hybrid algorithm described in the previous section for application to problem (LIP). We implemented the generalized Benders algorithm on the same version of GAMS, where we used CPLEX 11.2 for solving the 0-1 integer programming master problem (RMP) and CONOPT for solving the convex subproblem (LISP). The Generalized Benders algorithm was able to solve all instances within a few seconds. The average CPU times are provided in column "Avg CPU" under "Generalized Benders." Finally, we implemented our hybrid algorithm in C++ on the same computer. We used CPLEX 11.2 for solving both (RMP) and linearization of the subproblems (MPOA). Our hybrid algorithm solved all instances to optimality within a fraction of a second. The average CPU times are given in the column labeled "Avg CPU" under "Hybrid Algorithm" column. We also calculated the density of the Benders cuts used in the instances given above and provide them in the last column of Table 2. The density of the cuts are calculated by dividing the number of binary variables that have positive coefficients in each cut to the number of all binary variables that we have in that instance. As it can seen from the table, the cuts that are produced by our hybrid algorithm are high density cuts whose density vary between 91.5% and 94.2%.

Our next experiment is aimed at comparing the performances of the generalized Benders decomposition and our hybrid algorithm on larger problem instances. We use the same data generation procedure for generating problem instances having 10 supply facilities and 5 customers. We set the limit on the number of supply facilities to which a customer can be assigned to 5. The results are given in Table 3.

Table 3 Comparison of generalized Benders Algorithm with Hybrid Algorithm.

	Generalized Benders			Hybrid Algorithm	
Data Set	**# of Solved**	**Avg CPU (sec)**	**Avg Gap (%)**	**Avg CPU (sec)**	**Avg Density (%)**
M1	3	66.40	34.28	1.35	94.50
M2	8	346.88	8.35	1.38	92.47
M3	10	192.46	-	1.88	91.32
M4	3	34.02	65.80	1.44	96.18
M5	2	886.84	35.52	1.92	92.14
M6	7	203.11	6.49	1.98	92.05

Generalized Benders algorithm solved 43 instances out of 60 to optimality within the given time limit of 1200 seconds. We provide the number of these instances in the column labeled "# of Solved". For those instances we calculate the average CPU times, provide them in the column "Avg CPU" under "Generalized Benders" column. The average CPU time varies between 34.02 and 886.84 seconds. For the instances that are interrupted because of the time limit, we calculate the average gap at the end of the time limit and provide them in column "Avg Gap." We observe that the average gap can be as high as 65.8% for some data sets. On the other hand, our hybrid algorithm solved all instances within a few seconds. We provide the average CPU times in the column labeled "Avg CPU" under "Hybrid Algorithm" column. Moreover, we provide the average density of the cuts for each instance in column labeled "Avg Density." As seen in this column, the Benders cuts that we generate are high density cuts.

The results shown on Table 3 show that our hybrid algorithm significantly outperforms the generalized Benders decomposition algorithm on data sets with 10 customers and 5 supply facilities. However, the size of these instances is relatively small compared to potential practical problems. Therefore, we also test the performance of our algorithm for larger problem instances. We used the values of the parameters given in Table 1 corresponding to Data Set 6, which is the data set that takes the longest to solve using our algorithm as shown on Table 3. Using these parameter settings, we generated data sets for different numbers of customers and supply facilities. These data sets are given in Table 4. For each data set given in Table 4, we generated 10 random instances,

Table 4 Numbers of supply facilities and customers for large problem instances

Data set	M	N	Data set	M	N
L1	20	5	L6	40	15
L2	20	10	L7	50	10
L3	30	5	L8	50	15
L4	30	10	L9	60	10
L5	40	10	L10	60	15

resulting in 100 test instances in total. We limit the CPU time to 1200 seconds for each data set-cardinality pair. As explained in the algorithm section, we first solve the model with no cardinality constraint and obtain the maximum number of facilities to which a customer is assigned. Then, we start solving the model with a cardinality constraint that is equal to the actual cardinality at the previous step minus one.

Table 5 summarizes the results. We provide the cardinality in column labeled "N_i" and the number of instances solved to optimality within the time limit in column labeled "Solved". The optimality gap is calculated as the ratio of the difference between the best upper and lower bound to the best lower bound at the time limit. We calculate the gap for every test instance that is not solved to optimality within the time limit. Then we take the averages of these gaps and provide these figures in the column labeled "Avg Gap."

As shown in Table 5, most of the test instances were solved optimally when no cardinality constraint exists on the number of facilities to which a customer can be assigned, although exceptions exist for data sets L6, L8 and L10. As the cardinality constraint becomes tighter, the amount of time that the algorithm spends finding the optimal solution for the master problem increases. However, because at each iteration we add cutting planes to the subproblem and better approximate the objective function,

