ARTICLES

STOCHASTIC NETWORK INTERDICTION

KELLY J. CORMICAN

Naval Postgraduate School, Monterey, California

DAVID P. MORTON

The University of Texas at Austin, Austin, Texas

R. KEVIN WOOD

Naval Postgraduate School, Monterey, California

(Received January 1996; revisions received July 1996, October 1996; accepted December 1996)

Using limited assets, an interdictor attempts to destroy parts of a capacitated network through which an adversary will subsequently maximize flow. We formulate and solve a stochastic version of the interdictor's problem: Minimize the expected maximum flow through the network when interdiction successes are binary random variables. Extensions are made to handle uncertain arc capacities and other realistic variations. These two-stage stochastic integer programs have applications to interdicting illegal drugs and to reducing the effectiveness of a military force moving materiel, troops, information, etc., through a network in wartime. Two equivalent model formulations allow Jensen's inequality to be used to compute both lower and upper bounds on the objective, and these bounds are improved within a sequential approximation algorithm. Successful computational results are reported on networks with over 100 nodes, 80 interdictable arcs, and 180 total arcs.

This paper investigates stochastic variants of a network interdiction problem where an "interdictor," using limited assets, "interdicts" (destroys or at least stops the use of) parts of a capacitated network through which an adversary will subsequently maximize flow. The deterministic problem is to minimize the maximum achievable flow through the network subject to constraints on interdiction resources. In the stochastic variants, interdiction successes can be uncertain and/or arc capacities can be random variables. The objective of these stochastic network interdiction problems is to minimize the expected maximum flow through the network by selecting the best set of arcs to interdict, or to attempt to interdict.

The deterministic network interdiction problem has been studied in McMasters and Mustin (1970), Steinrauf (1991), Phillips (1992), and Wood (1993), with military applications and with applications to the interdiction of illegal drugs and precursor chemicals. Game-theoretic network interdiction models have also been studied (Wollmer 1964, Washburn and Wood 1995), but these models are substantially different: they determine optimal arc-inspection strategies for detecting an evader moving through a network surreptitiously.

Our analysis of stochastic network interdiction begins with this basic problem: We wish to interdict arcs in a network so as to minimize the expected maximum flow from a source node s to a sink node t. Assets available to perform these interdictions are limited. Interdiction successes are assumed to be independent binary random variables such that an unsuccessful interdiction attempt on an arc leaves the arc with its nominal capacity, and a successful interdiction leaves the arc with no capacity.

A stochastic maximum flow problem cannot be solved by solving a deterministic maximum flow problem where random arc capacities are replaced by their expected values. Likewise, our interdiction problem cannot be solved via the deterministic "expected-value problem" (Birge 1982) that minimizes maximum flow assuming uninterdicted arcs have nominal capacities and interdicted arcs have their expected residual (after-interdiction) capacities. Consider the simple interdiction problem represented in Figure 1, where each arc is marked with its nominal capacity. Assume that the probability of a successful interdiction on any arc is 0.6, at most one interdiction on each arc may be attempted, and resources limit the number of interdictions to two. The expected-value problem selects arcs (s, t) and $(s, 2)$, or arcs (s, t) and $(2, t)$ for interdiction. The capacity of the network based on expected capacities following interdiction is $(1 - 0.6)10 + (1 - 0.6)100 = 44$. This capacity would be $(1)10 + (1 - 0.6)100 = 50$ if arcs $(s, 2)$ and $(2, t)$ were interdicted, since nothing is gained in a deterministic model by interdicting identical arcs in series. The

Subject classifications: Military, targeting: network interdiction. Programming, stochastic: sequential approximation. Networks graphs, stochastic: maximum flow.
Area of review: PUBLIC SERVICES AND MILITARY APPLICATIONS.

0030-364X/98/4602-0184 $05.00

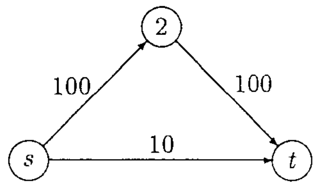

Figure 1. Interdiction decisions based on expected residual arc capacities may be incorrect.

correct solution is to interdict arcs (s, 2) and (2, t) leaving the stochastic network with an expected capacity (and expected maximum flow) of $(1)10 + (1 - 0.6)^2100 = 26$. It is clear that the expected-value model above will always interdict arcs that are part of some cut, say C (other irrelevant arcs may be interdicted if the interdiction budget is sufficient), yet the correct solution may be to interdict arcs that cannot all belong to the same cut C.

We first formulate the stochastic network interdiction problem (SNIP), as in the above example, with uncertain, binary interdiction successes, "SNIP(IB)." This problem is a two-stage stochastic program with recourse, but it has an unusual "min-max" objective. The integer first-stage decision variables represent the interdictor's choice of which arcs to attempt to destroy. After this decision has been implemented, a random subset of the interdiction attempts is successful and completely destroy the corresponding arcs, i.e., drop their capacities to zero. Unsuccessful interdiction attempts have no effect on the network. With complete knowledge regarding the status of the residual network, the adversary solves the continuous recourse problem of maximizing flow through this network. Our goal, i.e., the interdictor's goal, is to select arcs to attempt to interdict so as to minimize the expected value of the maximum flow that the adversary can achieve.

We also develop several variants of SNIP(IB):

(a) SNIP(ICB) has uncertain, binary interdiction successes and uncertain, binary arc capacities (the capacity of an arc can take on one of two nonnegative values); (b) SNIP(CB) has uncertain, binary arc capacities; (c) SNIP(CD) has uncertain arc capacities that can take on a finite number of nonnegative values; and (d) SNIP(IM) allows multiple uncertain interdictions on an arc. These are realistic variants of the initial model. For instance, SNIP(CB) could be used when the very existence of an arc in the adversary's network is uncertain because of incomplete intelligence.

