Proceedings of the 2006 Winter Simulation Conference
L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, eds.

THE "BEST" ALGORITHM FOR SOLVING STOCHASTIC MIXED INTEGER PROGRAMS

Susan M. Sanchez
R. Kevin Wood

Operations Research Department
Naval Postgraduate School
Monterey, CA 93943, U.S.A.

ABSTRACT

We present a new algorithm for solving two-stage stochastic mixed-integer programs (SMIPs) having discrete first-stage variables, and continuous or discrete second-stage variables. For a minimizing SMIP, the **BEST** algorithm (1) computes an upper **B**ound on the optimal objective value (typically a probabilistic bound), and identifies a deterministic lower-bounding function, (2) uses the bounds to **E**numerate a set of first-stage solutions that contains an optimal solution with pre-specified confidence, (3) for each first-stage solution, **S**imulates second-stage operations by repeatedly sampling random parameters and solving the resulting model instances, and (4) applies statistical **T**ests (e.g., "screening procedures") to the simulated outcomes to identify a near-optimal first-stage solution with pre-specified confidence. We demonstrate the algorithm's performance on a stochastic facility-location problem.

1 INTRODUCTION

Stochastic mixed-integer programs (SMIP)s arise in varied contexts such as industrial capacity planning (e.g., Stafford 1997, Ahmed, et al. 2000), vehicle routing or allocation (e.g., Frantzeskakis and Powell 1990, Morton and Kenyon 2001), facility location (e.g., Laporte et al. 1994), and network interdiction (Cormican et al. 1998, Israeli 1999). The common thread is that, in the problems's first stage, the user must make discrete, resource-constrained decisions about the configuration of a system, and in the second stage uncertainty resolves itself and the user ("adversary" in the case of interdiction problems) operates the configured system optimally. The second-stage decision variables may be continuous or discrete. This paper standardizes on a minimizing SMIP (e.g., minimizing cost).

Most of the literature on SMIPs focuses on solving problems for which all second-stage scenarios can be enumerated (Klein Haneveld and van der Vlerk 1998, Ahmed 2004). Two exceptions include *sequential approximation* (SA) (e.g., Kall et al. 1988) and the *sample average approximation method* (SAAM) (Mak et al. 1999, Kleywegt et al. 2001). SA sequentially improves lower and upper bounds by partitioning the state space of the random parameters. Typically, SA solves a lower-bounding problem defined across n "conditional-average scenarios," and must increase n to refine the partition and tighten the bound. But, the computational workload tends to increase superlinearly in n. SAAM solves similar, n-scenario problems but with sampled scenario data. It too must increase n for better accuracy, and therefore suffers from the computational difficulities that SA exhibits. Both methods may require specialized techniques to handle integer second-stage variables. Our goal is to develop an easy-to-implement alternative to SA and SAAM that does not suffer from their drawbacks.

We develop a fundamentally new algorithm for solving SMIPs, and call it **BEST**: *Bound, Enumerate, Simulate and Test*. **BEST** asks the user to preselect $\epsilon > 0$ and $\alpha > 0$, and then produces an ϵ-optimal solution with a lower bound on the confidence level of approximately $1 - \alpha$; hereafter we refer to this bound as the "approximate confidence level." For most practical problems, the approximate confidence level is likely to be conservative. Furthermore, **BEST** may produce a truly optimal solution with an approximate confidence level that is strictly greater than $1 - \alpha$.

In its basic form, **BEST** first computes a global upper bound (typically probabilistic) on the SMIP's optimal objective value z^*. It also identifies a lower-bounding model whose solution, for fixed first-stage variables, provides a restricted lower bound on z^*. **BEST** then applies these bounds to enumerate a candidate set of first-stage solutions that, with pre-specified confidence, contains at least one optimal solution. Typically, the candidate set is not a singleton, so **BEST** proceeds by simulating second-stage operations for each candidate: A Monte Carlo simulation samples random second-stage parameters, and solves the resulting optimization problems. The algorithm then applies a statistical test—we use bootstrapping—to screen out solutions that are unlikely to be optimal. If a single can-

1-4244-0501-7/06/$20.00 ©2006 IEEE

didate remains, that solution can be declared to be optimal with high confidence. Even if more than a single candidate remains, a second test may enable us to declare the "apparent best solution," i.e., the solution having the smallest average objective value, to be ϵ-optimal with pre-specified confidence. If that test is not satisfied, the method specifies a second round of simulation and testing that makes such a declaration valid.

BEST places modest requirements on the types of SMIPs it can solve. A deterministic lower bound must be available as a function of the first-stage variables. And, the expected cost of the SMIP, given a fixed first-stage solution, should be reasonably easy to estimate using Monte Carlo simulation. The SMIP should also incorporate "relatively complete recourse," defined below. Typically, we solve SMIPs with linear constraints and linear objective functions, but "linear" is not an inherent requirement. **BEST** does not require the solution of any multi-scenario models as do SA and SAAM: This is its key computational advantage.

We seek to make the statistical-testing portions of **BEST** easily accessible the optimization community, so we propose a novel vector bootstrap approach to screen candidate solutions. This eliminates the need for parametric characterizations of the joint distribution of objective-function values across candidate solutions (e.g., constant variance), and it means that **BEST** can be implemented without specialized statistical functions or tables. In fact, it can be implemented entirely within an algebraic modeling system such as GAMS (Brooke et al. 1992).