Table 5 Computational results of hybrid algorithm for large problem instances

Data Set	M	N	N_i	Solved	Avg Gap	Data Set	M	N	N_i	Solved	Avg Gap
L1	20	5	4	10	-	L7	50	10	6	10	-
	20	5	3	10	-		50	10	5	10	-
	20	5	2	10	-		50	10	4	10	-
L2	20	10	5	10	-		50	10	3	9	0.03%
	20	10	4	10	-		50	10	2	-	0.56%
	20	10	3	10	-	L8	50	15	6	5	2.79%
	20	10	2	9	1.11%		50	15	5	5	1.85%
L3	30	5	4	10	-		50	15	4	5	1.80%
	30	5	3	10	-		50	15	3	3	1.10%
	30	5	2	6	0.34%		50	15	2	-	-
L4	30	10	5	10	-	L9	60	10	6	10	-
	30	10	4	10	-		60	10	5	10	-
	30	10	3	10	-		60	10	4	9	0.04%
	30	10	2	3	0.47%		60	10	3	6	0.10%
L5	40	10	5	10	-		60	10	2	-	0.62%
	40	10	4	10	-	L10	60	15	6	3	2.26%
	40	10	3	7	0.12%		60	15	5	5	1.67%
	40	10	2	-	0.78%		60	15	4	4	1.04%
L6	40	15	6	9	2.41%		60	15	3	3	0.38%
	40	15	5	10	-		60	15	2	-	1.15%
	40	15	4	5	2.00%						
	40	15	3	6	1.26%						
	40	15	2	5	0.81%						

in some cases the gap may decrease as the cardinality constraint becomes tighter. For example, for data set L6, where $M = 40$ and $N = 15$, and data set L8, where $M = 50$ and $N = 15$, the gap decreases as N_i decreases.

In other data sets, the solution time of the master problem increases as N_i decreases and the gap becomes larger. Among these test instances, the highest gap is 2.79% for data set L8 with no cardinality constraint. As Table 5 shows, our algorithm is able to provide optimal or near optimal solutions for problems of practical size within a reasonable amount of computing time.

5.2 Analysis of Single-Sourcing Strategy

We would like to characterize the percentage difference in the costs of problem instances when single-sourcing is enforced relative to the case in which demand splitting is allowed. With this goal in mind, we conducted a broad set of computational tests using a range of parameter settings, and then compared the results that we obtained for both problems. All of the test problems discussed in the rest of this section used $M = 10$ customers and $N = 5$ supply facilities.

The limit on the number of supply facilities that can serve each customer, i.e., N_i for customer $i \in I$, is an important parameter for our model. Since the maximum number of supply facilities for all instances was 5, we parametrically varied N_i between 1 and 5 for each problem instance (and used the same value of N_i for each customer). Obviously, when we set N_i to 1 for each customer $i \in I$, we obtain an optimal solution for the problem with single-sourcing requirements. Let Z_k denote the optimal objective function value when $N_i = k$. Our main goal is to analyze the effect of different parameters on the percentage difference between the minimum cost when demand

splitting is allowed and when single-sourcing is imposed. We therefore calculated the percentage difference, ΔZ_k, as $\Delta Z_k = (Z_1 - Z_k)/Z_k$ for $k = 1, \ldots, 5$ and for each set of parameter values. Note that ΔZ_5 characterizes the percentage cost difference between the single-sourcing case and the case in which demand splitting is unrestricted.

The relative values of average assignment cost and holding cost play important roles in our model. We would therefore like to analyze the impacts of these parameters simultaneously. Since these parameters tend to have opposing effects on the relative cost difference ΔZ_5, we analyze the effect of the expected value of the ratio of the (per unit) assignment cost to the holding cost, i.e. $E[\hat{c}/h]$. We used 5 different parameter settings for $E[\hat{c}]$, i.e., 0.3, 0.4, 0.5, 0.6, and 0.7 and we set the holding cost equal to 1 for all facilities (therefore $E[\hat{c}/h] = E[\hat{c}]$). The individual c_{ij} values were randomly generated from a uniform distribution that ensures the prescribed value of $E[\hat{c}/h]$. Table 6 provides the uniform distribution parameters for each setting of $E[\hat{c}/h]$. Our choice of values of $E[\hat{c}]$ was based on the fact that in practice, the holding cost is often a percentage of the total value of an item. That is, suppose $h = ic'$, where i is a percentage holding cost rate (often between 15% and 25%) and c' is the item's value. Next, suppose $\hat{c} = \hat{\imath}c'$, i.e., where $\hat{\imath}$ reflects the percentage of total value that constitutes transportation cost. Then, for example, if $\hat{\imath} = 10\%$, and $i = 20\%$, we have $\hat{c} = 0.5$.

The ratio of the standard deviation to the mean demand is another important parameter that affects the cost performance of single-sourcing relative to demand splitting. Since the standard deviation affects the magnitude of safety stock holding cost and the mean affects the magnitude of assignment costs, instead of analyzing the effects of these two parameters separately, we analyzed their ratio, i.e., the coefficient of variation (CoV$= \sigma/\mu$) of demand. We randomly generated mean demands between 4000 and 6000 and used 3 different values for CoV, 0.35, 0.40, and 0.45, to determine the associated standard deviation values.

The other important parameter affecting cost performance is the fixed cost of a supply facility. A high fixed cost decreases the number of open supply facilities, which in turn affects the assignment of customers to supply facilities. We randomly generated four different data sets for F_j values from the uniform distributions shown in Table 6. While these values of fixed costs may appear relatively small, these values might reflect the portion of fixed cost that is allocated to the single product in question. Clearly, as our results later show, higher fixed costs lead to a choice of fewer facilities. In such cases, the difference in cost between an optimal single-sourcing strategy and an optimal demand-splitting strategy will naturally decrease.

Table 6 Data parameter settings.

