A stochastic programming approach for supply chain network design under uncertainty

Tjendera Santoso, Shabbir Ahmed*, Marc Goetschalckx, Alexander Shapiro

School of Industrial & Systems Engineering,
Georgia Institute of Technology, 765 Ferst Drive, Atlanta, GA 30332.

June 16, 2003

Abstract

This paper proposes a stochastic programming model and solution algorithm for solving supply chain network design problems of a realistic scale. Existing approaches for these problems are either restricted to deterministic environments or can only address a modest number of scenarios for the uncertain problem parameters. Our solution methodology integrates a recently proposed sampling strategy, the Sample Average Approximation scheme, with an accelerated Benders decomposition algorithm to quickly compute high quality solutions to large-scale stochastic supply chain design problems with a huge (potentially infinite) number of scenarios. A computational study involving two real supply chain networks are presented to highlight the significance of the stochastic model as well as the efficiency of the proposed solution strategy.

Keywords: Facilities planning and design; Supply chain network design; Stochastic programming; Decomposition methods; Sampling.

1 Introduction

A crucial component of the planning activities of a manufacturing firm is the efficient design and operation of its supply chain. A supply chain is a network of suppliers, manufacturing plants, warehouses, and distribution channels organized to acquire raw materials, convert these raw materials to finished products, and distribute these products to customers. Strategic level supply chain planning involves deciding the configuration of the network, i.e., the number, location, capacity, and technology of the facilities. The tactical level planning of supply chain operations involves deciding the aggregate quantities and material flows for purchasing, processing, and distribution of products. The strategic configuration of the supply chain is a key factor influencing efficient tactical operations, and therefore has a long lasting impact on the firm. Furthermore, the fact that the supply chain configuration involves the commitment of substantial capital resources over long periods of time makes the supply chain network design problem an extremely important one.

Beginning with the seminal work of Geoffrion and Graves [8] on multi-commodity distribution system design, a large number of optimization-based approaches have been proposed for the design

*Corresponding author. E-mail: sahmed@isye.gatech.edu, Phone: 404-894-2320, Fax: 404-894-2301.

of supply chain networks (see for example [1, 9, 31]). However, the majority of this research assumes that the operational characteristics of, and hence the design parameters for, the supply chain are deterministic. Unfortunately, critical parameters such as customer demands, prices, and resource capacity are quite uncertain. Moreover, the arrival of regional economic alliances, for instance the Asian Pacific Economic Alliance and the European Union, have prompted many corporations to move more and more towards global supply chains, and therefore to become exposed to risky factors such as exchange rates, reliability of transportation channels, and transfer prices [32]. Unless the supply chain is designed to be robust with respect to the uncertain operating conditions, the impact of operational inefficiencies such as delays and disruptions will be larger than necessary. A recent study [12] found that after a company announces a supply chain disruption, such as a production or shipment delay, its stock price can decrease significantly, with an average decrease of 8.6% on the day of the announcement, and is often followed by further decreases, as much as 20% over the next six months.

The significance of uncertainty has prompted a number of researchers to address stochastic parameters in tactical level supply chain planning involving distribution of raw materials and products (see for example [6, 17, 33]). At the strategic level, there is a great deal of research in the facility location component of supply chain network design under uncertainty (see for example [24]). However, research addressing comprehensive design of supply chain networks under uncertainty is significantly smaller in number. Gutierrez et al. [10] proposed a robust optimization framework for network design under uncertainty. This approach seeks network configurations that are good (nearly optimal) for a variety of scenarios of the design parameters at the expense of being suboptimal for any one scenario. The authors proposed a modification of the Benders decomposition algorithm [3], commonly used for deterministic network design problems [8], to generate robust designs. Computational testing of the algorithm using randomly generated test problems with networks comprised of up to 33 nodes, 100 arcs, 64 commodities, and 11 scenarios was carried out. Mirhassani et al. [22] considered a two-stage model for multi-period capacity planning of supply chain networks. Here the first stage decisions, comprised of openings and closings of the plants and distribution centers and setting their capacity levels, are to be decided prior to the realization of future demands. Then, based upon the particular demand scenario realized, the production and distribution decisions are to be decided optimally. The overall objective is to minimize the cost of the first-stage strategic decisions and the expected production and distribution costs over the uncertain demand scenarios. The authors used Benders decomposition to solve the resulting stochastic integer program, and presented computational results on supply chain networks involving up to 8 plant sites, 15 distribution centers, 30 customer locations, and with 100 scenarios. Tsiakis et al. [28] also considered a two-stage stochastic programming model for supply chain network design under demand uncertainty. The authors developed a large-scale mixed-integer linear programming model for this problem, and presented a case study using a European supply chain network involving 14 products, 18 customer locations, 6 distribution center locations, and 3 demand scenarios. Alonso-Ayuso et al. [2] proposed a branch-and-fix heuristic for solving two-stage stochastic supply chain design problems. Computational results on networks involving 6 plants, 12 products, 24 markets, and 23 scenarios were presented.

As evident from the above discussion, the existing stochastic programming approaches for supply chain design under uncertainty are suited for a very small number of scenarios. Consider a distribution network with just 50 facilities, each with an uncertain operating level. Assume that the operating level for a facility can be one of only three possibilities and are independent across

facilities. Then there are a total of $3^{50} \approx 7 \times 10^{23}$ scenarios for the joint realization of the uncertainties! This is far more than any of the existing stochastic programming approaches for supply chain design can handle. In this paper, we integrate a recently proposed sampling strategy, the Sample Average Approximation scheme, with an accelerated Benders decomposition algorithm to solve supply chain design problems with continuous distributions for the uncertain parameters, and hence an infinite number of scenarios. The proposed algorithmic technique is immediately applicable to problems with finite but an extremely large number of scenarios. Our computational results involving two real supply chain networks indicate that the proposed methodology can serve as a viable strategic planning tool, allowing planners to investigate a multitude of possibilities, and investigate a variety of solutions.

The remainder of this paper is organized as follows. In the next section we describe a two-stage model for supply chain design under uncertainty and identify some of the challenges involved in developing a solution strategy. In Section 3, we develop the proposed solution methodology. We explain the Sample Average Approximation technique, its integration with Benders decomposition, and develop techniques for accelerating Benders decomposition. In Section 4, we present a computational study involving two real supply chain networks to illustrate the proposed methodology. Finally, we offer some concluding remarks in Section 5.

2 Problem Description

Let us first describe a deterministic mathematical formulation for the supply chain design problem considered in this paper. Consider a supply chain network $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ where $\mathcal{N}$ is the set of nodes and $\mathcal{A}$ is the set of arcs. The set $\mathcal{N}$ consists of the set of suppliers $\mathcal{S}$, the set of processing facilities $\mathcal{P}$ and the set of customers $\mathcal{C}$, i.e., $\mathcal{N} = \mathcal{S} \cup \mathcal{P} \cup \mathcal{C}$. The processing facilities include manufacturing centers $\mathcal{M}$, finishing facilities $\mathcal{F}$ and warehouses $\mathcal{W}$, i.e., $\mathcal{P} = \mathcal{M} \cup \mathcal{F} \cup W$. Further, a manufacturing center $i \in \mathcal{M}$ or a finishing facility $i \in \mathcal{F}$ consists of a set of manufacturing or finishing machines $\mathcal{N}_i$. Thus the set $\mathcal{P}$ includes the processing centers as well as the machines in these centers. Let $\mathcal{K}$ be the set of products flowing through the supply chain.

The supply chain configuration decisions consist of deciding which of the processing centers to build (major configuration decisions) and which processing and finishing machines to procure (minor configuration decisions). We associate a binary variable y_i to these decisions: $y_i = 1$ if a processing facility i is built or machine i is procured, and 0 otherwise. The operational decisions consist of routing the flow of product $k \in \mathcal{K}$ from the supplier to the customers. We let x_{ij}^k denote the flow of product k from a node i to a node j of the network where $(ij) \in \mathcal{A}$. We are now ready to state a deterministic mathematical model for our supply chain design problem.