The rest of the paper is outlined as follows. "Preliminaries," Section 2, specifies the general formulation of SMIP and describes the stochastic facility-location problem we use throughout for illustrative purposes. Section 3 outlines the **BEST** algorithm, and briefly describes the bounds, enumeration mechanism, and statistical testing methods we use. Section 4 presents computational results. Section 5 provides conclusions and discusses directions for further research.

2 PRELIMINARIES

2.1 The Stochastic Mixed-Integer Program (SMIP)

We wish to solve, at least approximately, a two-stage SMIP with discrete first-stage variables

$$\mathbf{x} \in X \equiv \{\mathbf{x} \in Z^{n_1} | A\mathbf{x} = \mathbf{b},\ \mathbf{0} \le \mathbf{x} \le \mathbf{u}\}, \tag{1}$$

and with continuous or discrete second-stage variables

$$\mathbf{y} \in Y \equiv \{\mathbf{y} \in R_+^{n_2} | \text{ some } y_j \text{ may be integer}\}. \tag{2}$$

Using tildes to identify random parameters, this SMIP is

$$\text{SMIP} \quad \min_{\mathbf{x}\in X} Eh(\mathbf{x}, \tilde{\boldsymbol{\xi}}), \text{ where} \tag{3}$$

$$h(\mathbf{x}, \tilde{\boldsymbol{\xi}}) = \mathbf{c}^T\mathbf{x} + \min_{\mathbf{y}\in Y} \tilde{\mathbf{f}}^T\mathbf{y} \tag{4}$$

$$\text{s.t. } \tilde{D}\mathbf{y} = \tilde{\mathbf{d}} + \tilde{B}\mathbf{x}, \tag{5}$$

and where $\tilde{\boldsymbol{\xi}} = \text{vec}(\tilde{\mathbf{f}}, \tilde{D}, \tilde{\mathbf{d}}, \tilde{B})$. Typically, $\mathbf{u} = \mathbf{1}$, i.e., all first-stage variables are binary. We assume that, with probability one, the second-stage problem in $\mathbf{y}$ has a bounded, feasible solution for any $\mathbf{x} \in X$. Thus, SMIP is a two-stage stochastic program with *relatively complete recourse* (Rockafellar and Wets 1976).

We use a simple example of an SMIP throughout the paper to illustrate the **BEST** approach, a single-product, capacitated *stochastic facility-location problem* (SFLP). Birge and Louveaux (1997, pp. 57-59) and Laporte et al. (1994) describe similar models:

Stochastic Facility-Location Problem (SFLP)

Indices:

$i \in I$	candidate facilities, e.g., warehouses
$j \in J$	customer zones

Inputs:

c_i	deterministic cost to construct facility i (\$)
b	maximum number of facilities
u_i	planned capacity of facility i if built (tons)
f_{ij}	deterministic shipping cost from i to j (\$/ton)
$\tilde{d}_j$	random demand in customer zone j (tons)
r_j	penalty for unmet demand at j (\$/ton)

Decision Variables:

x_i	1 if facility i is built; 0 otherwise (1st stage)
y_{ij}	tons shipped from i to j (2nd stage)
v_j	tons unmet demand at zone j (2nd stage)

Formulation

$$z^* = \min_{\mathbf{x}\in\{0,1\}^{|I|}} E\,h(\mathbf{x}, \tilde{\mathbf{d}})$$

$$\text{s.t.} \sum_{i\in I} x_i \le b, \quad \text{where} \tag{6}$$

$$h(\mathbf{x}, \tilde{\mathbf{d}}) = \sum_{i\in I} c_i x_i + \tag{7}$$

$$\min_{\mathbf{y}\ge\mathbf{0},\mathbf{v}\ge\mathbf{0}} \sum_{i\in I}\sum_{j\in J} f_{ij}y_{ij} + \sum_{j\in J} r_j v_j$$

$$\text{s.t.} \sum_{j\in J} y_{ij} \le u_i x_i, \quad \forall\ i \in I \tag{8}$$

$$\sum_{i \in I} y_{ij} + v_j = \tilde{d}_j, \quad \forall\ j \in J \qquad (9)$$

In the first stage of SFLP, we choose which facilities to construct, but in the face of uncertain future demands for product. In the second stage, actual demands are realized and the constructed facilities ship to meet those demands as cheaply as possible. A penalty is paid for each unit of unmet demand. For fixed $\mathbf{x}$, the deterministic version of SFLP is a transportation problem with elastic demand constraints.

3 THE "BEST" ALGORITHM

BEST is based on this self-evident proposition:

Proposition 1 *Suppose z'' and $h'(\mathbf{x})$ are defined for SMIP such that $z'' \geq z^*$ with confidence $1-\alpha_u$, and $h'(\mathbf{x}) \leq Eh(\mathbf{x}, \tilde{\boldsymbol{\xi}})$ for all $\mathbf{x} \in X$. Enumerate $\mathcal{X} = \{\hat{\mathbf{x}} \in X | h'(\hat{\mathbf{x}}) \leq z''\}$ and assume $\mathcal{X} \neq \emptyset$. Then, with confidence $1-\alpha_u$, $\mathcal{X}$ contains at least one optimal solution to SMIP.* ∎

Our methods for computing z'' will never give $\mathcal{X} = \emptyset$, so we can now provide a well-defined outline of **BEST**.