E$[\hat{c}/h]$	c_{ij}	μ_i	**CoV** (σ/μ)	**Fixed Cost** (F_j)
0.3	$U[0.05, 0.55]$	$U[4000, 6000]$	0.30	$U[100, 200]$
0.4	$U[0.05, 0.75]$		0.35	$U[200, 300]$
0.5	$U[0.05, 0.95]$		0.40	$U[300, 400]$
0.6	$U[0.05, 1.15]$		0.45	$U[400, 500]$
0.7	$U[0.05, 1.35]$		0.50	

By using the cross combinations of these three parameter settings, i.e., $E[\hat{c}/h]$, CoV, and F_j, we generated 100 ($5 \times 5 \times 4$) different data sets. For each data set we generated 10 random test instances, resulting 1000 test instances in total. We set the service level to 97.5% ($z^{\delta} = 1.96$) for all test instances.

First, we analyzed the effect of $E[\hat{c}/h]$. Table 7 summarizes the results for different values of $E[\hat{c}/h]$. We provide the maximum and minimum values of ΔZ_5 from among the 6000 instances in the columns labeled **max** and **min**, respectively, with the average value in the column labeled **average**.

Table 7 The maximum, minimum, and average value of ΔZ_5 for different values of $E[\hat{c}/h]$.

	ΔZ_5		
$E(\hat{c}/h)$	**max**	**min**	**average**
0.3	6.57%	0.00%	1.40%
0.4	**7.43%**	0.00%	2.66%
0.5	6.80%	0.00%	2.81%
0.6	5.64%	0.00%	2.65%
0.7	5.80%	0.00%	2.41%

The highest percentage difference obtained among 6000 instances equals 7.43%. The minimum percentage difference is 0%, which means that in some of the cases a single-sourcing solution is optimal even though single sourcing is not enforced. As seen in Figure 2, both low and high levels of $E[\hat{c}/h]$ lead to the optimality of single-sourcing solutions. At higher levels of $E[\hat{c}/h]$, the problem becomes similar to an uncapaci-

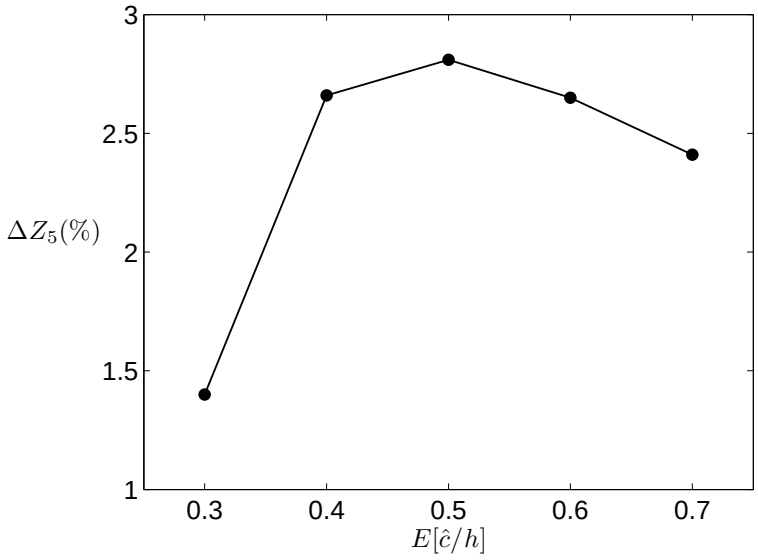

Fig. 2 The effect of $E[\hat{c}/h]$ on ΔZ_5.

tated facility location problem, where single sourcing is optimal. Also, at lower levels of $E[\hat{c}/h]$, the facility and safety stock costs dominate the objective function. In the presence of fixed facility location costs, the model reduces the number of facilities and uses aggregation to obtain risk pooling benefits. However, at intermediate values of the ratio of the transportation cost to the holding cost, the model seeks to reduce

transportation costs by utilizing more locations, and simultaneously benefits from risk pooling by mixing the demands of multiple customers at the open locations. This illustrates the fact that even in the presence of fixed facility costs, the benefits of deviating from a single-sourcing policy are non-negligible. However, when the facility and/or transportation costs dominate, the best single-sourcing solution value approaches the optimal solution value.

We illustrate the average value of ΔZ_k for different values of k (where $k = N_i$ for each $i \in I$) in Figure 3. As can be seen from Figure 3, when there is no limit on

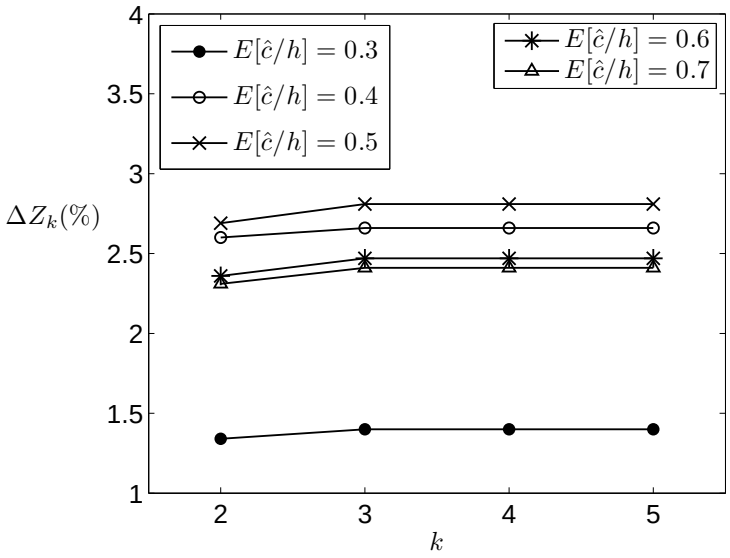

Fig. 3 The effect of N_i on ΔZ_k for different values of $E(\hat{c}/h)$.

the number of facilities that can supply any customer, i.e., when $N_i = 5$, an optimal solution assigns customers to at most 3 different supply facilities. In the majority of cases, assigning each customer to at most 2 supply facilities is optimal. The gap between the performance of the single sourcing and multiple sourcing solutions is significant. However, the difference when we increase N_i from 2 to 3 is not significant.