$$\min \quad \sum_{i\in\mathcal{P}} c_i y_i + \sum_{k\in\mathcal{K}} \sum_{(ij)\in\mathcal{A}} q_{ij}^k x_{ij}^k \tag{2.1}$$

$$\text{s.t.} \quad y \in Y \subseteq \{0,1\}^{|\mathcal{P}|} \tag{2.2}$$

$$\sum_{i\in\mathcal{N}} x_{ij}^k - \sum_{l\in\mathcal{N}} x_{jl}^k = 0 \;\; \forall\, j \in \mathcal{P}, \;\; \forall\, k \in \mathcal{K} \tag{2.3}$$

$$\sum_{i\in\mathcal{N}} x_{ij}^k \geq d_j^k \;\; \forall\, j \in \mathcal{C}, \;\; \forall\, k \in \mathcal{K} \tag{2.4}$$

$$\sum_{j\in\mathcal{N}} x_{ij}^k \leq s_i^k \;\; \forall\, i \in \mathcal{S}, \;\; \forall\, k \in \mathcal{K} \tag{2.5}$$

$$\sum_{k\in\mathcal{K}} r_j^k \left(\sum_{i\in\mathcal{N}} x_{ij}^k \right) \leq m_j y_j \;\; \forall\, j \in \mathcal{P} \tag{2.6}$$

$$x \in \mathbb{R}_+^{|\mathcal{A}|\times|\mathcal{K}|} \tag{2.7}$$

In the above mode c_i denotes the investment cost for building facility i or procuring machine i, and q_{ij}^k denotes the per-unit cost of processing product k at facility i and/or transporting product k on arc (ij). All cost components are assumed to be on an annualized basis. The objective function (2.1) consists of minimizing total investment and operational costs. Constraint (2.2) includes the set Y of logical dependencies and restrictions, such as $y_i \leq y_j$ for all $i \in \mathcal{N}_j$ and $j \in \mathcal{P}$ or $\mathcal{F}$, i.e., machine $i \in \mathcal{N}_j$ should only be procured if facility j is built. This constraint also enforces the binary nature of the configuration decisions for the processing facilities. Constraint (2.3) enforces the flow conservation of product k across each processing node j. Constraint (2.4) requires that the total flow of product k to a customer node j, should exceed the demand d_j^k at that node. Constraint (2.5) requires that the total flow of product k from a supplier node j, should be less than the supply s_j^k at that node. Constraint (2.6) enforces capacity constraints of the processing nodes. Here r_j^k denotes per-unit processing requirement for product k at node j. The capacity constraint then requires that the total processing requirement of all products flowing into a processing node j should be smaller than the capacity m_j of facility j if it is built ($y_j = 1$). If facility j is not built ($y_j = 0$) the constraint will force all flow variables $x_{ij}^k = 0$ for all $i \in \mathcal{N}$. Finally, constraint (2.7) enforces the non-negativity of the flow variables corresponding to an arc $(ij) \in \mathcal{A}$ and product $k \in \mathcal{K}$.

It will be convenient to work with the following compact notation for model (2.1)-(2.7).

$$\min \quad c^T y + q^T x \tag{2.8}$$

$$\text{s.t.} \quad y \in Y \subseteq \{0,1\}^{|\mathcal{P}|} \tag{2.9}$$

$$Nx = 0 \tag{2.10}$$

$$Dx \geq d \tag{2.11}$$

$$Sx \leq s \tag{2.12}$$

$$Rx \leq My \tag{2.13}$$

$$x \in \mathbb{R}_+^{|\mathcal{A}|\times|\mathcal{K}|} \tag{2.14}$$

Above the vectors c, q, d, and s correspond to investment costs, processing/transportation costs, demands, and supplies respectively. The matrices N, D and S are appropriate matrices correspond-

ing to the summations on the left-hand-side of the expressions (2.3),(2.4) and (2.5), respectively. The notation R corresponds to a matrix of r_j^k, and the notation M corresponds to a matrix with m_j along the diagonal.

To extend the above model to a stochastic setting, we assume that processing/transportation costs, demands, supplies, and capacities are stochastic parameters with known joint distribution. We use bold face to denote random variables (random vectors) in order to distinguish them from their particular realizations. In particular, $\boldsymbol{\xi} = (\boldsymbol{q}, \boldsymbol{d}, \boldsymbol{s}, \boldsymbol{M})$ represents the random data vector while $\xi = (q, d, s, M)$ stands for its particular realization. The design objective is to minimize the sum of current investment costs and *expected* future processing and transportation costs. Given that it may be impossible to meet demand for certain realizations of the uncertain parameters, we include an additional cost term to penalize shortfall. The resulting formulation is as follows

$$\min_{y} \quad \left\{ f(y) := c^T y + \mathbb{E}[Q(y, \boldsymbol{\xi})] \right\} \tag{2.15}$$

$$\text{s.t.} \quad y \in Y \subseteq \{0,1\}^{|\mathcal{P}|}, \tag{2.16}$$

where $Q(y, \xi)$ is the optimal value of the following problem

$$\min_{x,z} \quad q^T x + h^T z \tag{2.17}$$

$$\text{s.t.} \quad Nx = 0 \tag{2.18}$$

$$Dx + z \geq d \tag{2.19}$$

$$Sx \leq s \tag{2.20}$$

$$Rx \leq My \tag{2.21}$$

$$x \in \mathbb{R}_+^{|\mathcal{A}|\times|\mathcal{K}|} \tag{2.22}$$

Note again that $\boldsymbol{\xi}$ in (2.15) is a random vector corresponding to the uncertain processing/transportation costs, demands, supplies, and capacities, and the optimal value $Q(y, \xi)$ of the second stage problem (2.17)–(2.22) is a function of the first stage decision variable y and a realization (or a scenario) $\xi = (q, d, s, M)$ of the uncertain parameters. The expectation in (2.15) is taken with respect to the probability distribution of $\boldsymbol{\xi}$ which is supposed to be known. The variable z in constraint (2.19) and the cost component $h^T z$ in (2.17) corresponds to the penalty incurred for failing to meet demand.

Model (2.15)-(2.22) is a two-stage stochastic program. The first-stage consists of the deciding the configuration decisions y, and the second-stage consists of processing and transporting products from suppliers to customers in an optimal fashion based upon the configuration and the realized uncertain scenario. The objective is to minimize current investment costs $c^T y$ and *expected* future operating costs $\mathbb{E}[Q(y, \boldsymbol{\xi})]$. The demand-shortage penalty $h^T z$ guarantees that $Q(y, \xi) < \infty$ for all y and ξ. Furthermore, we assume that possible realizations of the processing and transportation costs $\boldsymbol{q}$, and the penalty costs h are sufficiently high such that $Q(y, \xi) > -\infty$ for all y and ξ, and hence $Q(y, \xi)$ is finite valued for all $y \in Y$ and possible realizations of the random data. We assume further that the expected value $\mathbb{E}[Q(y, \boldsymbol{\xi})]$ is well defined and finite valued for the considered distribution of $\boldsymbol{\xi}$. Consequently problem (2.15)-(2.16) has a well-defined objective function $f(y)$ and, since the set Y is nonempty and finite, possesses an optimal solution.

There are two potential sources of difficulty in solving problem (2.15)-(2.16). First, an evaluation of the objective function $f(y)$ for a given configuration y, involves computing the expected value of

the linear programming value function $Q(y, \xi)$. For continuous distributions, exact computation of this expectation involves taking multiple integrals and is quite impossible. For discrete distributions, computing the expectation might involve solving a prohibitively huge number of linear programs (2.17)-(2.22), one for each scenario of the uncertain problem parameter realizations. Second, even if the expectation in (2.15) can be computed exactly, optimization of this function presents significant difficulties. It is well-known that $\mathbb{E}[Q(y, \boldsymbol{\xi})]$ is a convex nonlinear (often non-smooth) function of y (see, e.g., [4]). However this function is not available in a closed analytical form and is only implicitly defined. Thus problem (2.15)-(2.16) involves minimizing an implicitly defined non-linear function with respect to binary variables, and can be quite difficult.

3 Algorithmic Strategy

In this section, we detail an algorithmic strategy for solving the stochastic supply chain network design problem (2.15)-(2.16). Our method integrates a sampling strategy with an accelerated Benders decomposition scheme.

3.1 Sample Average Approximation

As mentioned earlier, a key difficulty in solving the stochastic program (2.15)-(2.16) is in evaluating the expectation in the objective. We deal with this problem using the *Sample Average Approximation* (SAA) scheme (cf., [16, 21, 27]).

In the SAA scheme, a random sample $\xi^1, \ldots, \xi^N$ of N realizations (scenarios) of the random vector $\boldsymbol{\xi}$ is generated, and the expectation $\mathbb{E}[Q(y, \boldsymbol{\xi})]$ is approximated by the sample average function $N^{-1} \sum_{n=1}^{N} Q(y, \xi^n)$. Consequently, the "true" problem (2.15)-(2.16) is approximated by the problem

$$\min_{y \in Y} \quad \left\{ \widehat{f}_N(y) := c^T y + \tfrac{1}{N} \textstyle\sum_{n=1}^{N} Q(y, \xi^n) \right\}. \tag{3.1}$$

Let v_N and $\widehat{y}_N$ be the optimal value and an optimal solution vector, respectively, of the SAA problem (3.1). Note that v_N and $\widehat{y}_N$ are random in the sense that they are functions of the corresponding random sample. However, for a particular realization $\xi^1, \ldots, \xi^N$ of the random sample, the problem (3.1) is deterministic and can be solved by appropriate optimization techniques.

It is possible to show that under mild regularity conditions, as the sample size N increases, v_N and $\widehat{y}_N$ converge with probability one to their true counterparts, and moreover $\widehat{y}_N$ converges to an optimal solution of the true problem with probability approaching one exponentially fast [16]. This convergence analysis suggests that a fairly good approximate solution to the true problem (2.15)-(2.16) can be obtained by solving an SAA problem (3.1) with a modest sample size. In particular, suppose that the SAA problem is solved to an absolute optimality tolerance of $\delta \geq 0$ and let $\varepsilon > \delta$ and $\alpha \in (0, 1)$. Then a sample size of

$$N \geq \frac{3\sigma_{\max}^2}{(\varepsilon - \delta)^2} \log\left(\frac{|Y|}{\alpha}\right) \tag{3.2}$$

guarantees that the SAA solution $\widehat{y}_N$ is an ε-optimal solution (i.e., a solution with an absolute optimality gap of ε) to the true problem with a probability of at least $1 - \alpha$. Here $\sigma_{\max}^2$ is a maximal variance of certain function differences (see [16] for details of the estimate (3.2)). Note that

$|Y| \leq 2^{|\mathcal{P}|}$, and hence $\log|Y| \leq (\log 2)|\mathcal{P}|$. Even though the above estimate (3.2) is too conservative for practical applications, it suggests that the required sample size is at most linear in the number of processing facilities $|\mathcal{P}|$. In practice, the SAA scheme involves repeated solutions of the SAA problem (3.1) with independent samples. Statistical confidence intervals are then derived on the quality of the approximate solutions. This procedure is described next.