Algorithm BEST
Input: Data for an instance of SMIP; confidence values α_u, α_s, and α_t for bounding, initial testing ("screening"), and final testing, respectively, all chosen so that $1-(\alpha_u+\alpha_s+\alpha_t)$ equals the desired overall confidence $1-\alpha$; allowable optimality gap $\epsilon > 0$; initial sample size n_0.
Output: A solution $\hat{\mathbf{x}}$ to SMIP that is optimal with with confidence at least $(1-\alpha_u)(1-\alpha_s)$, or is ϵ-optimal with confidence at least $(1-\alpha_u)(1-\alpha_s)(1-\alpha_t)$.
{

Call **Bound** to compute z'', an upper bound on z^* having confidence level $1-\alpha_u$;

Call **Enumerate** to find the *initial candidate set* of solutions $\mathcal{X} = \{\hat{\mathbf{x}} \in X | h'(\hat{\mathbf{x}}) \leq z''\}$. $\mathcal{X}$ contains an optimal solution with confidence $1-\alpha_u$;

If $\mathcal{X} = \{\hat{\mathbf{x}}_{[1]}\}$, set $\epsilon \leftarrow \alpha_s \leftarrow \alpha_t \leftarrow 0$ and go to End;

Call **Simulate** to generate samples $\hat{\boldsymbol{\xi}}_n$, $n = 1, \ldots, n_0$ of $\tilde{\boldsymbol{\xi}}$, and to evaluate $h(\hat{\mathbf{x}}, \hat{\boldsymbol{\xi}})$ for each sample and each $\hat{\mathbf{x}} \in \mathcal{X}$;

Call **Test1** with observations from **Simulate** to screen out convincingly inferior solutions, leaving the *selected subset* $\mathcal{X}^* \subseteq \mathcal{X}$. $\mathcal{X}^*$ contains an optimal solution with (approximate) confidence at least $(1-\alpha_u)(1-\alpha_s)$;

If $\mathcal{X}^* = \{\hat{\mathbf{x}}_{[1]}\}$, set $\epsilon \leftarrow \alpha_t \leftarrow 0$ and go to End;

Call **Test2** with parameter $\epsilon > 0$, input $\mathcal{X}^*$ and observations on $\hat{\mathbf{x}} \in \mathcal{X}^*$ from **Simulate**;

If **Test2** returns $n^+ > 0$, call **Simulate** with $\mathcal{X}^*$ replacing $\mathcal{X}$ and n^+ replacing n_0, but compute the apparent best solution $\hat{\mathbf{x}}_{[1]}$ with respect to all $n^+ + n_0$ observations;

End: Print($\hat{\mathbf{x}}_{[1]}$, "is an", ϵ,"-optimal solution with approximate confidence", $(1-\alpha_u)(1-\alpha_s)(1-\alpha_t)$);

}

3.1 Bounds for SMIP

BEST requires a global upper bound on z^*, and a lower-bounding function on $Eh(\hat{\mathbf{x}}, \tilde{\boldsymbol{\xi}})$ for any $\hat{\mathbf{x}} \in X$. The number of possibilities is large, and many are problem dependent, so we only discuss a few options that apply to SFLP.

3.1.1 Upper Bounds

Specialized deterministic bounds could be used here, for example, the Edmunson-Madansky bound (Edmunson 1956, Madansky 1959), or a bound based on dual restricted recourse (Morton and Wood 1999). We would set $\alpha_u = 0$ if such a bound were used. However, the following, standard, probabilistic bound (e.g., Mak et al. 1999) applies to most, if not all SMIPs, and is easily described and computed.

Procedure UpperBound
Input: Coefficients and distribution parameters that define SMIP; sample size n_u; confidence parameter α_u.
Output: A probabilistic upper bound on SMIP $z'' \geq z^*$ having confidence level $1-\alpha_u$.
{

Use a heuristic to identify a "good" first-stage solution $\hat{\mathbf{x}}$ to SMIP.

According to the distribution of $\tilde{\boldsymbol{\xi}}$, generate n_u random samples, $\hat{\boldsymbol{\xi}}_1, \hat{\boldsymbol{\xi}}_2, \ldots, \hat{\boldsymbol{\xi}}_{n_u}$, and evaluate $h(\hat{\mathbf{x}}, \hat{\boldsymbol{\xi}}_\ell)$ for each;

Compute $z'' = \frac{1}{n_u}\sum_{\ell=1}^{n_u} h(\hat{\mathbf{x}}, \hat{\boldsymbol{\xi}}_\ell) + t_{\alpha_u, n_u - 1} S/\sqrt{n_u}$ where S is the sample variance estimator for $h(\hat{\mathbf{x}}, \tilde{\boldsymbol{\xi}})$ and t_{α_u, n_u-1} is the upper $1-\alpha_u$ quantile of the t distribution with $n_u - 1$ degrees of freedom;

Return z'';

}

The bound z'' is valid because $Eh(\hat{\mathbf{x}}, \tilde{\boldsymbol{\xi}}) \geq z^*$ for every $\hat{\mathbf{x}} \in X$. One heuristic that might be used to obtain an acceptable $\hat{\mathbf{x}}$ would simply solve the *expected-value problem*, which is this deterministic MIP:

$$\min_{\mathbf{x} \in X} h(\mathbf{x}, E\tilde{\boldsymbol{\xi}}). \qquad (10)$$

Or, we could solve an *approximating problem* with a modest number of samples $\hat{\boldsymbol{\xi}}_\ell$, $\ell = 1, \ldots, n'$, taken from the

distribution of $\tilde{\boldsymbol{\xi}}$ (e.g., Mak et al. 1999):

$$\min_{\mathbf{x}\in X} \frac{1}{n'} \sum_{\ell=1}^{n'} h(\mathbf{x}, \hat{\boldsymbol{\xi}}_\ell). \tag{11}$$

This is also a deterministic MIP. Our computational examples in Section 4 exploit the latter technique.