A large-scale deterministic equivalent binary integer program may be formed for SNIP(IB) by (a) reformulating the problem as a simple minimization problem involving binary interdiction variables and binary second-stage variables (these are related to dual variables of a maximum flow problem), and (b) enumerating all possible realizations of the network with respect to the success or failure of each possible interdiction. Analogous models can be built for the other SNIPs. Unfortunately, solving such models would be computationally impractical for all but the smallest problems.

The stochastic programming literature suggests that sequential approximation techniques (e.g., Kall et al. 1988 and Frauendorfer 1992) coupled with decomposition algorithms (Van Slyke and Wets 1969) are practical methods for solving difficult stochastic programming problems such as the SNIP variants; we use this approach. A sequential approximation algorithm successively refines lower and upper bounds on the optimal objective value as it divides the state space of the random variables into finer and finer partitions. For solving a minimization problem, such an algorithm typically applies Jensen's inequality to obtain lower bounds and applies the Edmundson-Madansky inequality (Madansky 1959) or some other technique, for fixed values of the first-stage variables, to obtain upper bounds. The lower bound is usually easy to compute and the upper bound difficult.

The stochastic network interdiction problem is a minimization with respect to the first-stage variables and has random parameters appearing in the right-hand side of the second stage. Typically, Jensen's inequality is used to compute a lower bound in such situations, but because our second-stage problem is a maximization, it is the upper bound that is easy to compute via Jensen's inequality. We are able to use this bound effectively, so the key to solving our problem is to develop an efficient lower-bounding method. We do this by reformulating the model to move the random variables into the objective of the second stage and by applying Jensen's inequality again.

One might attempt to modify standard upper-bounding techniques (for minimization problems) to obtain a lower bound; e.g., the Edmundson-Madansky bound (Madansky 1959), a piecewise-linear bound (Birge and Wets 1989, Wallace 1987b, Birge and Wallace 1988), and a bound for problems with "convex marginal return functions" (Donohue and Birge 1995a, 1995b). For binary interdiction successes, the Edmundson-Madansky bound is equivalent to enumerating the state space of the random variables, which is impractical. The Donohue-Birge approach reduces computational effort relative to the Edmundson-Madansky bound, but can still require work that is exponential in the size of the network. The outer minimization of our min-max problems would be difficult to carry through to those bounds (for reasons of convexity discussed below), but would probably be impossible to carry through for a piecewise-linear bound. One might also attempt to adapt a specialized lower bound for the stochastic maximum flow problem, in particular, the bound of Aneja and Nair (1980). This bound involves products of arc reliabilities and these lead to an undesirable, nonlinear bounding model for interdiction. Our reformulation technique avoids all of the above difficulties.

As we show, the recourse function to the interdiction problem, in its "natural formulation," is a concave function of the interdiction decisions, when these first-stage variables are viewed as continuous. Since we want to minimize the expected recourse function, this precludes application of the "standard" integer version of the L-shaped method (Van Slyke and Wets 1969). However, we derive an equivalent formulation of the recourse problem that, for given interdiction successes, is a convex function of the first-stage decision variables and, for fixed first-stage decisions, is a convex function of interdiction success parameters. The second type of convexity allows us to apply Jensen's inequality to obtain a lower bound for fixed first-stage decisions, and the first type of convexity allows us to obtain a global lower bound by minimizing a convex function using the L-shaped method. Our method is efficient because we compute both lower and upper bounds through Jensen's inequality, and this computation does not depend on the number of stochastic parameters.

Our algorithm, applied to SNIP(IB), uses a state-space partitioning scheme that recursively conditions on whether or not a specific interdiction would be successful if attempted. Partitioning is similar for the other variants. For instance, in SNIP(CB), we can sequentially refine the partition by conditioning on whether or not a particular arc's capacity exists or not. The gap between upper and lower bounds on the optimal objective value is continually tightened with iteratively refined partitions of the state space.

The models we develop are all meant for interdicting arcs in a directed network but, in some networks such as road and communications networks, undirected arcs and node interdiction could be important. Our basic models can be easily modified to deal with these variations. The standard transformation of an undirected arc into two directed arcs in anti-parallel (e.g., Ahuja et al. 1993, p. 38) will convert undirected arcs appropriately. Only slight modifications of our models would be needed to accommodate the resulting dependent pairs of directed arcs. Another standard transformation converts nodes into directed arcs (e.g., Ahuja et al., p. 41–43).

Our techniques have applications to areas beyond network interdiction. When no arcs can be interdicted or sufficient resources exist to interdict every arc, SNIP(ICB) simplifies to a stochastic maximum flow problem (e.g., Carey and Hendrickson 1984, Evans 1976); see Section 5. If we convert the objective of SNIP(CB) from "min-max" to "max-max" and reformulate interdiction as capacity expansion, the resulting problem is a two-stage stochastic program where expected maximum flow can be improved by making investments in network infrastructure. This problem has been studied by Wallace (1987a), but our bounds would allow an alternate solution approach. The "vulnerability" of networks to attack or arc failure has been the topic of a number of papers that are largely theoretical in nature (e.g., Caccetta 1984, Goddard 1994). Our models or variants of them could be used, at least in an ad hoc fashion, for studying vulnerability from a more practical point of view. Finally, we note that our bounding techniques can be generalized for use in other stochastic programming problems; see Section 6.