Next, we analyze the effect of CoV. Table 8 summarizes the results. As we can see

Table 8 The maximum, minimum, and average value of ΔZ_5 values for different values of CoV.

	ΔZ_5		
CoV	**max**	**min**	**average**
0.3	5.60%	0.00%	2.30%
0.35	7.43%	0.00%	2.46%
0.4	5.97%	0.00%	2.44%
0.45	6.80%	0.00%	2.30%
0.50	5.98%	0.00%	2.25%

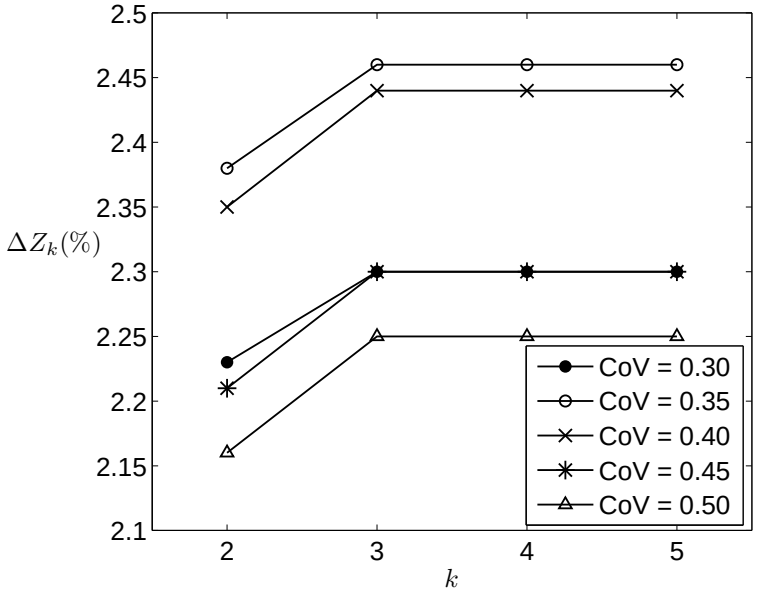

Fig. 4 The effect of N_i on ΔZ_k for different values of CoV.

in both Table 8 and Figure 4, as the coefficient of variation increases from 0.3 to 0.5, the percentage cost difference between optimal single sourcing and demand splitting solutions first increases and then decreases. The main reason for this is that as the CoV increases, the standard deviation of demand increases. In turn, this leads to higher safety stock holding costs. The model tends to open fewer supply facilities and benefits from risk pooling by assigning more customers to fewer supply facilities. Similarly, the percentage cost difference decreases as the CoV approaches the origin because, in this case, the safety stock holding cost becomes so small that the problem becomes similar to an uncapacitated facility location problem. We next analyze the effect of the fixed

Table 9 The maximum, minimum, and average value of ΔZ_5 for different values of fixed cost.

	ΔZ_5		
fixed cost	**max**	**min**	**average**
U(100,200)	6.80%	0.00%	2.57%
U(200,300)	7.43%	0.00%	2.42%
U(300,400)	5.94%	0.00%	2.22%
U(400,500)	5.52%	0.00%	2.20%

facility opening cost. This effect is shown in Table 9. As we would expect, as the fixed cost increases, fewer locations are opened, and customers are therefore assigned to fewer locations. Thus, the benefits of demand splitting decrease as the fixed facility costs increase.

Finally, we consider the CPU times for different values of N_i. In each of these tests, we first solve the model without a cardinality constraint, i.e., $N_i = N$, and

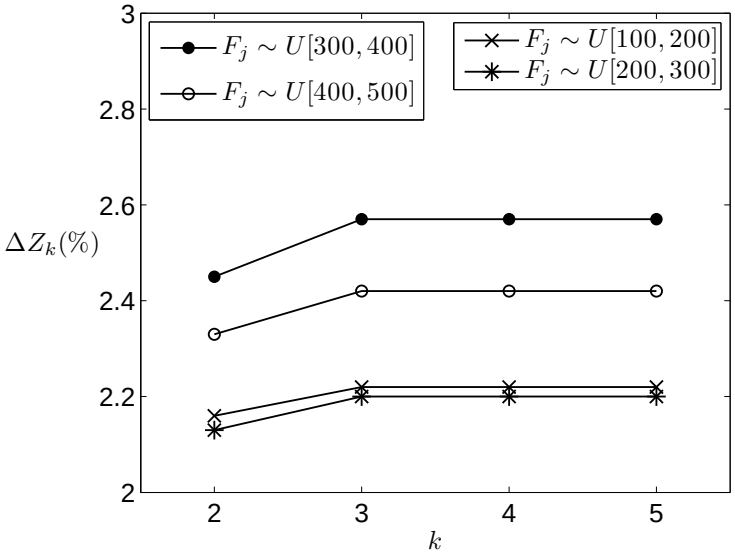

Fig. 5 The effect of N_i on ΔZ_k for different values of fixed cost.