The SAA algorithm:

Step 1. Generate M independent samples each of size N, i.e., $(\xi_j^1, \ldots, \xi_j^N)$ for $j = 1, \ldots, M$. For each sample solve the corresponding SAA problem

$$\min_{y \in Y} c^T y + \frac{1}{N} \sum_{n=1}^{N} Q(y, \xi_j^n).$$

Let v_N^j and $\widehat{y}_N^j$, $j = 1, \ldots, M$, be the corresponding optimal objective value and an optimal solution, respectively.

Step 2. Compute

$$\bar{v}_{N,M} := \frac{1}{M} \sum_{j=1}^{M} v_N^j \quad \text{and} \quad \sigma^2_{\bar{v}_{N,M}} := \frac{1}{(M-1)M} \sum_{j=1}^{M} (v_N^j - \bar{v}_{N,M})^2. \tag{3.3}$$

It is well known that the expected value of v_N is less than or equal to the optimal value v^* of the true problem (see, e.g., [23, 21]). Since $\bar{v}_{N,M}$ is an unbiased estimator of $\mathbb{E}[v_N]$, we obtain that $\mathbb{E}[\bar{v}_{N,M}] \leq v^*$. Thus $\bar{v}_{N,M}$ provides a lower statistical bound for the optimal value v^* of the true problem, and $\sigma^2_{\bar{v}_{N,M}}$ is an estimate of the variance of this estimator.

Step 3. Choose a feasible solution $\bar{y} \in Y$ of the true problem, for example, use one of the computed solutions $\widehat{y}_N^j$. Estimate the true objective function value $f(\bar{y})$ as follows:

$$\tilde{f}_{N'}(\bar{y}) := c^T \bar{y} + \frac{1}{N'} \sum_{n=1}^{N'} Q(\bar{y}, \xi^n). \tag{3.4}$$

Here $\xi^1, \ldots, \xi^{N'}$ is a sample of size N' generated *independently* of the sample used to obtain $\bar{y}$. Typically one can take N' much bigger than the sample size N used in solving the SAA problems. This step involves solution of N' independent second-stage subproblems (2.17)-(2.22). Note that $\tilde{f}_{N'}(\bar{y})$ is an unbiased estimator of $f(\bar{y})$. Since $\bar{y}$ is a feasible solution to the true problem, we have $f(\bar{y}) \geq v^*$. Thus $\tilde{f}_{N'}(\bar{y})$ is an estimate of an upper bound on v^*. If the sample $\xi^1, \ldots, \xi^{N'}$ is iid (*independent* identically distributed), then the variance of this estimate can be estimated as

$$\sigma^2_{N'}(\bar{y}) := \frac{1}{(N'-1)N'} \sum_{n=1}^{N'} \left(c^T \bar{y} + Q(\bar{y}, \xi^n) - \tilde{f}_{N'}(\bar{y}) \right)^2. \tag{3.5}$$

Step 4. Compute an estimate of the optimality gap of the solution $\bar{y}$ using the lower bound estimate and the objective function value estimate from Steps 2 and 3 respectively, as follows:

$$\mathrm{gap}_{N,M,N'}(\bar{y}) := \tilde{f}_{N'}(\bar{y}) - \bar{v}_{N,M}. \tag{3.6}$$

The estimated variance of the above gap estimator is then given by

$$\sigma^2_{\text{gap}} = \sigma^2_{N'}(\bar{y}) + \sigma^2_{\bar{v}_{N,M}}. \tag{3.7}$$

It is natural to choose $\bar{y}$ as one of the calculated estimates $\widehat{y}^j_N$ with the smallest estimated objective value $\tilde{f}_{N'}(\bar{y})$. Let us emphasize again that in order to estimate the corresponding true objective function value $f(\bar{y})$ one needs to generate a sample independent of the samples used in calculation of $\widehat{y}^j_N$. The above SAA procedure involving statistical evaluation of candidate solutions was suggested in Norkin et al. [23] and further developed in Mak et al. [21]. Theoretical analysis of the SAA algorithm for solving stochastic programs with discrete first-stage variables was carried out in Kleywegt et al. [16]. Computational studies using the SAA method for solving stochastic linear programs is presented in Linderoth et al. [18], and for solving stochastic routing problems in Verweij et al. [30].

3.2 Accelerated Benders Decomposition

Step 1 of the SAA method outlined in the previous section calls for repeated solutions of the sample average approximating problem (3.1). This problem is itself a two-stage stochastic program, albeit with a much fewer number of scenarios than the true problem (2.15)-(2.16). Recall that the function $Q(y, \xi^n)$ in (3.1) is the value function of a linear program (2.17)-(2.22) and is piece-wise linear and convex in y. Consequently the SAA problem (3.1) involves minimizing the sum of a fairly large number, N, of piece-wise linear convex functions (plus the first stage linear cost function $c^T y$) over $0-1$ variables. Moreover since the functions $Q(y, \xi^n)$ are not available in closed form, the problem is quite difficult. Cutting plane algorithms such as Benders decomposition [3] (also known as the L-shaped decomposition method in the stochastic programming literature [29]) is quite suitable for this class of problems. We briefly state the general Benders decomposition algorithm as it applies to problem (3.1).

Benders decomposition algorithm:

Initialization step: Set lower and upper bounds $lb = -\infty$ and $ub = +\infty$ respectively. Set the iteration counter $i = 0$. Let $\widehat{y}$ denote the incumbent solution.

Step 1: Solve the master problem

$$\begin{aligned} lb = \quad & \min_{y,\theta} \quad c^T y + \theta \\ & \text{s.t.} \quad\; y \in Y \\ & \qquad\quad \theta \geq a_k^T y + b_k \quad k = 1, \ldots, i. \end{aligned}$$

Let y^i be an optimal solution of the master problem.

Step 2: For $n = 1, \ldots, N$, solve the subproblems (2.17)-(2.22) corresponding to y^i and $\xi^n =$

(q^n, d^n, s^n, M^n). We re-state this problem for convenience:

$$\begin{aligned} Q(y^i, \xi^n) = \quad & \min \quad (q^n)^T x + h^T z \\ & \text{s.t.} \quad Nx = 0 \qquad (\mu) \\ & \qquad Dx + z \geq d^n \qquad (\nu) \\ & \qquad Sx \leq s^n \qquad (\pi) \\ & \qquad Rx \leq (M^n) y^i \qquad (\rho) \\ & \qquad x \in \mathbb{R}_+^{|\mathcal{A}|\times|\mathcal{K}|}, \end{aligned}$$

where the Greek terms in parenthesis next to the constraints denote to the corresponding dual variables. Using the subproblem objective values, compute the objective function value

$$\widehat{f}_N(y^i) = c^T y^i + \frac{1}{N} \sum_{n=1}^{N} Q(y^i, \xi^n)$$

corresponding to the current feasible solution y^i. If $ub < \widehat{f}_N(y^i)$, update the upper bound $ub = \widehat{f}_N(y^i)$ and the incumbent solution $\widehat{y} = y^i$.

Step 3: If $ub - lb < \delta$, where $\delta \geq 0$ is pre-specified tolerance, stop and return $\widehat{y}$ as the optimal solution and ub as the optimal objective value; otherwise proceed to Step 4.

Step 4: For each $n = 1, \ldots, N$, let $(\mu_i^n, \nu_i^n, \pi_i^n, \rho_i^n)$ be the optimal dual solutions for the subproblem corresponding to y^i and ξ^n solved in Step 2. Compute the cut coefficients

$$a_{i+1}^T = \frac{1}{N} \sum_{n=1}^{N} (\rho_i^n)^T M^n, \quad \text{and} \quad b_{i+1} = \frac{1}{N} \sum_{n=1}^{N} \left((\nu_i^n)^T d^n + (\pi_i^n)^T s^n \right).$$

Update $i = i + 1$ and go to Step 1.

In Benders decomposition, the master problem solved in Step 1 involves a lower approximation of the nonlinear function $N^{-1} \sum_{n=1}^{N} Q(y, \xi^n)$ through the θ variables and the cuts $\theta \geq a_k^T y + b_k$. This problem is a mixed-integer linear program involving $|\mathcal{P}|$ binary variables and one continuous variable. The optimal objective value and the optimal solution obtained from the master problem correspond to a lower bound to the optimal objective value and a feasible solution for the SAA problem (3.1), respectively. The objective value corresponding to this feasible solution is evaluated by solving the subproblems in Step 2. These N linear programs can be solved independent of each other, allowing for a computationally convenient decomposition. Since the current solution is feasible, this yields an upper bound information. If the lower and upper bounds are sufficiently close together we conclude optimality. Otherwise, dual information from the subproblems is used to compute an optimality cut which is added to the master problem. The new optimality cut serves to improve the approximation of $N^{-1} \sum_{n=1}^{N} Q(y, \xi^n)$ in the master problem, which is then resolved and the process iterates.