3.1.2 Lower Bounds

Depending on where an SMIP's random coefficients appear, **BEST** may be able to use lower bounds based on Jensen's inequality (e.g., Birge and Louveaux 1997, pg. 140) or dual restricted recourse (Morton and Wood 1999). Because $h(\mathbf{x}, \tilde{\mathbf{d}})$ for SFLP is convex in $\tilde{\mathbf{d}}$, Jensen's inequality applies:

Proposition 2 For SFLP, $h'(\mathbf{x}) \equiv h(\mathbf{x}, E\tilde{\mathbf{d}}) \leq Eh(\mathbf{x}, \tilde{\mathbf{d}})$. ∎

Thus, we can compute a lower bound on $Eh(\hat{\mathbf{x}}, \tilde{\mathbf{d}})$, for any $\hat{\mathbf{x}}$, by solving a single, deterministic, elastic transportation problem with demands set at expected values.

The need for a good deterministic lower bound is, admittedly, the most restrictive aspect of **BEST**. Later in the paper, we discuss how to tighten the simple, deterministic bound described above for SFLP. When simple bounds like this do not apply, we conjecture that probabilistic lower-bounding techniques will prove useful (Bayraksan and Morton 2006).

3.2 Enumerating Candidate Solutions

Enumerate in **BEST** requires that we identify all first-stage solutions $\hat{\mathbf{x}}$ to SMIP that satisfy $h'(\hat{\mathbf{x}}) \leq z''$. Given binary $\mathbf{x}$, the following procedure, which can be implemented easily in an algebraic model system like GAMS (Brooke et al. 1992), will accomplish the task.

Procedure Enumerate
Input: Deterministic or probabilistic global upper bound z'' for SMIP; data to define the lower-bounding function $h'(\mathbf{x})$;
Output: Candidate solution set $\mathcal{X} = \{\hat{\mathbf{x}} \in X | h'(\hat{\mathbf{x}}) \leq z''\}$;
{
  $\mathcal{X} \leftarrow \emptyset$;
  Repeat{
    Attempt to solve $\min_{\mathbf{x}\in X} h'(\mathbf{x})$ for $\hat{\mathbf{x}}$;
    If this is infeasible or $h'(\hat{\mathbf{x}}) > z''$, Return $\mathcal{X}$;
    $\mathcal{X} \leftarrow \mathcal{X} \cup \{\hat{\mathbf{x}}\}$;
    Add a constraint to the description of X to eliminate $\hat{\mathbf{x}}$ but only $\hat{\mathbf{x}}$, e.g.,

$$\sum_{i\in I|\hat{x}_i=1} x_i + \sum_{i\in I|\hat{x}_i=0} (1 - x_i) \leq |I| - 1\,; \tag{12}$$

  }
}

The problem $\min_{\mathbf{x}\in X} h'(\mathbf{x})$ is a sequentially restricted, deterministic, integer program or mixed-integer program. Clearly, **Enumerate** terminates finitely.

The reader will probably see that the minimizing problem in **Enumerate** need not be solved exactly. In fact, much computational effort can be saved by halting the optimization as soon as it finds any $\hat{\mathbf{x}} \in X$ satisfying $h'(\hat{\mathbf{x}}) \leq z''$. Computational efficiency may also improve if equation (12) can be replaced by a stronger constraint. For instance, if X enforces a simple cardinality requirement, $\mathbf{1}^T\mathbf{x} = b$, (12) can be replaced by

$$\hat{\mathbf{x}}^T\mathbf{x} \quad \leq \quad b - 1. \tag{13}$$

If any first-stage variable is a general, bounded integer, it can be replaced with the standard expansion in terms of binary variables (e.g., Owen and Mehrotra 2002), and the enumeration technique described above then applies. We mention a general, more computationally efficient procedure in section 5.

3.3 Simulating Candidate Solutions

To estimate the performance of solutions $\hat{\mathbf{x}} \in \mathcal{X}$ under uncertainty, we use common random numbers (CRNs) to simulate realizations of the random second-stage parameters, and then solve the resulting optimization models to collect optimal objective values for statistical analysis. CRNs result in greater efficiency for statistical comparisons when they induce positive correlation, and we expect such correlation for many SMIPs. For example, a pattern of generally high demands in SFLP is likely to result in high distribution costs and unmet demand penalties for any set of constructed facilities.

Procedure Simulate
Input: Data to define SMIP; candidate solution set $\mathcal{X}$ having confidence level $1 - \alpha_u$; initial sample size n_0.
Output: Randomly generated objective-function data for each $\hat{\mathbf{x}} \in \mathcal{X}$.
{
  According to the distribution of $\tilde{\boldsymbol{\xi}}$, generate n_0 random samples, $\hat{\boldsymbol{\xi}}_\ell$, $\ell = 1, \ldots, n_0$;
  Evaluate $z_{k\ell} \leftarrow h(\hat{\mathbf{x}}_k, \hat{\boldsymbol{\xi}}_\ell)$ for all $\hat{\mathbf{x}}_k \in \mathcal{X}$ and all ℓ;
}

3.4 Testing Candidate Solutions

There are two steps to the testing phase. First, a screening test, **Test1**, eliminates solutions from $\mathcal{X}$ that are unlikely to be optimal; we use bootstrapping here. The remaining candidate solutions, the selected subset $\mathcal{X}^* \subseteq \mathcal{X}$, will con-

tain an optimal solution with pre-specified confidence of at least $(1-\alpha_u)(1-\alpha_s)$. Typically, $|\mathcal{X}^*|$ is much smaller than $|\mathcal{X}|$, and the algorithm can terminate immediately if $|\mathcal{X}^*| = 1$. This type of screening procedure is known as *subset selection* (e.g., Bechhofer et al. 1995). The second phase, **Test2**, handles situations with $|\mathcal{X}^*| > 1$, and may be optional.