Network concepts and notation are defined in the next section and a key theorem is provided that will lead to two equivalent formulations of SNIP(IB). Section 2 describes the deterministic network interdiction problem and the stochastic variant that is SNIP(IB). It also derives objective function bounds for use in the algorithm that is then described in Section 3. Section 4 describes extensions of SNIP(IB) that allow uncertain arc capacities, certain combinations of random capacities and interdiction attempts, and multiple interdiction attempts. Section 5 provides computational results for SNIP(IB), SNIP(ICB) and for a special case of SNIP(ICB) when no arcs may be interdicted—this is the classic stochastic maximum flow problem in a network with unreliable arcs. The paper is summarized in Section 6.

1. PRELIMINARIES

Let $G = (N, A)$ denote a directed network with node set N and arc set A. An *arc* is an ordered of pair of nodes (i, j) where i is the *tail* of the arc and j is the *head* of the arc. The set of arcs directed out of a node i, i.e., with tail i, is the *forward star of* i, denoted $\mathrm{FS}(i)$. The set of arcs directed into node i, i.e., with head i, is the *reverse star of* i, denoted $\mathrm{RS}(i)$.

A single commodity flows through the arcs of G from a *source node* s to a *sink node* t. The flow on any arc cannot exceed its *capacity*, u_{ij}. Let (t, s) be an artificial *return arc* associated with G and define $A' = A \cup \{(t, s)\}$, $u_{ts} = \sum_{(i,j)\in A} u_{ij} + 1$, and redefine $\mathrm{FS}(\ )$ and $\mathrm{RS}(\ )$ to include the return arc. The problem of finding the maximum feasible flow from s to t in G, i.e., the *maximum flow problem* on G, is given by the following linear program (LP) which maximizes the flow on the return arc subject to arc capacities and flow balance constraints:

MAXFLOW

$$\max_{\mathbf{x}} x_{ts},$$

$$\text{s.t.} \sum_{(i,j)\in FS(i)} x_{ij} - \sum_{(j,i)\in RS(i)} x_{ji} = 0 \qquad \forall\, i \in N, \tag{1}$$

$$0 \leqslant x_{ij} \leqslant u_{ij} \qquad \forall\, (i, j) \in A'. \tag{2}$$

Note that u_{ts} is effectively infinite since the maximum flow cannot exceed $u_{ts} - 1$.

Lemma 1 regarding MAXFLOW follows from the observation that the maximum flow through a network can be increased by at most one unit by increasing the capacity of an arc by one unit.

Lemma 1. *The dual variables associated with arc capacities in MAXFLOW are bounded above by* 1.

In the sections that follow, we will use the following theorem to establish equivalence of alternative formulations of deterministic and stochastic network interdiction models. The set of arcs A^* will later correspond to successfully interdicted arcs.

Theorem 2. *Let $A^* \subseteq A'$. Then, $z^* = z^{**}$ where*:

$$
\begin{aligned}
z^* = \max_{\mathbf{x}} & & & \textit{Dual Vars.}\\
\text{s.t.} \sum_{(i,j)\in FS(i)} x_{ij} - \sum_{(j,i)\in RS(i)} x_{ji} &= 0 \quad \forall\, i \in N & & :\pi_i\\
x_{ij} &\leqslant u_{ij} \quad \forall\,(i,j)\in A' & & :\alpha_{ij}\\
x_{ij} &\leq 0 \quad \forall\,(i,j)\in A^* & &\\
x_{ij} &\geq 0 \quad \forall\,(i,j)\in A' & & :M_{ij}
\end{aligned}
\tag{3}
$$

and:

$$
\begin{aligned}
z^{**} = \max_{\mathbf{x}} x_{ts} - \sum_{(i,j)\in A^*} x_{ij} & & & \textit{Dual Vars.}\\
\text{s.t.} \sum_{(i,j)\in FS(i)} x_{ij} - \sum_{(j,i)\in RS(i)} x_{ji} &= 0 \quad \forall\, i \in N & & :\pi_i\\
x_{ij} &\leqslant u_{ij} \quad \forall\,(i,j)\in A' & & :\alpha_{ij}\\
x_{ij} &\geqslant 0 \quad \forall\,(i,j)\in A'. & &
\end{aligned}
\tag{4}
$$

Proof. Let [$\mathbf{x}^*$, ($\boldsymbol{\pi}^*$, $\boldsymbol{\alpha}^*$, $\boldsymbol{\mu}^*$)] denote an optimal primal and dual solution to (3) with dual variables corresponding to constraints as noted. It suffices to show that [$\mathbf{x}^{**}$, ($\boldsymbol{\pi}^{**}$, $\boldsymbol{\alpha}^{**}$)] $\equiv$ [$\mathbf{x}^*$, ($\boldsymbol{\pi}^*$, $\boldsymbol{\alpha}^*$)] is an optimal solution to (4) with objective value z^*. That the primal and dual objective values of (4) are z^*, when evaluated at $\mathbf{x}^{**}$ and ($\boldsymbol{\pi}^{**}$, $\boldsymbol{\alpha}^{**}$), can be seen by simple substitution. The solution $\mathbf{x}^{**} = \mathbf{x}^*$ is primal feasible to (4) since the primal constraints of (3) are a restriction of those of (4). Using the theorem of strong duality, the proof will be complete if we can show dual feasibility of ($\boldsymbol{\pi}^{**}$, $\boldsymbol{\alpha}^{**}$) in (4). The dual constraints associated with $(i, j) \in A' \setminus A^*$ in (4) are satisfied by ($\boldsymbol{\pi}^{**}$, $\boldsymbol{\alpha}^{**}$) since they are identical to those in (3). The remaining dual constraints of (4), which are $\pi_i - \pi_j + \alpha_{ij} \geqslant -1$ for $(i, j) \in A^*$, are satisfied since:

$$\pi_i^* - \pi_j^* + \alpha_{ij}^* + \mu_{ij}^* \geqslant 0 \quad \forall\,(i,j)\in A^*$$

by dual feasibility of (1),

$$\text{f}\ \pi_i^* - \pi_j^* + \alpha_{ij}^* \geqslant -\mu_{ij}^* \geqslant -1 \quad \forall\,(i,j)\in A^*$$

by rearranging and Lemma 1,

$$\text{f}\ \pi_i^{**} - \pi_j^{**} + \alpha_{ij}^{**} \geqslant -1 \quad \forall\,(i,j)\in A^*$$

by substitution. □

2. THE DETERMINISTIC AND STOCHASTIC MODELS

This section reviews the basic deterministic network interdiction model and then describes a stochastic variant, SNIP(IB), where interdiction successes are binary random variables. Lower and upper bounds on the optimal objective value of the stochastic model are developed in subsection 2.3 for use in the sequential approximation algorithm of Section 3.

2.1. The Deterministic Model

First consider the basic, deterministic network interdiction model from Wood (1993). This model assumes that an interdictor has R total units of resource that can be expended to interdict arcs and that r_{ij} units of resource are required to interdict arc (i, j). Let γ_{ij} be a binary decision variable that is 1 if arc (i, j) is interdicted and is 0 otherwise. For notational simplicity, let γ_{ts} exist, but define $r_{ts} = R + 1$ so that the return arc will never be interdicted. Also, let $\Gamma \equiv \{\boldsymbol{\gamma} | \Sigma_{(i,j)\in A'} r_{ij}\gamma_{ij} \leqslant R,\ \gamma_{ij} \in \{0, 1\}\ \forall\ (i, j) \in A'\}$ and let x $\in X$ denote the flow-balance and capacity constraints of MAXFLOW, i.e., constraints (1) and (2). The deterministic network interdiction problem can now be described by the following min-max model:

D-MIN-MAX-1 $\qquad z^* = \min_{\boldsymbol{\gamma}\in\Gamma} h_d(\boldsymbol{\gamma})$ (Model 1)

where:

$$
\begin{aligned}
h_d(\boldsymbol{\gamma}) = \max_{\mathbf{x}\in X}\ & x_{ts},\\
\text{s.t. } & 0 \leqslant x_{ij} \leqslant u_{ij}(1-\gamma_{ij}) \quad \forall\,(i,j)\in A'.
\end{aligned}
$$

The exact form of Γ is unimportant provided that its constraints are linear. However, even when $r_{ij} = 1$ for all $(i, j) \in A$, D-MIN-MAX-1 is an NP-complete problem (Wood 1993).

We may reformulate the deterministic network interdiction problem as:

D-MIN-MAX-2 $\qquad z^* = \min_{\boldsymbol{\gamma}\in\Gamma} g_d(\boldsymbol{\gamma})$ (Model 2)

where:

$$g_d(\boldsymbol{\gamma}) = \max_{\mathbf{x}\in X} x_{ts} - \sum_{(i,j)\in A'} \gamma_{ij}x_{ij}.$$

Applying Theorem 2 with $A^* = \{(i, j) | \gamma_{ij} = 1\}$, we see that $g_d(\boldsymbol{\gamma}) = h_d(\boldsymbol{\gamma})$ for all $\boldsymbol{\gamma} \in \{0, 1\}^{|A'|}$. Hence, this reformulation is equivalent to D-MIN-MAX-1 from the perspective of determining optimal interdiction decisions.

2.2. A Stochastic Model: Uncertain Interdiction

Interdiction of an arc need not be successful; it may be completely unsuccessful, partially successful (part of the arc's capacity is destroyed) or completely successful. Initially, we consider the binary case where an attempted interdiction of arc (i, j) is completely successful with probability p_{ij} and is completely unsuccessful with probability $1 - p_{ij}$. Independence of interdiction successes is assumed, and only a single interdiction may be attempted on any arc. This model is denoted "SNIP(IB)."

Let $\tilde{I}_{ij}$ be an indicator random variable that is 1 with probability p_{ij} and is 0 with probability $1 - p_{ij}$. The state space; i.e., support, for $\tilde{\mathbf{I}}$ is denoted $\mathcal{I} \subseteq \{0, 1\}^{|A'|}$. For the case of binary random interdiction successes, the following min-max model is analogous to D-MIN-MAX-1 (Model 1) and models SNIP(IB):

S-MIN-MAX-1(IB) $\qquad w^* = \min_{\boldsymbol{\gamma}\in\Gamma} Eh(\boldsymbol{\gamma}, \tilde{\mathbf{I}})$ (Model 3)

where:

$$
\begin{aligned}
h(\boldsymbol{\gamma}, \tilde{\mathbf{I}}) = \max_{\mathbf{x}\in X}\ & x_{ts},\\
\text{s.t. } & 0 \leqslant x_{ij} \leqslant u_{ij}(1-\tilde{I}_{ij}\gamma_{ij}) \quad \forall\,(i,j)\in A'.
\end{aligned}
$$