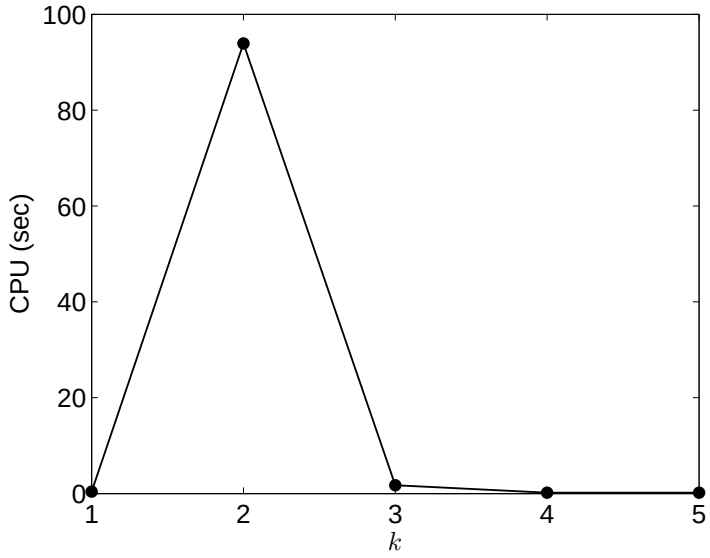

Fig. 6 CPU times for different values of N_i.

determine the actual cardinality at optimality (we call this the unconstrained optimal cardinality level). We do not then re-solve the problem for the cardinality constraint levels that are between the unconstrained optimal cardinality level and N. That is, if the unconstrained optimal cardinality level equals $\bar{N}$ when $N_i = N$, then we only re-solve the problem for cardinality constraint value of $k < \bar{N}$. Therefore, the CPU times for these unsolved instances are taken as equal to the CPU time when $N_i = N$. Figure 6 illustrates these results. As Figure 6 shows, the greatest CPU time is needed when $N_i = 2$. In most of the instances when there is no cardinality limit for a customer, the optimal solution assigns a customer to at most 3 or 4 supply facilities. When we limit the number of supply facilities to $N_i = 2$, the corresponding constraint becomes tight and the required CPU time increases. This increase in CPU time comes as a result of the increased time CPLEX must spend solving the 0-1 integer master problem (RMP). However, when $N_i = 5$, the constraint is loose in almost all instances, and the required CPU time is significantly lower (when this constraint is loose, CPLEX is able to solve the (RMP) much more quickly). In most of these test instances, the model finds optimal solutions in less than a second.

6 Conclusion and Future Research Directions

In this paper, we discussed a supply chain setting where customers with stochastic demand are assigned to uncapacitated supply facilities. Our model determines the locations of facilities and the assignment of customers to supply facilities in order to minimize the total supply facility opening cost, customer-supply facility assignment cost and the safety stock costs at supply facilities. In the literature, similar problems have been investigated with a single-sourcing requirement for each customer. We relax this constraint and apply an upper bound on the number of facilities to which a customer can be assigned. Our goal was to characterize the difference between the costs of problems where demand-splitting is allowed and those that enforce single-sourcing.

The resulting location-inventory problem falls into a class of difficult mixed-integer nonlinear programming problems. The structure of the objective function, however, leads us to characterize interesting solution properties for some special cases. For the general problem, we proposed a generalized Benders decomposition algorithm and a hybrid algorithm that allows us to solve significantly larger problem sizes to optimality. We implemented our algorithm and conducted a broad set of computational tests to analyze the effects of key parameters on the percentage difference in costs when demand-splitting is allowed and when single sourcing is required. According to our computational study, with the parameter settings we tested, this percentage difference can be as high as 7%.

The relative values of assignment holding costs, the coefficient of variation of customer demands, and the fixed opening costs of facilities are the most important parameters that affect the optimal assignment of customers. Therefore, we analyzed the effects of these parameters by using a range of settings. According to our computational study, both low and high levels of the ratio of assignment cost to holding cost lead to solutions where single sourcing is optimal (or near optimal). However, at intermediate values, the model benefits from risk pooling by mixing the demands of multiple customers at the open locations. Similarly, low and high levels of coefficient of variation lead to solutions where single sourcing is optimal because either the assignment cost or the safety stock cost is dominant. At intermediate values, where there is a balance in

the costs, the model benefits from multiple-sourcing. Furthermore, high values of fixed cost naturally lead to opening fewer facilities, which in turn leads to the assignment of customers to fewer locations. Therefore, as the fixed cost increases, the benefits of demand splitting decrease.

This research can be extended in a number of different ways. One possible extension would consider the addition of finite capacities to supply facilities. Another extension might be adding a penalty cost for assigning a customer to more than one facility, instead of using a restriction on the number of facilities to which a customer can be assigned. However, this affects the form of the objective function, leading to an objective function that is neither convex nor concave. An additional interesting extension considers service-level-dependent assignment costs, which reflect cases in which some facilities may require a higher service level and increase in associated assignment cost. In this setting, customers may accept reduced service levels instead of paying higher costs. Thus, instead of defining pre-specified service levels at the supply facilities, we may treat facility service levels as decision variables.

Appendix

Note: This appendix may be included as an on-line appendix if required.