Procedure Test1 (Bootstrap Screen)
Input: Candidate solution set $\mathcal{X}$ created with confidence $1-\alpha_u$; objective-function samples $z_{k\ell}$ for all k, ℓ from **Simulate**; screening confidence parameter α_s; bootstrap sample size B.
Output: Selected subset $\mathcal{X}^* \subseteq \mathcal{X}$ which contains an optimal solution with approximate confidence level $(1-\alpha_u)(1-\alpha_s)$.
{

Initialize $w_k \leftarrow 0$ for $k = 1, \ldots, K \equiv |\mathcal{X}|$;

For each bootstrap replication $b = 1, \ldots, B$,

{

Generate n_0 indices from $\{1, 2, \ldots, n_0\}$ with replacement. Call this set L;

Compute average, optimal objective values

$$\bar{z}_k^{(b)} \leftarrow \frac{1}{n_0} \sum_{i \in L} z_{ki}, \quad k = 1, \ldots, K; \tag{14}$$

Update the tally for the "winning" solution:

$$k^* \leftarrow \underset{k\,=\,1,\ldots,K}{\operatorname{argmin}} \bar{z}_k^{(b)}; \quad w_{k^*} \leftarrow w_{k^*} + 1; \tag{15}$$

}

Let $[1], [2], \ldots, [K]$ denote indices of candidate solutions so that $w_{[1]} \geq w_{[2]} \geq \ldots \geq w_{[k]}$. Then

$$\mathcal{X}^* \leftarrow \left\{ \hat{\mathbf{x}}_{[1]}, \ldots, \hat{\mathbf{x}}_{[s]} \mid \frac{1}{n_0} \sum_{k=1}^{s} w_k \geq 1-\alpha_s, \; \frac{1}{n_0} \sum_{k=1}^{s-1} w_k < 1-\alpha_s \right\}; \tag{16}$$

Return $\mathcal{X}^*$;
}

This bootstrapping approach preserves the correlation induced by CRNs for greater efficiency. For $\hat{\mathbf{x}} \in \mathcal{X}$, **Test2** provides a direct estimate of the confidence level associated with declaring this solution to be "best." Asymptotically in n_0, w_k/B must converge to 0 for any non-optimal solution, to $1/m$ for any of m multiple optimal solutions, and hence to 1 for a unique optimal solution, if one exists. (The nature of SFLP makes a unique optimum highly likely.)

The selected subset $\mathcal{X}^*$ has random size that depends on the underlying distribution of objective values. If a few solutions dominate the others, then $E|\mathcal{X}^*|$ will be small even if $|\mathcal{X}|$ is large and we can use a modest value of n_0: As in other subset selection procedures, the most difficult situation from a screening perspective occurs when multiple optima exist. The size of $\mathcal{X}^*$ also depends on B, n_0, and the confidence level. $B \geq 1000$ is recommended for estimating percentiles. We do not yet know how to choose n_0 *a priori*, but we note that bootstrapping in other contexts can use sample sizes as small as nine (Efron and Tibshirani 1993). Higher confidence (i.e., lower α_s) increases the expected size of $\mathcal{X}^*$. For any particular problem, increasing or decreasing α_s over limited ranges may not alter the identification of solutions in $\mathcal{X}^*$. Thus, the procedure is conservative, i.e., actual confidence level is typically higher than the nominal one.

Additional sampling in the **Test2** phase may not be required, even if $|\mathcal{X}^*| > 1$. If estimated objective values for all $\hat{\mathbf{x}} \in \mathcal{X}^*$ are sufficiently close, a simple test may allow us to declare the apparent best solution to be ϵ-optimal with required confidence $1-\alpha_t$ (assuming that $\mathcal{X}^*$ contains an optimal solution to begin with). **Test2**—we borrow "Procedure $\mathcal{KN}$" from Kim and Nelson (2001)—uses the initial simulated samples to determine whether such a declaration can be made, or if additional sampling is required. In the latter case, the procedure goes on to specify a bound on the number of additional samples required so that, when the (possibly new) apparent best solution is identified, we can validly declare it to be ϵ-optimal with confidence $1-\alpha_t$.