In a fashion that is analogous to reformulating D-MIN-MAX-1 (Model 1) as D-MIN-MAX-2 (Model 2), we reformulate S-MIN-MAX-1(IB) as follows:

$$\textbf{S-MIN-MAX-2(IB)} \qquad w^* = \min_{\boldsymbol{\gamma}\in\Gamma} Eg(\boldsymbol{\gamma}, \tilde{\mathbf{I}}) \qquad \text{(Model 4)}$$

where:

$$g(\boldsymbol{\gamma}, \tilde{\mathbf{I}}) = \max_{\mathbf{x}\in X} x_{ts} - \sum_{(i,j)\in A'} \tilde{I}_{ij}\gamma_{ij}x_{ij}.$$

It can be seen that $g(\hat{\boldsymbol{\gamma}}, \hat{\mathbf{I}}) = h(\hat{\boldsymbol{\gamma}}, \hat{\mathbf{I}})$ for all $(\hat{\boldsymbol{\gamma}}, \hat{\mathbf{I}}) \in \{0, 1\}^{|A'|} \times \mathcal{I}$ by applying Theorem 2 with $A^* = \{(i, j) \in A' | \hat{\gamma}_{ij} = 1, \hat{I}_{ij} = 1\}$. Thus, the *recourse functions* $h(\boldsymbol{\gamma}, \tilde{\mathbf{I}})$ and $g(\boldsymbol{\gamma}, \tilde{\mathbf{I}})$ are equivalent and this reformulation is equivalent to S-MIN-MAX-1(IB). Again, by "equivalent" we mean that the first-stage solution sets for S-MIN-MAX-1(IB) and S-MIN-MAX-2(IB) are identical; the second-stage solution sets may differ.

2.3. Lower and Upper Bounds for Uncertain Interdiction

The sequential approximation method developed in Section 3 requires lower and upper bounds on w^*, which we develop here. The following proposition is key to these bounds. The proposition follows from standard LP theory, but its application to stochastic programming is discussed in Wets (1966).

Proposition 3. *For fixed* $\boldsymbol{\gamma}$, $h(\boldsymbol{\gamma}, \tilde{I})$ *and* $g(\boldsymbol{\gamma}, \tilde{I})$ *are concave and convex functions on the convex hull of* $\mathcal{I}$, *respectively*.

As a result, we may apply Jensen's inequality, twice, with $\bar{I}_{ij} = E\tilde{I}_{ij}$ to obtain:

$$g(\boldsymbol{\gamma}, \bar{\mathbf{I}}) \leq Eg(\boldsymbol{\gamma}, \tilde{\mathbf{I}}) = Eh(\boldsymbol{\gamma}, \tilde{\mathbf{I}}) \leq h(\boldsymbol{\gamma}, \bar{\mathbf{I}}).$$

The following bounds on w^* may be formed by solving $\min_{\boldsymbol{\gamma}\in\Gamma} g(\boldsymbol{\gamma}, \bar{\mathbf{I}})$ to obtain $\hat{\gamma}$:

$$g(\hat{\boldsymbol{\gamma}}, \bar{\mathbf{I}}) \leq w^* \leq h(\hat{\boldsymbol{\gamma}}, \bar{\mathbf{I}}). \qquad (5)$$

In Section 3, we will refine the bounds (5) by extending them to a partition of the state space of the indicator random variables (Huang et al. 1977). Let $\mathcal{S} = \{\mathcal{I}^1, \mathcal{I}^2, \ldots, \mathcal{I}^K\}$ be such a partition of $\mathcal{I}$; i.e., $\cup_{k=1}^{K}\mathcal{I}^k = \mathcal{I}$ and $\mathcal{I}^j \cap \mathcal{I}^k = \emptyset$ for $j \neq k$. The elements of $\mathcal{S}$ are called *cells*. Also, let $\bar{\mathbf{I}}^k \equiv E(\tilde{\mathbf{I}} | \tilde{\mathbf{I}} \in \mathcal{I}^k)$ and $p^k \equiv Prob(\tilde{\mathbf{I}} \in \mathcal{I}^k)$. Then, a lower bound $\underline{w}(\mathcal{S})$ on w^*, with respect to the partition $\mathcal{S}$, is derived as follows:

$$\textbf{LBMIN}(\mathcal{S}) \qquad w^* = \min_{\boldsymbol{\gamma}\in\Gamma} \sum_{k=1}^{K} p^k E[g(\boldsymbol{\gamma}, \tilde{\mathbf{I}}) | \tilde{\mathbf{I}} \in \mathcal{I}^k]$$

$$\geq \min_{\boldsymbol{\gamma}\in\Gamma} \sum_{k=1}^{K} p^k g(\boldsymbol{\gamma}, \bar{\mathbf{I}}^k) \equiv \underline{w}(\mathcal{S}), \qquad \text{(Model 5)}$$

where "Model 5" refers to the minimization problem on the right. In Section 3, we will convert this min-max model to a simple minimization model and it will become apparent that the number of rows and columns in the problem is proportional to $|\mathcal{S}|$. An upper bound $\bar{w}(\mathcal{S}, \boldsymbol{\gamma})$ for fixed $\boldsymbol{\gamma}$ is obtained using:

$$\textbf{UB}(\mathcal{S}, \boldsymbol{\gamma}) \qquad w^* = \sum_{k=1}^{K} p^k E[h(\boldsymbol{\gamma}, \tilde{\mathbf{I}}) | \tilde{\mathbf{I}} \in \mathcal{I}^k]$$

$$\leq \sum_{k=1}^{K} p^k h(\boldsymbol{\gamma}, \bar{\mathbf{I}}^k) \equiv \bar{w}(\mathcal{S}, \boldsymbol{\gamma}), \qquad \text{(Model 6)}$$

where "Model 6" refers to the $|\mathcal{S}|$ recourse function evaluations (maximizations) required to calculate $\bar{w}(\mathcal{S}, \boldsymbol{\gamma})$. In the algorithm, we apply $\text{UB}(\mathcal{S}, \boldsymbol{\gamma})$ to the optimal decision $\hat{\boldsymbol{\gamma}}$ obtained in calculating LBMIN($\mathcal{S}$).