By duality it follows that the optimality cut added at the end of the i-th iteration is "exact" at y^i, i.e.,

$$a_{i+1}^T y^i + b_{i+1} = N^{-1} \sum_{n=1}^{N} Q(y^i, \xi^n),$$

and is a lower approximate at all other solutions. Using this fact together with the finiteness of the solution set it is easy to show that the algorithm terminates in a finite number of iterations with an δ-optimal solution to the SAA problem (3.1).

While the Benders decomposition algorithm is a finite scheme, the number of iterations required may be too large in practice. To improve the convergence behavior of the generic Benders decomposition algorithm outlined above, we use a number of acceleration techniques. These strategies are described next.

Trust Region:
An undesirable feature of cutting plane algorithms such as Benders decomposition is that in the early iterations, the solutions tend to oscillate wildly from one region of the feasible set to another, thereby slowing convergence [13]. For continuous problems, this drawback is effectively controlled either by adding a regularizing term to the objective of the master problem that penalizes the ℓ_2-distance of the master problem solution from the previous solution [13, 25]; or by adding constraints that bound the ℓ_∞ distance of master problem variables from the previous solution to within a trust region [19]. These extensions prevent the master problem solution from moving too far from the previous iterate. The cutting plane algorithm is modified to allow for increasing or decreasing the regularizing penalty or trust region size based upon the progress. By proper control of these, convergence of the modified algorithm can be ensured [19, 25].

In our supply chain network design problem, the first-stage variables are binary. In this case, adding a ℓ_2 regularizing term would lead to a mixed-integer quadratic master problem. On the other hand, if a ℓ_∞ trust region is used, then since the feasible master problem solutions are all extreme points of the $[0,1]^{|\mathcal{P}|}$ hypercube, a trust region with size greater than or equal to one would include all binary solutions, and a trust region with size less than one would include only the previous feasible solution. Consequently, such a scheme is not meaningful. In our implementation, we use a trust region that bounds the Hamming distance [11] of the next master problem solution from the previous solution. Suppose y^i is the master problem solution obtained in the i-th iteration and let $Y^i := \{j \ : \ y^i_j = 1\}$. We impose the following trust region constraint in the master problem of iteration $i+1$:

$$\sum_{j \in Y^i} (1 - y_j) + \sum_{j \notin Y^i} y_j \leq \Delta^i,$$

where $\Delta^{i+1} < |\mathcal{P}|$ is the trust region size at iteration $i+1$. The trust region size can be constant or dependent on the iteration. Unfortunately, unlike in case of continuous first-stage variables, convergence of the algorithm cannot be ensured if a non-redundant trust region is maintained throughout the algorithm. Since the initial iterations exhibit poor convergence behavior, we impose the trust region in the initial iterations, and drop it once the iterates have stabilized.

Knapsack Inequalities:
Let $\theta \geq a_i^T y + b_i$ be the optimality cut derived at the end of the i-th iteration, and let ub be the current best known upper bound. Since $ub \geq c^T y + \theta$, we have the following valid inequality for the master problem in iteration $i+1$:

$$\lfloor (c + a_i)^T \rfloor y \leq \lfloor ub - b_i \rfloor,$$

where $\lfloor a \rfloor$ is the component-wise floor of a. If a good upper bound ub is available, then adding the above knapsack inequality along with the optimality cut can have a significant impact in generating

a good quality solution from the master problem in iteration $i+1$. A state-of-the-art solver such as CPLEX can derive a variety of valid inequalities from the above knapsack inequality, and expedite convergence.

Upper Bounding Heuristic:
The upper bound and incumbent solution identified in Step 2 of the Benders decomposition algorithm corresponds to the solution y^i of the master problem. This solution may be quite far from the optimal solution, and may cause the algorithm to explore inferior parts of the feasible region. If a good solution is available through some heuristic, we can replace y^i by this solution in Step 2 and proceed from there. Our implementation makes use of a particular heuristic strategy for this purpose which is described next.

In our implementation of the generic Benders decomposition algorithm, we observed that after the optimality gap becomes quite small, the upper bound exhibits a tailing off behavior as the iterations progresses. This is due to the fact, that in these iterations, the incumbent solutions all have identical or very similar major configuration decisions (corresponding to the processing centers) solutions, and the only changes are due to the minor configuration decisions (corresponding to the machines). Since the minor configuration decisions have relatively small implication on the objective, the upper bound changes very little. To avoid this behavior, we call the following heuristic strategy in Step 2 if the upper bound has not improved in several successive iterations.

Step 1: Fix all the major configuration decisions to those in the incumbent solution.

Step 2: Consider a subset of the sampled scenarios, and solve the corresponding deterministic equivalent problem to solve for the minor configuration decisions.

Step 3: Evaluate the objective function corresponding to the solution (major and minor configuration) obtained above by solving all the subproblems as in Step 2. If the solution is better than y^i (the current master problem solution), replace y^i by this solution. Update upper bound and incumbent solution if necessary and proceed as before.

Cut Strengthening:
Note that the subproblem (2.17)-(2.22) has a network substructure. Typically such problems have multiple dual optimal solutions. Consequently, there may be a number of alternatives for the optimality cut computed in Step 3. While all of these alternative cuts are valid, and exact at the current solution y^i, one cut may be *dominated* by another in the vicinity of the optimal solution. For example, let $(\acute{a}, \acute{b})$ and $(\grave{a}, \grave{b})$ be two alternative optimality cut coefficients at y^i corresponding to alternative sets of dual optimal solutions, then

$$\acute{a}^T y^i + \acute{b} = \grave{a}^T y^i + \grave{b}.$$

Suppose y^* is the optimal solution, then cut $(\acute{a}, \acute{b})$ dominates cut $(\grave{a}, \grave{b})$ if

$$\acute{a}^T y^* + \acute{b} > \grave{a}^T y^* + \grave{b}.$$

Clearly cut $(\acute{a}, \acute{b})$ is preferable in iteration i since it will typically lead to better lower bounds and expedite convergence. Unfortunately, since the optimal solution y^* is not known *a priori* it is

difficult to identify dominating cuts. Magnanti and Wong [20] considered a *core point*, i.e., a point in the relative interior of the convex hull of feasible region, as a proxy for the optimal solution y^*. They proved that if a cut is selected such that it attains the maximum value at a core point amongst the set of all alternative cuts, then such a cut is not dominated by other cuts at any feasible solution. Such a cut is termed pareto-optimal. We follow this strategy to identify good dual solutions and hence pareto-optimal cuts. Typically, a core point is not easy to identify. In our implementation, we choose a fractional optimal solution from the LP relaxation of the master problem y^0 as a candidate core point. Although, such a solution is not guaranteed to be a core point, it is often in the neighborhood of the integer optima. Furthermore, after some cuts are added to the master problem, the LP relaxation solution will typically be in the interior of the convex hull of Y, and hence satisfy the requirement of being a core point. After solving the subproblem (2.17)-(2.22) to compute $Q(y^i, \xi^n)$ in Step 2, we solve the following linear program to identify good dual solutions.

$$\begin{aligned} \max_{\mu,\nu,\pi,\rho} \quad & (\rho^T M^n) y^0 + \nu^T d^n + \pi^T s^n \\ \text{s.t.} \quad & N^T \mu + D^T \nu + S^T \pi + R^T \rho \leq q^n \\ & \nu \leq h \\ & \mu \text{ unrestricted}, \ \nu \geq 0, \ \pi \leq 0, \ \rho \leq 0 \\ & (\rho^T M^n) y^i + \nu^T d^n + \pi^T s^n = Q(y^i, \xi^n). \end{aligned}$$

The first three constraints of the above linear program enforces the dual feasible region of the subproblem (2.17)-(2.22) corresponding to y^i and ξ^n, and the fourth constraint restricts the feasible dual solutions to the set of alterative dual optima. The objective function corresponds to maximizing the cut value at y^0. Note that, the above scheme for identifying alternative dual solutions does not prevent the Benders decomposition algorithm to converge even when y^0 is not a core point.

Logistics Constraints:
In the early iterations, the master problem solved in Step 1 of the Benders decomposition algorithm includes only the logical constraints Y and a few cuts. Consequently, the configurations produced consist of very few open facilities and therefore correspond to a small objective function value (lower bound). On the other hand, for such sparse configurations, little demand can be met in the scenario subproblems, thereby resulting high shortage penalties. Therefore the resulting upper bound is very high. To avoid such bad master problem solutions, it is important to add some subproblem information into the master. However adding additional constraints and variables into the master may impede solution efficiency. Therefore, a trade-off between the master problem size and the quality of the solution generated needs to reached. Next, we derive a simple set of constraints that we found to be quite useful in improving the master problem solution.