Procedure Test2 (Select)
Input: Selected subset $\mathcal{X}^*$, $|\mathcal{X}^*| > 1$, having (approximate) confidence level $(1-\alpha_u)(1-\alpha_s)$; n_0 objective-function samples from **Simulate** for each $\hat{\mathbf{x}} \in \mathcal{X}^*$; testing confidence parameter α_t; optimality tolerance $\epsilon > 0$.
Output: The number of samples in addition to n_0 that must be applied to each $\hat{\mathbf{x}} \in \mathcal{X}^*$ to ensure that the apparent best solution is ϵ-optimal with (approximate) confidence $(1-\alpha_u)(1-\alpha_s)(1-\alpha_t)$.
{

For all $\hat{\mathbf{x}}_k, \hat{\mathbf{x}}_{k'} \in \mathcal{X}^*$, $k \neq k'$, compute sample variances for the pairwise difference using the initial n_0 samples:

$$S_{kk'}^2 \leftarrow \frac{1}{n_0 - 1} \sum_{\ell=1}^{n_0} \left(z_{k\ell} - z_{k'\ell} - (\bar{z}_{k\cdot} - \bar{z}_{k'\cdot}) \right)^2, \tag{17}$$

and set $S^2 \leftarrow \max_{k \neq k'} S_{kk'}^2$;

Compute

$$h^2 \leftarrow (n_0 - 1) \left[\left(\frac{2\alpha_t}{|\mathcal{X}^*| - 1} \right)^{-2/(n_0-1)} - 1 \right]; \tag{18}$$

Compute the maximum, total sample size required:

$$N \leftarrow \lfloor h^2 S^2/\epsilon^2 \rfloor; \tag{19}$$

Return $n^+ \leftarrow \max\{N - n_0, 0\}$;
}

The **Test2** procedure is less intuitive than the bootstrap in **Test1**, but in contrast to many alternatives, it requires no special tables or calculations. In general, N will be large if $|\mathcal{X}^*|$ is large, if the (positive) correlation induced by CRNs is low, or if ϵ is small. As in Kim and Nelson, we reuse the initial data, but note that the estimated savings in the total number of samples can be substantial even without this reuse.

4 COMPUTATIONAL RESULTS

To demonstrate **BEST**'s empirical performance, we randomly generate 15 SFLP test problems with different characteristics, solve them using **BEST**, and present results in Table 1. These problems vary by number of customer zones and number of potential facilities. Customers and potential facilities are randomly scattered across a rectangle with aspect ratio 1:3. Each shipping cost is proportional to the Euclidean distances a shipment must travel. Deterministic facility-construction costs, facility capacities, and penalties for unmet demand are provided as inputs, as are expected demands μ_j in each customer zone j. The actual demand in zone $j \in J$ is modeled as Unif$(\mu_j - \beta\mu_j, \mu_j + \beta\mu_j)$ for $\beta = 0.1, 0.2, 0.4$. We use $\alpha_u = \alpha_s = \alpha_t = .025$ for all problems, and desire a solution within 5% of the optimum. Upper bounds z'' are computed in **Bound** using $n_u = 100$ samples applied to a heuristic solution $\hat{\mathbf{x}}$ computed by solving (11) with $n' = 20$. We desire a relative optimality gap of 5%, so we set $\epsilon = 0.05z' \equiv 0.05 \min_{\mathbf{x}\in X} h'(\mathbf{x})$;

As anticipated, the results obtained using CRNs are highly correlated. The average pairwise correlations (not shown) range from a low of 0.8706 (for problem 13) to a high of 0.9995 (for problem 1).

The easiest problems are clearly those with smaller variability in the average demand. Indeed, **Bound** and **Enumerate** yield only a few candidate solutions when $\beta = 0.1$. But, even in instances with many candidate solutions, **Test1** eliminates all but a handful of these. Eight of the 15 problems are solved completely after **Test1**, and only one of the remaining problems has more than two candidate solutions. The *estimated maximum relative gap* for $\mathcal{X}$, denoted $\overline{\Delta}_{\max}$, is displayed in the table, and is defined as $\overline{\Delta}_{\max} = \max_{\mathbf{x}_k, \mathbf{x}_{k'} \in \mathcal{X}^*, k \neq k'} (\bar{z}_k - \bar{z}_{k'})/z'$. This may be viewed as a conservative point estimate of the true relative gap associated with the apparent best solution, and may therefore compared directly to the desired maximum gap of 0.05 (i.e., 5%).

Column 11 of Table 1 provides the upper bound, computed from (19), on the additional sampling required if the initial data are reused; only problem 15 requires further simulation and testing. (The value of n^+ will tend to increase as $|\mathcal{X}^*|$ increases, irrespective of S, and we note that the largest $|\mathcal{X}^*|$ occurs for problem 15. However, n^+ is positive here primarily because S is large.) Column 12 shows the total time required for the Bound, Enumerate, and initial Simulate (**BES**) steps, and Column 13 shows the total times required to run the bootstrap screening test **Test1**.

4.1 Improved Lower Bounds

A tighter lower-bounding function $h'(\mathbf{x})$ for **BEST** may lead to a reduced initial candidate set $\mathcal{X}$ and therefore reduced computational workload. The ideas of sequential approximation (SA) can help here. Applied to SFLP, SA would partition the state space of the random demand vector into Q regions $\mathcal{D}_q$, $q = 1, ..., Q$, compute conditional expectations for each element, $\bar{\mathbf{d}}_q = E[\tilde{\mathbf{d}}|\tilde{\mathbf{d}} \in \mathcal{D}_q]$, and solve the lower-bounding problem

$$\min_{\mathbf{x}\in X} h'_Q(\mathbf{x}) \quad = \quad \min_{\mathbf{x}\in X} \sum_{q=1}^{Q} P\{\tilde{\mathbf{d}} \in \mathcal{D}_q\} h(\mathbf{x}, \bar{\mathbf{d}}_q). \tag{20}$$

The value of $h'_Q(\mathbf{x})$ will approach $h(\mathbf{x})$ from below if the partition is refined and enlarged appropriately, i.e., as Q increases. These optimization problems resemble Q-scenario stochastic programs, which, of course, become more difficult to solve as Q increases.