A *refinement* of a partition $\mathcal{S}$ is another partition $\mathcal{S}'$ such that for any $\mathcal{I}^{k'} \in \mathcal{S}'$, $\mathcal{I}^{k'} \subseteq \mathcal{I}^k$ for some $\mathcal{I}^k \in \mathcal{S}$ and $\mathcal{I}^{k'} \subset \mathcal{I}^k$ for at least one $\mathcal{I}^k \in \mathcal{S}$ and $\mathcal{I}^{k'} \in \mathcal{S}'$. LBMIN($\mathcal{S}$) and UB($\mathcal{S}$, $\boldsymbol{\gamma}$) are monotonic in the sense that if $\mathcal{S}'$ is a refinement of $\mathcal{S}$:

$$\underline{w}(\mathcal{S}) \leq \underline{w}(\mathcal{S}') \quad \text{and} \quad \bar{w}(\mathcal{S}, \boldsymbol{\gamma}) \geq \bar{w}(\mathcal{S}', \boldsymbol{\gamma}).$$

See Hausch and Ziemba (1983) for details.

The idea behind our sequential approximation algorithm is to create a sequence of finer and finer partitions $\mathcal{S}$ until the gap between lower and upper bounds is sufficiently small and we can declare the problem solved, at least approximately. We recursively refine $\mathcal{S}$ by selecting one of its cells $\mathcal{I}^k$ and *subdividing* this cell, i.e., partitioning $\mathcal{I}^k$, into two cells $\mathcal{I}^{k'}$ and $\mathcal{I}^{k''}$, and then replacing $\mathcal{I}^k$ with $\mathcal{I}^{k'}$ and $\mathcal{I}^{k''}$ to obtain the refinement $\mathcal{S}'$. The algorithm is convergent since our refinement scheme could, in principle, create a partition that enumerates $\mathcal{I}$, at which point LBMIN($\mathcal{S}$) (Model 5) would solve SNIP(IB) exactly. Such an algorithm is clearly of exponential complexity so, for the method to be useful, we must demonstrate empirically that $\mathcal{S}$ does not grow too large before the gap between the bounds shrinks sufficiently to yield a high-quality solution.

3. SEQUENTIAL APPROXIMATION ALGORITHM

In this section we describe a sequential approximation algorithm for solving SNIP(IB) that utilizes the bounds developed in the previous section. The algorithm is listed in Figure 2.

3.1. Overview of the Algorithm

At each iteration of the algorithm (starting at Step 1) the partition of $\mathcal{I}$ induces an approximating problem (Model 5) that is solved to calculate LBMIN($\mathcal{S}$). Model 5 is an approximation to the original problem in the following sense: it is equivalent to Model 4 except that $\tilde{\mathbf{I}}$ is replaced by a random vector with fewer realizations (scenarios), namely, $\bar{\mathbf{I}}^k$, $k = 1, \ldots, K$, that occur with probability p^k, $k = 1, \ldots, K$. This min-max K-scenario stochastic integer program (Model 5) can be solved exactly at Step 1 using the following simple minimization model:

$$\textbf{E-MIN-2(IB)}(\mathcal{S}) \qquad \underline{w}(\mathcal{S}) = \min_{\boldsymbol{\gamma},\boldsymbol{\pi},\boldsymbol{\alpha}} \sum_{k=1}^{K} \sum_{(i,j)\in A'} p^k \alpha_{ij}^k u_{ij}$$

$$\text{(Model 7)}$$

$$\text{s.t. } \pi_i^k - \pi_j^k + \alpha_{ij}^k + \bar{I}_{ij}^k \gamma_{ij} \geq e_{ij} \qquad \forall\, k,\ (i, j) \in A',$$

$$\alpha_{ij}^k \geq 0 \qquad \forall\, k,\ (i, j) \in A',\ \boldsymbol{\gamma} \in \Gamma,$$

Input: A problem instance for SNIP(IB) and finite, absolute convergence tolerance $\epsilon > 0$

Output: $\boldsymbol{\gamma}^*$ approximately solving SNIP(IB), and lower bound L^* and upper bound U^* on w^* such that $L^* \leqslant w^* \leqslant Eh(\boldsymbol{\gamma}^*, \tilde{\mathbf{I}}) \leqslant U^*$ and $U^* - L^* \leqslant \epsilon$;

STEP 0. Let $\mathcal{S} = \{\mathcal{I}\}$, $U^* = +\infty$ and $L^* = 0$;

STEP 1. Solve LBMIN($\mathcal{S}$) for $\hat{\boldsymbol{\gamma}}$ and let $L^* = \underline{w}(\mathcal{S})$;

STEP 2. If ($U^* - L^* \leqslant \epsilon$) then go to (7).