Consider the subproblem corresponding to ξ^n, and a customer node $j \in \mathcal{C}$. Expanding the demand constraint (2.19) for this customer node, we have

$$\sum_{i \in \mathcal{P}(j)} x_{ij}^{n,k} + z_j^{n,k} \geq d_j^{n,k}, \tag{3.8}$$

where $\mathcal{P}(j)$ is the set of processing centers directly supplying to customer node j. Note that we have indexed the x and z variables also by $n \in \{1, ..., N\}$ since these variables are local to the current

subproblem only. Now consider a processing node $i \in \mathcal{P}(j)$. Expanding the capacity constraint (2.21), we have

$$\sum_{k\in\mathcal{K}} r_i^k \left(\sum_{l\in\mathcal{N}} x_{il}^{n,k} \right) \leq m_i^n y_i.$$

Since $r_k^i > 0$, and $j \in \mathcal{N}$, the above implies, for a given $k \in \mathcal{K}$, $x_{ij}^{n,k} \leq \frac{m_i^n}{r_k^i} y_i$. Summing over all $i \in \mathcal{P}(j)$, we have

$$\sum_{i\in\mathcal{P}(j)} x_{ij}^{n,k} \leq \sum_{i\in\mathcal{P}(j)} \frac{m_i^n}{r_k^i} y_i. \tag{3.9}$$

Combining inequalities (3.8) and (3.9), we have the following valid inequality

$$\sum_{i\in\mathcal{P}(j)} \frac{m_i^n}{r_k^i} y_i + z_j^{n,k} \geq d_j^{n,k} \quad \forall j \in \mathcal{C},\ k \in \mathcal{K},\ n = 1, \ldots, N.$$

Taking the probability weighted sum of the above inequalities, and substituting a new variables $\bar{z}$ as the average of all the z^n variables, we have

$$\sum_{i\in\mathcal{P}(j)} \frac{\bar{m}_i}{r_k^i} y_i + \bar{z}_j^k \geq \bar{d}_j^k \quad \forall j \in \mathcal{C},\ k \in \mathcal{K}, \tag{3.10}$$

where $\bar{d}$ and $\bar{m}_i$ are the average demand of product k for customer j, and capacity of facility i, respectively. We can add the above deterministic constraint to the master problem, along with the new variable $\bar{z}_j^k$ which captures the shortfall from meeting the average demand. The penalty of shortage also needs to be added to the master problem, i.e., the objective of the master should include $\sum_{k\in\mathcal{K}} \sum_{j\in\mathcal{C}} h_j^k \bar{z}_j^k$. However, since the shortage penalties are quite high, and we would want a configuration that is capable of satisfying at least the average demands, we drop the penalty variables $\bar{z}_j^k$, and modify the original master problem, by adding the inequalities

$$\sum_{i\in\mathcal{P}(j)} \frac{\bar{m}_i}{r_k^i} y_i \geq \bar{d}_j^k \quad \forall j \in \mathcal{C},\ k \in \mathcal{K}. \tag{3.11}$$

Cut disaggregation:
Recall that in the proposed Benders decomposition algorithm, one optimality cut is added in each iteration. The cut approximates the sample average of the second-stage value function at the current solution. Instead one could add N cuts to approximate the individual second-stage value functions corresponding to each of the N sampled scenarios. In this case the master problem solved in Step 1 of the i-th iteration takes the form

$$\begin{aligned} \min_{y,\theta} \quad & c^T y + \frac{1}{N} \sum_{n=1}^{N} \theta_n \\ \text{s.t.} \quad & y \in Y \\ & \theta_n \geq (a_k^n)^T y + b_k^n \quad k = 1, \ldots, i, \quad n = 1, \ldots, N. \end{aligned}$$

The cut coefficients a_i^n and b_i^n are computed as

$$(a_i^n)^T = (\rho_{i-1}^n)^T M^n, \quad \text{and} \quad b_i^n = (\nu_{i-1}^n)^T d^n + (\pi_{i-1}^n)^T s^n,$$

where $(\nu_{i-1}^n, \pi_{i-1}^n, \rho_{i-1}^n)$ are the optimal dual solutions for the subproblem corresponding to y^{i-1} and ξ^n solved in Step 2 of iteration $(i-1)$. The disaggregation of the optimality cut can provide a better approximation of the sample average of the second-stage value functions, and thereby improve convergence, at the expense of a larger master problem. The trade-off lies in between the fewer number of outer iterations of Benders decomposition and the more computational time required to solve the master problem in each iteration. For problems where N is not too large, this approach may be viable. This variant of Benders decomposition algorithm is often referred to as the multi-cut version [5]. A similar cut disaggregation approach has been shown to be very effective in Benders decomposition methods for solving deterministic supply chain design problems in [7].

4 Computational Results

In this section we describe numerical experiments using the proposed methodology for solving two realistic supply chain design problems. We first describe the characteristics of the test problems and some implementation details, then demonstrate the computational efficiencies afforded by our method, and finally comment on the quality of the stochastic programming solution in comparison to those obtained using a deterministic approach.

4.1 Data and Implementation

Our first test problem is that of the design of a domestic supply chain network for a U.S. company that supplies cardboard packages to breweries and soft-drink manufacturers. The problem is adapted from [7]. Henceforth this problem is referred to as the "domestic" problem.

Our second test problem is a global supply chain network design problem described in [32]. The chain encompasses the U.S. and seven Latin American countries. This problem is referred to as the "global" problem in the remainder of this paper. The global problem is slightly different from the cost-minimization formulation (2.15)-(2.22). This problem involves maximizing the expected net cash flow (NCF) after subtracting the first-stage design costs from the expected after-tax profits (revenues minus the processing and transportation costs) of the second-stage operational decisions. The second-stage problem involves maximizing after tax profits. The model formulation is

$$\begin{aligned} \max_{y} \quad & \left\{ f(y) := -c^T y + \mathbb{E}[Q(y, \boldsymbol{\xi})] \right\} \\ \text{s.t.} \quad & y \in Y \subseteq \{0,1\}^{|\mathcal{P}|}, \end{aligned}$$

where $Q(y, \xi)$ is the optimal value of the following problem

$$\begin{aligned} \max_x \quad & q^T x \\ \text{s.t.} \quad & Nx = 0 \\ & Dx \leq d \\ & Sx \leq s \\ & Rx \leq My \\ & x \in \mathbb{R}_+^{|\mathcal{A}|\times|\mathcal{K}|}. \end{aligned}$$

Here the parameters q denote per unit after-tax profits. These parameters are computed after taking into account the import and export taxes and currency exchange rates across different countries of

the supply chain network. Note that it is not required that all demand be satisfied, and hence no shortage penalties are imposed. It should be clear that conceptually the above model is no different from the cost minimization formulation (2.15)-(2.22).

The supply chain topology with all possible center locations and transportation channels for the domestic and global problem are presented in Figures 1 and 2, respectively. The main characteristics of these two networks are presented in Table 1.

	Domestic	Global
Product types	13	29
Total facilities	142	87
Internal suppliers	2	6
Manufacturing plants	8	8
Machines	28	10
Finishing facilities	9	10
Final process machines	93	36
Warehouses	2	17
Customers	238	17
Transportation channels	1,559	239
Country	1	7

Table 1: Supply chain network characteristics

We used the same data for the deterministic parameter values as in [7] (for the domestic problem) and [32] (for the global problem). To obtain stochastic instances of the problems, we assumed that the product demands and facility capacities are uncertain. We used log-normal distributions for these uncertain parameters. The non-negativity of the parameter values are preserved by using log-normal distributions. Furthermore, recently Kamath and Pakkala [15] provided evidence that log-normal distributions are well-suited for modelling economic stochastic variables such as demands. We used the values for the demand and capacity parameters used in the deterministic models of [7] and [32] as the mean values for the random parameters. The standard deviations for the distributions were chosen as certain fractions of the mean values.