BEST does not need an asymptotically convergent lower bound, but a tighter one may be useful. We find that substantially tighter bounds can accrue at modest computational cost by computing and exploiting a simple partition based on quartiles of total demand. In particular, for $0 = D_0 < D_1 < D_2 < D_3 < D_4 = \infty$, we define

$$\mathcal{D}_q \;=\; \{\tilde{\mathbf{d}} | D_{q-1} < \sum_{j\in J} \tilde{d}_j \leq D_q\},\; q = 1, \ldots, 4, \tag{21}$$

and then compute D_1, D_2 and D_3 so that $P\{\tilde{\mathbf{d}} \in \mathcal{D}_q\} = 0.25$ for $q = 1, \ldots, 4$. Defining $\bar{\mathbf{d}}_q$, $q = 1, \ldots, 4$, as indicated above, and solving (20) gives the new lower-bounding function.

We will not present detailed results, but summarize the effect of using the improved bound on the most difficult problems, problems 6, 9, 12 and 15: Computational workload for the improved bound increases by at most a factor of two, but $|\mathcal{X}|$ is reduced by at least an order of magnitude,

Table 1: Computational Results for Stochastic Facility Location Problems

Problem	Sites $\|I\|$	Zones $\|J\|$	Max Facil. b	β	Bounds z'	Bounds z''	$\|\mathcal{X}\|$	$\|\mathcal{X}^*\|$	$\overline{\Delta}_{\max}$	n^+	Computing Times† (secs) **BES**	**Test1**
1	10	20	5	.1	784.7	801.9	2	1	0	0	15.6	1.3
2				.2		840.0	5	1	0	0	17.4	4.5
3				.4		939.7	28	1	0	0	29.7	30.5
4	16	40	8	.1	873.4	878.2	2	1	0	0	192.5	1.3
5				.2		916.7	13	2	0.014	0	111.3	13.2
6				.4		1029.0	173	2	0.028	0	423.6	200.0
7	16	40	9	.1	834.3	856.0	8	2	0.007	0	85.3	7.7
8				.2		885.1	13	1	0	0	78.9	13.3
9				.4		1016.8	150	2	0.015	0	324.2	168.5
10	18	30	10	.1	518.2	523.1	2	1	0	0	22.1	1.3
11				.2		563.4	16	1	0	0	37.3	16.5
12				.4		687.8	566	2	0.020	0	2296.4	761.7
13	20	50	10	.1	958.8	966.2	3	2	0.001	0	132.8	1.1
14				.2		1006.3	72	1	0	0	255.3	74.7
15				.4		1155.1	1201	5	0.021	127	27458.1	1673.3

† **BES**, i.e., Bound, Enumerate and Simulate, from a 1GHz Pentium III computer operating under Windows 2000, **Test1** from a 1.5GHz Apple PowerPC G4 operating under Mac OSX.

so the computational time for the "**BES** steps" of **BEST** on each of these problems reduces to at most 20% of the original.

4.2 Reduced Simulation Sampling

The results in Table 1 have been produced using an arbitrarily chosen $n_0 = 50$ samples. But, bootstrap applications often involve only 10-20 samples (Efron and Tibshirani 1993). Accordingly, we now explore **BEST**'s behavior for $n_0 = 20$ by rerunning **Test1** using the first 20 samples produced by **Simulate** for each $\hat{\mathbf{x}} \in \mathcal{X}$. Problems 4, 8, and 15 each add one solution to the original set $\mathcal{X}^*$, problem 14 yields two, and no changes appear for the others. **Test2** then requires additional sampling for five problems, but the total number required is less than 45% of the original in all cases. The results suggest that the total number of simulated samples could be reduced by over 50% while sacrificing little in accuracy.

4.3 Simplified Testing

A few alternatives exist do exist for **Test1** and/or **Test2**. For example, Nelson and Matejcik (1995) describe a screening procedure that uses CRNs, which could be used in lieu of **Test1**; see also Nelson et al. 2001. However, we would prefer to replace **Test2** with a simpler bootstrap procedure, or combine **Test1** and **Test2** into a single, simple bootstrap procedure.

5 CONCLUSIONS AND FUTURE WORK

We have presented a new method for solving two-stage stochastic mixed-integer programs (SMIPs); first-stage variables must be discrete, but no conditions are placed on second-stage variables. The **BEST** algorithm (Bound, Enumerate, Simulate and Test) first uses bounding information to enumerate a candidate set of (first-stage) solutions that contains an optimal solution with high confidence. It then simulates the behavior of each candidate solution by sampling random second-stage parameters and solving the resulting, simple, deterministic problems. Next, it uses statistical tests—we apply bootstrapping—to screen out solutions that are unlikely to be optimal. If the screened candidate set contains a single element, the algorithm terminates. Otherwise, additional sampling and testing may be applied to select a single solution that is ϵ-optimal with high confidence.

Much work remains to make **BEST** better. We currently enumerate candidate solutions by solving a sequence of increasingly restricted MIPs. A procedure much like branch and bound would perform this more efficiently. The need for a deterministic lower-bounding function in the enumeration step limits **BEST**'s applicability, so a probabilistic alternative could prove useful. The bootstrapping screening test proves highly effective and is simple to implement. But, if that screening does not yield a single candidate solution, our follow-on test is more complicated: Future research will investigate simpler alternatives.