STEP 3. Evaluate UB($\mathcal{S}$, $\hat{\boldsymbol{\gamma}}$) and let $U' = \bar{w}(\mathcal{S}, \hat{\boldsymbol{\gamma}})$;

STEP 4. If ($U' < U^*$) then let $\boldsymbol{\gamma}^* = \hat{\boldsymbol{\gamma}}$ and let $U^* = U'$;

STEP 5. If ($U^* - L^* \leqslant \epsilon$) then go to (7);

STEP 6. Refine the partition $\mathcal{S}$ and go to (1);

STEP 7. Print ("Approximate solution is", $\boldsymbol{\gamma}^*$);

STEP 8. Print ("Lower bound =", L^*, "Upper bound =", U^*) and halt;

Figure 2. Sequential approximation algorithm for solving SNIP (IB).

where: $e_{ij} = 0$ for all $(i, j) \neq (t, s)$ and $e_{ts} = 1$. This model can be derived by setting up LBMIN($\mathcal{S}$), in the form analogous to S-MIN-MAX-2(IB) (Model 4), and taking the duals of the inner maximizations.

However, as the number of scenarios becomes large, solving E-MIN-2(IB)($\mathcal{S}$) directly can become difficult. Rather than doing this, we solve, or approximately solve, E-MIN-2(IB)($\mathcal{S}$) by Benders decomposition (the decomposition can also be derived directly from the min-max formulation of LBMIN($\mathcal{S}$)). The decomposition algorithm is identical to the continuous, L-shaped algorithm of Van Slyke and Wets (1969), except that the master problem is subject to both the linear and binary constraints of Γ. (See Laporte and Louveaux 1993 and Wollmer 1980 for more complete discussions on extensions of the L-shaped method to stochastic integer programming.)

One advantage of using the decomposition approach is that cuts from any iteration are valid, if not "tight," in subsequent iterations under finer partitions. Thus, the final master problem in iteration n of the algorithm may be used as the starting point for the master problem in iteration $n + 1$, and as n increases, the decompositions tend to converge more quickly than if one were "starting from scratch" in each iteration. Another advantage of the decomposition is that we need not solve E-MIN-2(IB)($\mathcal{S}$) exactly. At each iteration of the Benders decomposition, we obtain a lower bound L'' and an upper bound U'' such that $L'' \leqslant L^* = \underline{w}(\mathcal{S}) \leqslant U''$. Thus, we can stop the decomposition when L'' is "good enough" and use L'' as a global lower bound in place of L^*.

In Step 3, we evaluate $\bar{w}(\mathcal{S}, \boldsymbol{\gamma})$ (Model 6) at the optimal solution $\hat{\boldsymbol{\gamma}}$ of the lower-bounding problem and, if appropriate, update the best upper bound observed to date. However, it might be advantageous to improve the upper bound by minimizing $\bar{w}(\mathcal{S}, \boldsymbol{\gamma})$ subject to $\boldsymbol{\gamma} \in \Gamma$. We do not pursue this idea, but we note that it is possible to do so since this min-max problem can be converted to a minimization problem using the transformations exploited by Wood (1993) for the deterministic case.

We end this overview by noting that the absolute termination criterion for the algorithm is tested both in Steps 2 and 5. Of course, other criteria can be used and, in practice, we terminate on relative error; i.e., when (upper bound − lower bound)/(lower bound) is sufficiently small. The remainder of this section focuses on the approach we use to refine the partition in Step 6.

3.2. Partitioning

We refine the partition $\mathcal{S}$ by selecting a cell to subdivide and then subdividing that cell by conditioning on whether or not a potential interdiction is successful. Obviously, the algorithm slows down as the cardinality of the partition increases, so we must reduce the optimality gap to an acceptable level before $\mathrm{K} = |\mathcal{S}|$ becomes too large. Two heuristics are described here that tend to reduce the gap quickly by (a) selecting a "good" cell to subdivide and (b) subdividing the cell effectively. See Birge and Wets (1986), Frauendorfer (1992, § 19), and Kall et al. (1988) for related discussions.

3.2.1. Selecting a cell to subdivide. The gap between the upper and lower bounds may be expressed:

$$\bar{w}(\mathcal{S}, \hat{\boldsymbol{\gamma}}) - \underline{w}(\mathcal{S}) = \sum_{k=1}^{\mathrm{K}} p^k[h(\hat{\boldsymbol{\gamma}}, \bar{\mathbf{I}}^k) - g(\hat{\boldsymbol{\gamma}}, \bar{\mathbf{I}}^k)],$$

and hence, the difference due to each cell is:

$$D^k(\hat{\boldsymbol{\gamma}}) \equiv p^k[h(\hat{\boldsymbol{\gamma}}, \bar{\mathbf{I}}^k) - g(\hat{\boldsymbol{\gamma}}, \bar{\mathbf{I}}^k)]. \tag{6}$$

We will select a cell k' to subdivide such that $k' \in \operatorname{argmax}_k\{D^k(\hat{\boldsymbol{\gamma}})\}$, although other selection criteria could be used. It is important to note that (6) depends on $\hat{\boldsymbol{\gamma}}$, and $D^k(\ )$ should be re-evaluated for all cells when $\hat{\boldsymbol{\gamma}}$ has changed from the previous iteration. In subsection 5.2 we describe a computational enhancement that significantly increases solution speed by performing a sequence of cell refinements prior to re-solving the lower-bounding problem to obtain a new $\hat{\boldsymbol{\gamma}}$. Next, we need to decide how to partition cell k.