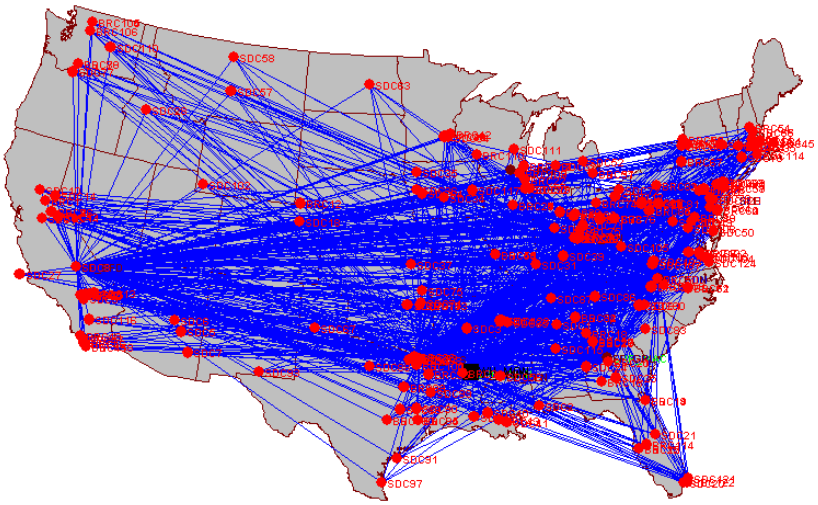

Figure 1: Supply chain network for the domestic problem

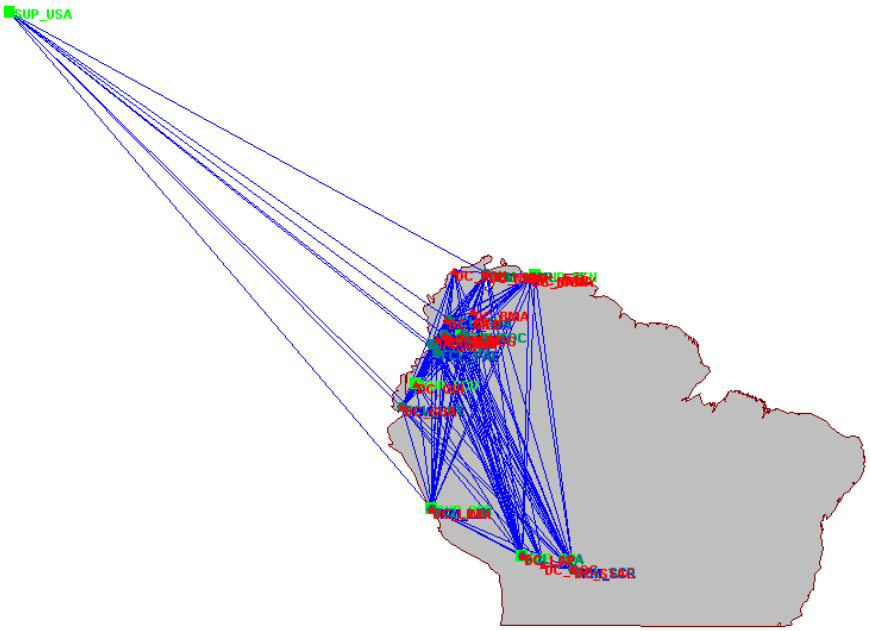

Figure 2: Supply chain network for the global problem

Recall from Section 3.1 that the SAA method calls for the solution of M instances of the approximating stochastic program (3.1), each involving N sampled scenarios. Statistical validation of a candidate solution is then carried out by evaluating the objective function using N' sampled scenarios. In our implementation, we used $N = 20, 30, 40$ and 60; $M = 20$; and $N' = 1000$. As will be evident in our numerical results, these parameters were sufficient to obtain good quality solutions with high confidence levels. To illustrate the complexity of solving (3.1) within the SAA scheme, we present the sizes of the deterministic equivalents of the SAA problems corresponding to the different values of N in Table 2.

	Domestic				Global			
	Constraints		Variables		Constraints		Variables	
N	Equality	Inequality	Continuous	Binary	Equality	Inequality	Continuous	Binary
1	3,498	4,324	20,912	140	1,065	402	6,824	70
20	69,960	86,480	418,240	140	21,300	8,040	136,480	70
40	139,920	172,960	836,480	140	42,600	16,080	272,960	70
60	209,880	259,440	1,254,860	140	63,900	24,120	409,440	70

Table 2: Size of the deterministic equivalent of the SAA problem

The proposed algorithmic scheme was implemented in C++ with CPLEX 7.0 [14] callable library routines for solving linear and mixed-integer programming sub-problems. All computations were carried out on a Pentium II 400 MHz PC running Windows NT.

4.2 Performance of Acceleration Techniques

Recall that, since the SAA method calls for repeated solutions of the SAA problems, computational efficiency of the solution procedure is of great significance. Here we discuss the performance of the accelerated Benders decomposition algorithm for solving the SAA problems.

Figures 3 and 4 chart the growth of the computational time with the sample size N. For each of the two problems, we compare the CPU seconds required for solving an SAA problem instance as a monolithic deterministic equivalent problem using the CPLEX MIP solver, the standard Benders decomposition algorithm, and the accelerated Benders decomposition algorithm proposed in Section 3.2. The efficacy of the proposed acceleration strategies is clearly observed.

Table 3 compares the performance of various combinations of the acceleration schemes described in Section 3.2 for solving SAA instances of 20 scenarios. The acceleration schemes are denoted as follows: LC (Logistics constraints); TR (Trust region); CD (cut disaggregation); KI (Knapsack inequalities); UH (Upper-bounding heuristic); and CS (Cut strengthening). Each row of the table presents computational performance measures for various combinations of these acceleration schemes. The trust region scheme was not effective for the global problem and the cut disaggregation scheme was not tested for the domestic problem. Hence, the notation 'TR/CD' in the rows of Table 3 refers to the trust region scheme for the domestic problem and the cut disaggregation scheme for the global problem. The row marked 'Standard' corresponds to the standard Benders decomposition algorithm, and the row marked 'All' corresponds to all five acceleration schemes (LC, TR, KI, UH, and CS for the domestic problem, and LC, CD, KI, UH and CS for the global problem) used together. For the domestic problem, we compare the relative optimality gap after the first iteration (1^{st} Gap), the relative optimality gap after the tenth iteration (10^{th} Gap), the

total CPU seconds, and the total number of iterations. The same performance are used for global problem, except we use the relative optimality gap after the fiftieth iteration (50^{th} Gap) instead of 10^{th} Gap.

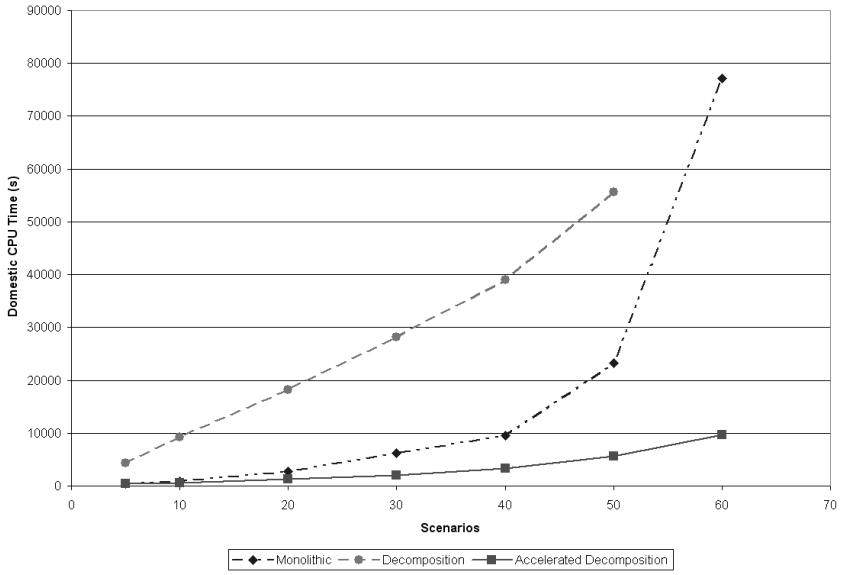

Figure 3: CPU seconds versus sample size for the domestic problem

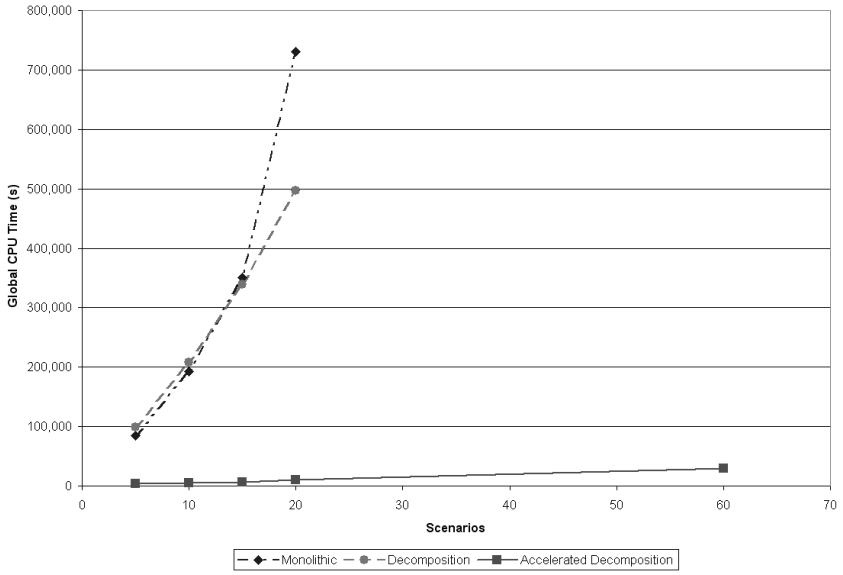

Figure 4: CPU seconds versus sample size for the global problem

Acceleration scheme	Domestic				Global			
	1^{st} Gap	10^{th} Gap	CPUs	Iter.	1^{st} Gap	50^{th} Gap	CPUs	Iter.
Standard	$> 100\%$	60%	> 4000	> 30	$> 100\%$	41%	> 13000	> 60
LC	31%	8%	> 4000	> 30	$> 100\%$	27%	> 13000	> 60
TR/CD	$> 100\%$	40%	> 4000	> 30	$> 100\%$	18%	> 13000	> 60
KI	60%	5%	> 4000	> 30	$> 100\%$	21%	> 13000	> 60
UH	$> 100\%$	60%	> 4000	> 30	$> 100\%$	41%	> 13000	> 60
CS	$> 100\%$	9%	> 4000	> 30	$> 100\%$	29%	> 13000	> 60
LC+TR/CD	31%	0.7%	> 4000	> 30	$> 100\%$	22%	11900	56
LC+KI	31%	0.1%	3860	26	$> 100\%$	12%	> 13000	> 60
LC+UH	31%	0.08%	2180	12	$> 100\%$	27%	> 13000	> 60
LC+CS	31%	0.5%	> 4000	> 30	$> 100\%$	19%	12300	58
TR/CD+KI	60%	3%	> 4000	> 30	$> 100\%$	5%	> 13000	> 60
LC+TR/CD+KI	31%	0.2%	3600	23	$> 100\%$	3%	10300	51
LC+KI+UH	31%	0.01%	1500	8	$> 100\%$	12%	> 13000	> 60
LC+TR/CD+KI+UH	31%	0.01%	1380	7	$> 100\%$	4%	10300	51
LC+TR/CD+KI+CS	31%	0.06%	3050	19	$> 100\%$	$< 1\%$	9800	45
All	31%	0.01%	1890	7	$> 100\%$	$< 1\%$	9800	45

Table 3: Comparison of acceleration schemes ($N = 20$)

From Table 3 it can be observed that, although the acceleration schemes improves the convergence behavior over standard Benders decomposition, no one acceleration technique is a clear winner. Rather, the schemes are most effective in concert. In particular, for the domestic problem, the best performance could be achieved by combining LC, TR, KI, and UH. While, for the global problem, the best combination is LC, CD, KI, and CS. Although, the results in Table 3 correspond to SAA instances of sample size 20, similar behavior has been observed for problems with different sample sizes (see [26] for detailed computational results).