REFERENCES

Ahmed, S. 2004. Tutorials on stochastic programming and applications. Tenth International Conference on Stochastic Programming, University of Arizona, Tucson, AZ, 11-15 October.

Ahmed, S., A. J. King, and G. Parija. 2000. Capacity planning of web-server farms. INFORMS National Meeting, San Antonio, TX, 5-8 November.

Bayraksan, G., and D. P. Morton. 2007. Assessing solution quality in stochastic programs. *Mathematical Programming* 108 (2-3): 495–514.

Bechhofer, R. E., T. J. Santner, and D. Goldsman. 1995. *Design and Analysis of Experiments for Statistical Selection, Screening and Multiple Comparisons*. New York: John Wiley.

Birge, J. R., and F. Louveaux. 1997. *Introduction to Stochastic Programming*. New York: Springer-Verlag.

Brooke, A., D. Kendrick, and A. Meeraus. 1992. *GAMS, A User's Guide, Release 2.25*. South San Francisco, California: The Scientific Press.

Cormican, K. J., D. P. Morton, and R. K. Wood. 1998. Stochastic network interdiction. *Operations Research* 46 (2): 184–197.

Edmundson, H. P. 1956. Bounds on the expectation of a convex function of a random variable. P-982, The RAND Corporation, Santa Monica, CA.

Efron, B., and R. J. Tibshirani. 1993. *An Introduction to the Bootstrap*. New York: Chapman and Hall.

Frantzeskakis, L. and W. B. Powell. 1990. A successive linear approximation procedure for stochastic, dynamic vehicle allocation problems. *Transportation Science* 24 (1): 40–57.

Israeli, E. 1999. System interdiction and defense. Doctoral dissertation, Operations Research Department, Naval Postgraduate School, Monterey, California.

Kall, P., A. Ruszczyński, and K. Frauendorfer. 1988. Approximation techniques in stochastic programming. In *Numerical Techniques for Stochastic Optimization*, ed. Y. Ermoliev and R. J.-B. Wets, 33–64. Berlin: Springer-Verlag.

Kim, S.-H., and B. L. Nelson. 2001. A fully sequential procedure for indifference-zone selection in simulation. *ACM Transactions on Modeling and Computer Simulation*, 11(3): 251–273.

Klein Haneveld, W. K. and M. H. van der Vlerk. 1998. Stochastic integer programming: state of the art. Available via <http://citeseer.ist.psu.edu/kleinhaneveld98stochastic.html> [accessed 16 June 2006].

Kleywegt, A. J., A. Shapiro, A., and T. Homem-de-Mello. 2001. The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization* 12 (2): 479–502.

Laporte, G., F. V. Louveaux, and L. van Hamme. 1994. Exact solution of a stochastic location problem by an integer L-shaped algorithm. *Transportation Science* 28 (2): 95–103.

Madansky, A. 1959. Bounds on the expectation of a convex function of a multivariate random variable. *Annals of Mathematical Statistics* 30 (3): 743–746.

Mak, W-K., D. P. Morton, and R. K. Wood. 1999. Monte Carlo bounding techniques for verifying solution quality in stochastic programs. *Operations Research Letters* 24 (1-2): 47–56.

Morton, D. P., and A. S. Kenyon. 2003. Stochastic vehicle routing with random travel times. *Transportation Science* 37 (1): 69–82.

Morton, D. P., and R. K. Wood. 1999. Restricted-recourse bounds for stochastic linear programming. *Operations Research* 47 (6): 943–956.

Nelson, B. L., and F. J. Matejcik. 1995. Using common random numbers for indifference-zone selection and multiple comparisons in simulation. *Management Science* 41 (12): 1935–1945.

Nelson, B. L., J. Swann, D. Goldsman, and W. Song. 2001. Simple procedures for selecting the best simulated system when the number of alternatives is large. *Operations Research* 49 (6): 950–963.

Owen, J. H., and S. Mehrotra. 2002. On the value of binary expansions for general mixed-integer linear programs. *Operations Research* 50 (5): 810–819.

Stafford, M. 1997. A product-mix capacity planning model. Technical Report ORP97-03, Graduate Program in Operations Research, The University of Texas at Austin.

Rockafellar, R. T., and Wets, R. J.-B. 1976. Stochastic convex programming: relatively complete recourse and induced feasibility. *SIAM Journal on Control and Optimization* 14 (3): 574–589.

AUTHOR BIOGRAPHIES

SUSAN M. SANCHEZ is Professor of Operations Research at the Naval Postgraduate School, where she holds a joint appointment in the Graduate School of Business and Public Policy. Her research interests include experimental design for simulation studies, data-intensive statistics, and robust selection. She has a Ph.D. in Operations Research from Cornell University. She is currently the Simulation Area Editor for the *INFORMS Journal on Computing* and the ASA representative to the WSC Board of Directors. Her e-mail and web addresses are <ssanchez@nps.edu> and <http://www.nps.navy.mil/orfacpag/resumePages/sanchs.htm>, respectively.

R. KEVIN WOOD is Professor of Operations Research at the Naval Postgraduate School. His research interests include integer, stochastic, bilevel and trilevel program-

ming, with military and industrial applications. He is a member of INFORMS. His e-mail and web addresses are <kwood@nps.edu> and <http://www.nps.navy.mil/orfacpag/resumePages/woodrk.htm>, respectively.