Proof of Lemma 1: Let $f_{ij}(x_{ij}) = \sigma_i x_{ij}$ and $F(x) = \sqrt{\sum_{i\in I}[f_{ij}(x_{ij})]^2}$. Now we need to show that $F(x)$ is convex. Let $\mathbf{F}(x) = [f_{11}(x_{11}), \ldots, f_{ij}(x_{ij})]$. Then $F(x)$ is the l_2 norm of $\mathbf{F}(x)$, i.e.,$F(x) = \|\mathbf{F}(x)\|$.

$$\begin{aligned} F(\lambda x_1 + (1-\lambda)x_2) &= \|\mathbf{F}(\lambda x_1 + (1-\lambda)x_2)\| \\ &= \|\lambda \mathbf{F}(x_1) + (1-\lambda)\mathbf{F}(x_2)\| \quad \text{(because } \mathbf{F}(x) \text{ is linear in } x_{ij}) \\ &\le \|\lambda \mathbf{F}(x_1)\| + \|(1-\lambda)\mathbf{F}(x_2)\| \quad \text{(triangular inequality)} \\ &= \lambda F(x_1) + (1-\lambda)F(x_2). \end{aligned}$$

Since h_j and z_j^δ are nonnegative constants, $h_j z_j^\delta \sqrt{\sum_{i\in I}[f_{ij}(x_{ij})]^2}$ is also convex. Moreover, since the first term of $\phi(x)$ is linear and the second term is the summation of convex functions, $\phi(x)$ is convex in x. □

Proof of Theorem 1: When $|I(j)| = 0$, no customers are assigned to supply facility j. Without loss of generality, we assume that this supply facility is not open and we exclude this supply facility from consideration in our problem. Therefore we consider the KKT conditions for $j \in J$ such that $|I(j)| > 0$. Clearly each x_{ij} is between 0 and 1. Since $\frac{1}{\omega_j} > 0$, using condition (8) we set $\beta_{ij} = 0$ for all $i \in I(j)$. From condition (6), we require

$$\begin{aligned} c_i + hz\sigma \frac{1/\omega_j}{\sqrt{|I(j)|/\omega_j^2}} - \mu_i = 0, \quad \forall j \in J, i \in I(j), \\ \Rightarrow \mu_i = \frac{hz\sigma}{\sqrt{|I(j)|}} + c_i, \quad \forall j \in J, i \in I(j). \end{aligned}$$

Thus we have $\mu_i \ge 0$ for all $j \in J$ and $i \in I(j)$. For each $i \notin I(j)$ we set $x_{ij} = 0$ and $\beta_{ij} = c_i$, which ensures that (6) holds for all $i \in I$ and $j \in J$. We have therefore constructed a solution satisfying (6), (8), and (10). By assumption we have $\sum_{j\in J(i)} \frac{1}{\omega_j} = 1$ for all $i \in I$, which implies that (7) and (9) hold, and all KKT conditions are satisfied by the solution we have constructed. □

Proof of Proposition 1: From Lemma (1) we know that this two-supplier, two-customer problem is a convex programming problem. Therefore the generalized KKT Conditions are necessary and sufficient for optimality. The KKT conditions for this

problem can be written as follows:

$$c_{ij} + H\frac{x_{ij}}{\sqrt{\sum_{i\in I} x_{ij}^2}} - \mu_i - \beta_{ij} = 0 \qquad \text{for } i = 1,2 \text{ and } j = 1,2 \qquad \text{(A-1)}$$

$$\mu_i(1 - \sum_{j\in J} x_{ij}) = 0 \qquad \text{for } i = 1,2 \qquad \text{(A-2)}$$

$$\beta_{ij} x_{ij} = 0 \qquad \text{for } i = 1,2 \text{ and } j = 1,2 \qquad \text{(A-3)}$$

$$1 - \sum_{j\in J} x_{ij} \leq 0 \qquad \text{for } i = 1,2 \qquad \text{(A-4)}$$

$$x_{ij} \geq 0 \qquad \text{for } i = 1,2 \text{ and } j = 1,2 \qquad \text{(A-5)}$$

$$\mu_i \geq 0 \qquad \text{for } i = 1,2 \qquad \text{(A-6)}$$

$$\beta_{ij} \geq 0 \qquad \text{for } i = 1,2 \text{ and } j = 1,2 \qquad \text{(A-7)}$$

For the given solution, $x_{11} = x_{22} = \alpha$ and $x_{12} = x_{21} = 1 - \alpha$ where $0 < \alpha < 1$, from condition (A-3) we set $\beta_{ij} = 0$ for $i = 1,2$ and $j = 1,2$. Since $x_{11} + x_{12} = x_{21} + x_{22} = 1$, condition (A-2) is already satisfied. From condition (A-1), we require $\mu_1 = c_{12} + H\frac{(1-\alpha)}{\sqrt{\alpha^2+(1-\alpha)^2}}$ and $\mu_2 = c_{21} + H\frac{(1-\alpha)}{\sqrt{\alpha^2+(1-\alpha)^2}}$. Thus we have $\mu_1 \geq 0$ and $\mu_2 \geq 0$. Hence, the given solution satisfies all KKT conditions from (A-1) to (A-7) and is therefore optimal. The value of the objective function, Z^{opt}, equals $c_{12} + c_{21} + 2H\frac{(1-\alpha)}{\sqrt{\alpha^2+(1-\alpha)^2}}$ which also equals $c_{11} + c_{22} + 2H\frac{\alpha}{\sqrt{\alpha^2+(1-\alpha)^2}}$.