4.3 Quality of Stochastic Solutions

In this section we compare the solutions of the stochastic programming model to that of a deterministic optimization problem involving the mean values of the uncertain problem parameters (known as the mean-value problem). Recall that the SAA method produces a number of candidate solutions (at most M unique solutions). In Table 4, we compare statistics of the (uncertain) total cost for the domestic problem corresponding to the mean-value problem solution (denoted by y^{MVP}) to that of three candidate solutions (denoted by y^1, y^2, and y^3 respectively) identified by solving $M(= 20)$ SAA problem instances with $N = 20$. The candidate solutions were chosen as the three "best" solutions based upon their objective function value and optimality gap estimates as provided by the SAA method. As before, the total cost statistics for each solution are computed using a sample size of $N' = 1000$. The last two rows of Table 4 displays the estimated optimality gap and the standard deviation of the gap estimate (computed using (3.6) and (3.7) respectively) for the mean-value problem solution and each of the three best SAA solutions. Even for a sample size of just $N = 20$ the SAA problem solutions are very close to being optimal for the true stochastic supply chain design problem.

Table 5 compares the solution the mean-value solution to the SAA solution for the global problem. Once again a sample size of $N = 20$ was used. In this case, the SAA method identified a single "best" candidate solution (denoted by y^1). The estimated optimality gaps clearly indicate

the superiority of the SAA solution.

Cost (million \$)	y^{MVP}	y^1	y^2	y^3
Average	116.77	111.03	111.03	111.05
Std. Dev.	0.34	0.11	0.11	0.11
Min.	99.02	100.38	100.14	100.10
Max.	173.30	122.57	122.08	122.11
$\text{gap}_{N,M,N'}$	5.91	0.16	0.17	0.18
σ_{gap}	0.79	0.04	0.04	0.04

Table 4: Costs statistics for candidate solutions to the domestic problem

NCF (million \$)	y^{MVP}	y^1
Average	51.03	54.002
Std. Dev.	0.126	0.124
Min.	35.243	43.149
Max.	64.290	69.403
$\text{gap}_{N,M,N'}$	3.18	0.21
σ_{gap}	0.54	0.05

Table 5: NCF statistics for candidate solutions to the global problem

Tables 4 and 5 reveal that the solutions to the stochastic programming model are not only superior to the mean-value problem solution in terms of the optimality gap corresponding to the expected cost/NCF objective, but these solutions also lead to comparatively smaller variability of the cost/NCF objective.

In Tables 6 and 7, we present the optimality gap estimates of the "best" solution identified by the SAA method for different sample sizes N. It is clear that the SAA method with only a modest number of sampled scenarios can provide very high quality solutions to the true stochastic supply chain design problem involving potentially infinite number of scenarios.

N	$\text{gap}_{N,M,N'}$	σ_{gap}
20	0.16	0.04
30	0.11	0.03
40	0.08	0.02
60	0.07	0.02

Table 6: Optimality gap estimates for the domestic problem

N	$\text{gap}_{N,M,N'}$	σ_{gap}
5	1.24	0.22
20	0.21	0.05
60	0.09	0.02

Table 7: Optimality gap estimates for the global problem

Finally, to observe the effect of the variability of the uncertain problem parameters, we considered three different levels of the variability of the uncertain problem parameters. For each level, we considered the best candidate solution obtained from solving SAA problems of a given sample size N. In Figures 5 and 6, the ranges for the cost and NCF corresponding to these candidate solutions are compared against those of the corresponding mean-value problem solution. As before these statistics are computed using a sample size of $N' = 1000$. As mentioned earlier, the stochastic solutions have significantly smaller ranges than the mean-value problem solution. Furthermore, with the increase in the variability of the uncertain parameters, the cost/NCF variability (range) for the stochastic solutions increases at a slower rate than that of the mean-value problem solution. In Figures 7 and 8, the worst-case cost and NCF corresponding to the candidate stochastic solutions are compared against those of the corresponding mean-value problem solution. Once again, it is clear that the stochastic solution is more resilient to the variability of the problem parameters. For the global problem, the resiliency is improved by including more scenarios in the SAA problem.

5 Concluding Remarks

In this paper, we have developed a practical methodology for large-scale supply chain network design problems under uncertainty. The method integrates an accelerated decomposition scheme along with the recently developed sample average approximation method. The proposed methodology provides an efficient framework for identifying and statistically testing a variety of candidate design solutions. We have provided empirical results for the design two realistic supply chain networks. Our results reveal the computational efficacy of the proposed method. Furthermore, we have demonstrated that the candidate solutions identified by the proposed method are not only superior to traditional mean-value problem solutions in an expectation sense, but also result in significantly smaller cost/cash flow variability, and this reduction is more pronounced in case of higher variability in the uncertain environment.

Acknowledgements

This work has been funded in part by a grant from the Sloan Foundation to the Center for Paper Business and Industry Studies (CPBIS) at the Institute of Paper Science and Technology (IPST), and by the National Science Foundation under grants DMI-0099726 and DMI-0133943.

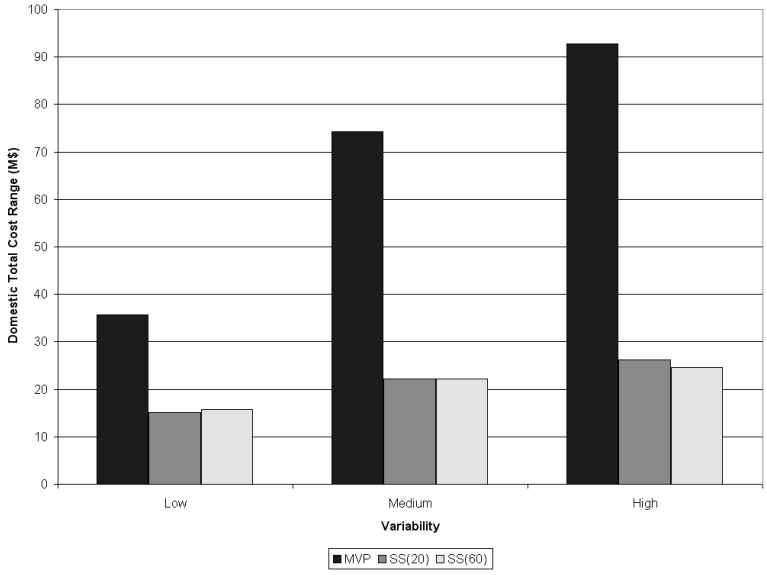

Figure 5: Effect of variability on the cost range for the domestic problem

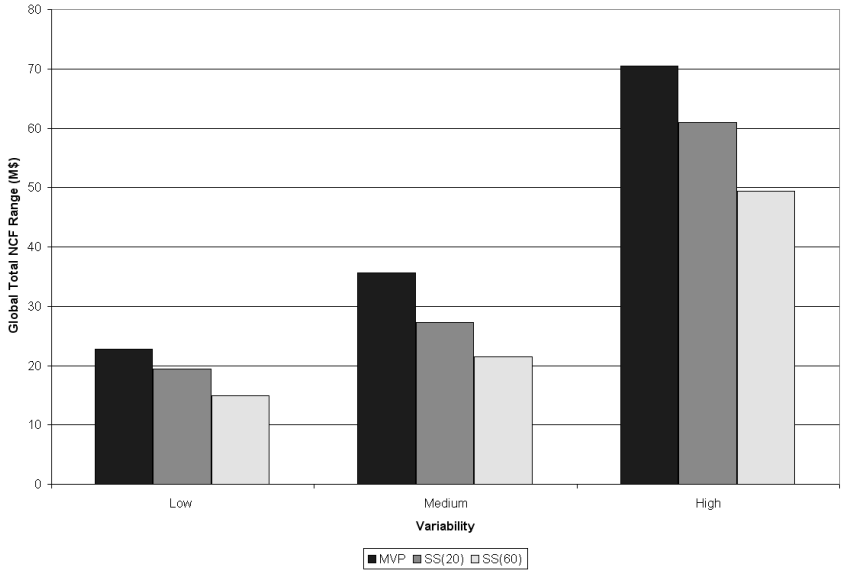

Figure 6: Effect of variability on the NCF range for the global problem

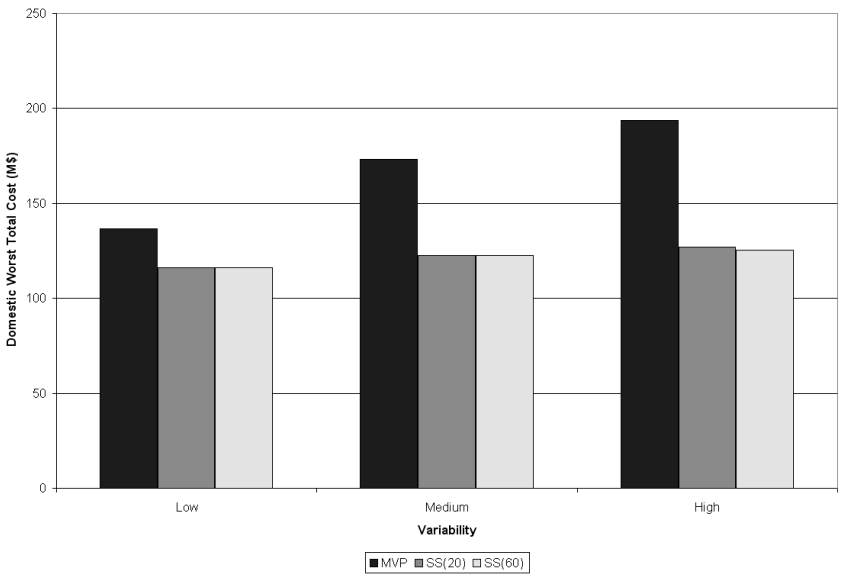

Figure 7: Effect of variability on the worst-case cost for the domestic problem

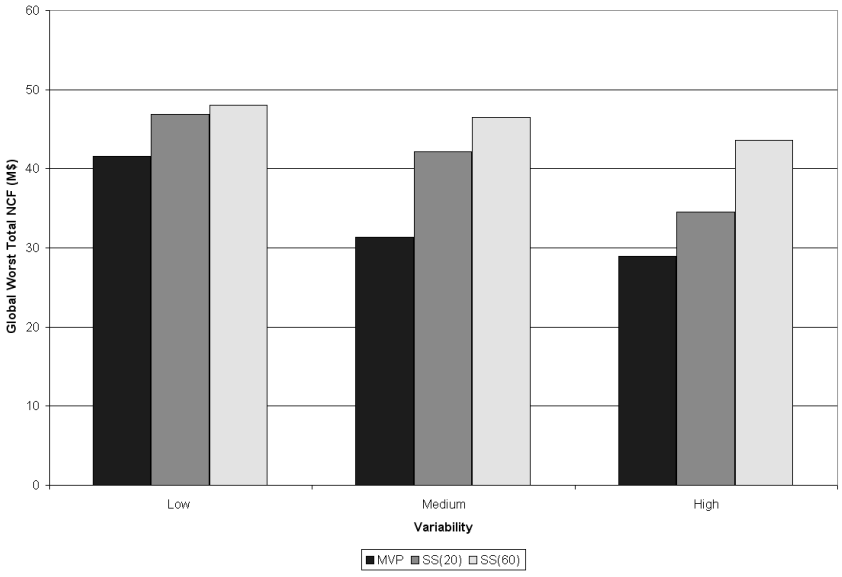

Figure 8: Effect of variability on the worst-case NCF for the global problem

References

[1] C.H. Aikens. Facility location models for distribution planning. *European Journal of Operational Research*, 22(3):263–279, 1985.

[2] A. Alonso-Ayuso, L.F. Escudero, A. Garin, M.T. Ortuno, and G. Perez. An approach for strategic supply chain planning under uncertainty based on stochastic 0-1 programming. Technical report, Universidad Rey Juan Carlos, Madrid, Spain, 2002.

[3] J. F. Benders. Partitioning procedures for solving mixed variables programming problems. *Numersiche Mathematik*, 4:238–252, 1962.

[4] J. R. Birge and F. Louveaux. *Introduction to Stochastic Programming*. Springer, New York, NY, 1997.

[5] J. R. Birge and F. V. Louveaux. A multicut algorithm for two-stage stochastic linear programs. *European Journal of Operational Research*, 34(3):384–392, 1988.

[6] R. K-M. Cheung and W.B. Powell. Models and algorithms for distribution problems with uncertain demands. *Transportation Science*, 30:43–59, 1996.

[7] K. Dogan and M. Goetschalckx. A primal decomposition method for the integrated design of multi-period production-distribution systems. *IIE Transactions*, 31:1027–1036, 1999.

[8] A. M. Geoffrion and G.W. Graves. Multi-commodity distribution system design by Benders decomposition. *Management Science*, 20:822–844, 1974.

[9] A. M. Geoffrion and R. F. Powers. Twenty years of strategic distribution system design: An evolutionary perspective. *Interfaces*, 25:105–128, 1995.

[10] G. J. Gutierrez, P. Kouvelis, and A. A. Kurawala. A robustness approach to uncapacitated network design problems. *European Journal of Operations Research*, 94:362–376, 1996.

[11] R. W. Hamming. Error detecting and error correcting codes. *Bell System Tech Journal*, 9:147–160, 1950.

[12] M. Hicks. When the chain snaps. *eWeek* <`http://www.eweek.com/article2/0,3959,19768,00.asp`>, 2002.

[13] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms II*. Springer-Verlag, Berlin Heidelberg, 1996.

[14] ILOG Inc. *CPLEX 7.0 User's Manual*. ILOG CPLEX Division, Incline Village, NV, 2000.

[15] K. R. Kamath and T.P.M. Pakkala. A Bayesian approach to a dynamic inventory model under an unknown demand distribution. *Computers & Operations Research*, 29:403–422, 2002.

[16] A. J. Kleywegt, A. Shapiro, and T. Homem-De-Mello. The sample average approximation method for stochastic discrete optimization. *SIAM Journal of Optimization*, 12:479–502, 2001.

[17] H. Van Landeghem and H. Vanmaele. Robust planning: A new paradigm for demand chain planning. *Journal of Operations Management*, 20:769–783, 2002.

[18] J. T. Linderoth, A. Shapiro, and S. J. Wright. The empirical behavior of sampling methods for stochastic programming. Optimization Technical Report 02-01, Computer Sciences Department, University of Wisconsin-Madison, 2002.

[19] Jeffrey T. Linderoth and Stephen J. Wright. Implementing a decomposition algorithm for stochastic programming on a computational grid. *Computational Optimization and Applications*, 24:207–250, 2003.

[20] T. L. Magnanti and R. T. Wong. Accelerating benders decomposition: Algorithmic enhancement and model selection criteria. *Operations Research*, 29:464–484, 1981.

[21] W. K. Mak, D. P. Morton, and R. K. Wood. Monte Carlo bounding techniques for determining solution quality in stochastic programs. *Operations Research Letters*, 24:47–56, 1999.

[22] S. A. MirHassani, C. Lucas, G. Mitra, E. Messina, and C. A. Poojari. Computational solution of capacity planning models under uncertainty. *Parallel Computing*, 26:511–538, 2000.

[23] V. I. Norkin, G. Ch. Pflug, and A. Ruszczyński. A branch and bound method for stochastic global optimization. *Mathematical Programming*, 83:425–450, 1998.

[24] S. H. Owen and M. S. Daskin. Strategic facility location: A review. *European Journal of Operational Research*, 111:423–447, 1998.

[25] A. Ruszczyński. A regularized decomposition method for minimizing a sum of polyhedral functions. *Mathematical Programming*, 35:309–333, 1986.

[26] T. Santoso. *A comprehensive model and efficient solution algorithm for the design of global supply chains under uncertainty.* PhD thesis, School of Industrial & Systems Engineering, Georgia Institute of Technology, 2002.

[27] A. Shapiro and T. Homem de Mello. A simulation-based approach to stochastic programming with recourse. *Mathematical Programming*, 81:301–325, 1998.

[28] P. Tsiakis, N. Shah, and C. C. Pantelides. Design of multi-echelon supply chain networks under demand uncertainty. *Industrial & Engineering Chemistry Research*, 40:3585–3604, 2001.

[29] R. Van Slyke and R. J.-B. Wets. L-Shaped linear programs with applications to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics*, 17:638–663, 1969.

[30] B. Verweij, S. Ahmed, A. J. Kleywegt, G. Nemhauser, and A. Shapiro. The sample average approximation method applied to stochastic routing problems: a computational study. *Computational Optimization and Applications*, 24:289–333, 2003.

[31] C. Vidal and M. Goetschalckx. Strategic production-distribution models: A critical review with emphasis on global supply chain models. *European Journal of Operational Research*, 98:1–18, 1997.

[32] C. Vidal and M. Goetschalckx. A global supply chain model with transfer pricing and transportation cost allocation. *European Journal Of Operational Research*, 129:134–158, 2001.

[33] C-S. Yu and H-L. Li. A robust optimization model for stochastic logistic problems. *International Journal of Production Economics*, 64:385–397, 2000.