Proof of Corollary 1: Define Z^{opt} be the objective function value for the given solution, $x_{11} = x_{22} = \alpha$ and $x_{12} = x_{21} = 1 - \alpha$, let Z^1 be the objective function value when $x_{11} = x_{22} = 1$; $x_{12} = x_{21} = 0$, let Z^2 be the objective function value when $x_{12} = x_{21} = 1$; $x_{11} = x_{22} = 0$, and finally let Z^3 be the objective function value when $x_{11} = x_{21} = 1$; $x_{12} = x_{22} = 0$ (from symmetry Z^3 also gives the objective value when $x_{12} = x_{22} = 1$; $x_{11} = x_{21} = 0$). Then these objective function values can be calculated as follows:

$$
\begin{aligned}
Z^{opt} &= c_{12} + c_{21} + 2H\frac{(1-\alpha)}{\sqrt{\alpha^2+(1-\alpha)^2}} \\
Z^1 &= c_{11} + c_{22} + 2H \\
&= c_{12} + c_{21} + 2H\frac{(1-\alpha)}{\sqrt{\alpha^2+(1-\alpha)^2}} + 2H\left(1 - \frac{\alpha}{\sqrt{\alpha^2+(1-\alpha)^2}}\right) \\
&= Z^{opt} + 2H\left(1 - \frac{\alpha}{\sqrt{\alpha^2+(1-\alpha)^2}}\right) \\
Z^2 &= c_{12} + c_{21} + 2H \\
&= c_{12} + c_{21} + 2H\frac{(1-\alpha)}{\sqrt{\alpha^2+(1-\alpha)^2}} + 2H\left(1 - \frac{1-\alpha}{\sqrt{\alpha^2+(1-\alpha)^2}}\right) \\
&= z^{opt} + 2H\left(1 - \frac{1-\alpha}{\sqrt{\alpha^2+(1-\alpha)^2}}\right) \\
Z^3 &= c_{11} + c_{21} + \sqrt{2}H (\text{or } c_{12} + c_{22} + \sqrt{2}H) \\
&= c_{12} + c_{21} + 2H\frac{(1-\alpha)}{\sqrt{\alpha^2+(1-\alpha)^2}} + H\left(\sqrt{2} - \frac{1}{\sqrt{\alpha^2+(1-\alpha)^2}}\right) \\
&= z^{opt} + H\left(\sqrt{2} - \frac{1}{\sqrt{\alpha^2+(1-\alpha)^2}}\right)
\end{aligned}
$$

Let $\Delta Z^1 = Z^1 - Z^{opt}$, $\Delta Z^2 = Z^2 - Z^{opt}$ and $\Delta Z^3 = Z^3 - Z^{opt}$. Clearly, the objective function value of the minimum-cost single-sourcing solution minus that of the minimum-cost solution with customer demand splitting, $\Delta\overline{Z}_{min}$, equals the minimum of ΔZ^1, ΔZ^2 and ΔZ^3.

$$
\begin{aligned}
\Delta\overline{Z}_{min} &= \min\left\{\Delta Z^1; \Delta Z^2; \Delta Z^3\right\} \\
&= \min\left\{2H\left(1 - \frac{\alpha}{\sqrt{\alpha^2+(1-\alpha)^2}}\right); 2H\left(1 - \frac{1-\alpha}{\sqrt{\alpha^2+(1-\alpha)^2}}\right); H\left(\sqrt{2} - \frac{1}{\sqrt{\alpha^2+(1-\alpha)^2}}\right)\right\} \\
&= H\left[\min\left\{2\left(1 - \frac{\max\{\alpha, 1-\alpha\}}{\sqrt{\alpha^2+(1-\alpha)^2}}\right); \sqrt{2} - \frac{1}{\sqrt{\alpha^2+(1-\alpha)^2}}\right\}\right] \\
&= H \times \rho(\alpha)
\end{aligned}
$$

where $\rho(\alpha) = \left[\min\left\{2\left(1 - \frac{\max\{\alpha,1-\alpha\}}{\sqrt{\alpha^2+(1-\alpha)^2}}\right); \sqrt{2} - \frac{1}{\sqrt{\alpha^2+(1-\alpha)^2}}\right\}\right]$.

References

1. O. Bilde and J. Krarup. Sharp lower bounds for the simple location problem. *Annals of Discrete Mathematics*, 1:79–97, 1977.
2. S. Chopra and P. Meindl. *Supply Chain Management: Strategy, Planning , and Operations*. Prentice-Hall, New Jersey, 4th edition, 2010.

3. G. Cote and M. Laughton. Large-scale mixed integer programming: Benders type heuristics. *Eurpoean Journal of Operational Research*, 16:327–333, 1984.
4. M.S. Daskin and S.H. Owen. *Location Models in Transportation in Handbook of Transportation Science*. Kluwer Academic Publishers, Boston, 1999.
5. 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*, 110:83–106, 2002.
6. M.S. Daskin, L.V. Snyder, and R.T. Berger. Facility location in supply chain design. Technical report, Department of Industrial Engineering and Management Science, Northwestern University, 2003.
7. M.A. Duran and I.E. Grossmann. An outer approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical Programming*, 36:307–339, 1986.
8. M.A. Duran and I.E. Grossmann. A mixed-integer nonlinear programming algorithm for process systems synthesis. *AIChE Journal*, 32(4):592–606, 1986.
9. S.J. Erlebacher and R.D. Meller. The interaction of location and inventory in designing distribution systems. *IIE Transactions*, 32:155–166, 2000.
10. D. Erlenkotter. A dual-based procedure for uncapacitated facility location. *Operations Research*, 26(6):992–1009, 1978.
11. C.A. Floudas. *Nonlinear and Mixed-Integer Optimization*. Oxford University Press, New York, 1995.
12. P.M. França and H.P.L. Luna. Solving stochastic transportation-location problems by generalized benders decomposition. *Transportation Science*, 16(2):113–126, 1982.
13. A.M. Geoffrion. Generalized benders decomposition. *Journal of Optimization Theory and Applications*, 10(4):237–260, 1972.
14. A.M. Geoffrion. Lagrangean relaxation for integer programming. *Mathematical Programming Study*, 2:82–114, 1974.
15. M. Haouari, M. Mrad, and H.D. Sherali. Optimum synthesis of discrete capacitated networks with multi-terminal commodity flow requirements. *Optimization Letters*, 1:341–354, 2007.
16. H.H. Hoc. Topological optimization of networks: A nonlinear mixed integer model employing generalized benders decomposition. *IEEE Transactions on Automatic Control*, 27(1):164–169, 1982.
17. T. L. Magnanti and R. T. Wong. Accelerating benders decomposition: Algorithmic enhancement and model selection criteria. *Operations Research*, 29(3):464–484, 1981.
18. P. Mahey, A. Benchakroun, and F. Boyer. Capacity and flow assignment of data networks by generalized benders decomposition. *Journal of Global Optimization*, 20:173–193, 2001.
19. T. Melo, S. Nickel, and F. Saldanha da Gama. Facility location and supply chain management - a comprehensive review. In *Berichte des Fraunhofer ITWM, Nr. 130*. 2007.
20. P.A. Miranda and R.A. Garrido. A simultaneous inventory control and facility location model with stochastic capacity constraint. *Netw Spat Econ*, 6:39–53, 2006.
21. E. Munõz and M. Stolpe. Generalized benders' decomposition for topology optimization problems. *forthcoming in Journal of Global Optimization*, 2010.
22. L.K. Nozick and M.A. Turnquist. 'integrating inventory impacts into a fixed-charge model for locating distribution centers. *Transportation Research Part E*, 34(3):173–186, 1998.

23. L.K. Nozick and M.A. Turnquist. 'a two-echelon inventory allocation and distribution center location analysis. *Transportation Research Part E*, 37:425–441, 2001.
24. L.K. Nozick and M.A. Turnquist. Inventory, transportation, service quality and the location of distribution centers. *European Journal of Operational Research*, 129:362–371, 2001.
25. S.H. Owen and M.S. Daskin. Strategic facility location: A review. *European Journal of Operational Research*, 111(3):423–447, 1998.
26. L. Ozsen, C.R. Coullard, and M.S. Daskin. Capacitated warehouse location model with risk pooling. *Naval Research Logistics*, 55:295–312, 2008.
27. L. Ozsen, C.R. Coullard, and M.S. Daskin. Facility location modeling and inventory management with multi-sourcing. *Transportation Science*, 43(4):455–472, 2009.
28. G.K.D. Saharidis and M.G. Ierapetritou. Improving benders decomposition using maximum feasible subset (mfs) cut generation strategy. *Computers and Chemical Engineering*, 34:1237–1245, 2010.
29. G.K.D. Saharidis, M. Minoux, and M.G. Ierapetritou. Accelerating benders method using covering cut bundle generation. *International Transactions in Operational Research*, 17:221–237, 2010.
30. Z-J. M. Shen. A multi-commodity supply chain design problem. *IIE Transactions*, 37:753–762, 2005.
31. Z-J. M. Shen and M.S. Daskin. Trade-offs between customer service and cost in integrated supply chain design. *Manufacturing & Service Operations Management*, 7(3):188–207, 2005.
32. Z-J. M. Shen, D. Coullard, and M.S. Daskin. A joint location-inventory model. *Transportation Science*, 37(1):40–55, 2003.
33. J. Shu, C-P. Teo, and Z-J. M. Shen. Stochastic transportation-inventory network design problem. *Operations Research*, 53(1):48–60, 2005.
34. L.V. Snyder. Facility location under uncertainty: A review. *IIE Transactions*, 38(7):547–564, 2006.
35. L.V. Snyder, M.S. Daskin, and C-P. Teo. The stochastic location model with risk pooling. *European Journal of Operational Research*, 179:1221–1238, 2007.
36. K. Sourirajan, L. Ozsen, and R. Uzsoy. A single-product network design model with lead time and safety stock considerations. *IIE Transactions*, 39:411–424, 2007.
37. N. Vidyarthi, E. Celebi, S. Elhedhli, and E. Jewkes. Integrated production-inventory-distribution system design with risk pooling: Model formulation and heuristic solution. *Transportation Science*, 41(3):392–408, 2007.
38. G. Zakeri, A.B. Philpott, and D.M. Ryan. Inexact cuts in benders decomposition. *SIAM Journal on Optimization*, 10:643–657, 2000.
39. P. Zipkin. *Foundations of Inventory Management*. McGraw-Hill, Boston, 2000.