3.2.2. Subdividing on arcs. In rectangular partitioning schemes, cells are recursively subdivided by planes orthogonal to the axis of one stochastic parameter. In our case, this corresponds to selecting some cell $\mathcal{I}^k$ together with an arc (i', j') such that $\tilde{I}_{i'j'}$ is not fixed in $\mathcal{I}^k$, and conditioning on whether $\tilde{I}_{i'j'} = 0$ or $\tilde{I}_{i'j'} = 1$. A generic cell $\mathcal{I}^k$ of $\mathcal{S}$ has the form $\Pi_{(i,j)\in A'}\{I_{ij}|I_{ij} = a^k_{ij}, b^k_{ij}\}$ where the possible values for the pair (a^k_{ij}, b^k_{ij}) indicate whether $\tilde{I}_{ij}$ on $\mathcal{I}^k$ is: fixed at 0, (0, 0), fixed at 1, (1, 1), or not fixed, (0, 1). Subdivision of a cell $\mathcal{I}^k$ with respect to arc (i', j'), satisfying $(a^k_{i'j'}, b^k_{i'j'}) =$ (0, 1), corresponds to replacing $\mathcal{I}^k$ with two disjoint cells $\mathcal{I}^{k'}$ and $\mathcal{I}^{k''}$ satisfying:

$$(a_{ij}^{k'}, b_{ij}^{k'}) = (a_{ij}^{k''}, b_{ij}^{k''}) = (a_{ij}^{k}, b_{ij}^{k}) \qquad \forall\,(i,j) \in A' \setminus \{(i',j')\},$$

and

$$(a_{i'j'}^{k'}, b_{i'j'}^{k'}) = (0,0) \quad \text{and} \quad (a_{i'j'}^{k''}, b_{i'j'}^{k''}) = (1,1).$$

Conditional probabilities and expectations are easy to calculate:

$$p^k = \text{Prob}(\tilde{\mathbf{I}} \in \mathcal{I}^k) = \left(\prod_{\{(i,j)\in A' \mid a_{ij}^k = b_{ij}^k = 1\}} p_{ij}\right) \cdot \left(\prod_{\{(i,j)\in A' \mid a_{ij}^k = b_{ij}^k = 0\}} (1 - p_{ij})\right),$$

$$\bar{I}_{ij}^k = E(\tilde{\mathbf{I}} \mid \tilde{\mathbf{I}} \in \mathcal{I}^k) = \begin{cases} 0 & \text{if } (a_{ij}^k, b_{ij}^k) = (0,0), \\ 1 & \text{if } (a_{ij}^k, b_{ij}^k) = (1,1), \\ p_{ij} & \text{if } (a_{ij}^k, b_{ij}^k) = (0,1). \end{cases}$$

Our method for selecting an arc for conditioning is described next, assuming that cell $\mathcal{I}^k$ has already been selected for subdivision. For each arc (i, j) satisfying $(a_{ij}^k, b_{ij}^k) = (0, 1)$, we form $\mathcal{I}^{k'}$ and $\mathcal{I}^{k''}$ as described above and calculate the deviation in bounds $D^{k'}(\hat{\boldsymbol{\gamma}}) + \mathrm{D}^{k''}(\hat{\boldsymbol{\gamma}})$ (which is guaranteed to be no larger than $D^k(\hat{\boldsymbol{\gamma}})$). We select an arc (i', j') to subdivide that minimizes $D^{k'}(\hat{\boldsymbol{\gamma}}) + \mathrm{D}^{k''}(\hat{\boldsymbol{\gamma}})$. We must solve four recourse problems for each candidate arc on which we consider conditioning. Note that such a rule would be considerably more expensive to implement when using the Edmundson-Madansky bound.

In the SNIP(IB) model, we can reduce the set of arcs to which we apply the above rule by never subdividing on an arc (i, j) that currently has $\hat{\gamma}_{ij} = 0$. Clearly, the algorithm still converges under this rule because a stochastic parameter $\tilde{I}_{ij}$ cannot contribute to the error unless $\hat{\gamma}_{ij} = 1$.

Selecting an arc that minimizes $D^{k'}(\hat{\boldsymbol{\gamma}}) + \mathrm{D}^{k''}(\hat{\boldsymbol{\gamma}})$ puts equal value on decreasing the upper bound or increasing the lower bound. This rule can be modified by putting a greater weight on the improvement in the lower bound, with the idea that it is more important to get a high quality lower-bounding approximation problem since it determines the incumbent solution.

4. EXTENSIONS

This section first considers an extension of SNIP(IB), denoted “SNIP(ICB),” where arc capacities as well as interdiction successes are binary random variables. As a special case, we immediately obtain SNIP(CB), which only has uncertain binary capacities, by setting the probabilities of successful interdictions in SNIP(ICB) to 1. We then extend SNIP(CB) to SNIP(CD) where capacity can take on a finite number of realizations. Finally, we consider SNIP(IM) where multiple interdiction attempts on a single arc are allowed. Such a model would be useful when the probability of a successful interdiction is relatively low and/or when interdiction success on a particular arc or group of arcs is critical to reducing flow.

4.1. Uncertain Binary Interdictions and Arc Capacities, SNIP(ICB)

Consider the model SNIP(ICB) where (a) at most one interdiction on an arc may be attempted, (b) interdiction successes are binary random variables as in SNIP(IB), (c) the capacity of each arc (i, j) is a binary random variable that can take on the values 0 or u_{ij}, and (d) all random variables are independent. (Modifications for nonzero minimum capacities are easily handled.) Define the indicator random variable $\tilde{J}_{ij}$ to be 1 if arc (i, j) exists and has capacity u_{ij}, and to be 0 if it does not exist (or it exists but has no capacity). Also, let $\phi_{ij} \equiv \text{Prob}(\tilde{J}_{ij} = 1)$; definitions related to uncertain interdictions remain as before. The analogs of S-MIN-MAX-1(IB) (Model 3) and S-MIN-MAX-2(IB) (Model 4) are:

S-MIN-MAX-1(ICB) $\qquad w^* = \min_{\boldsymbol{\gamma}\in\Gamma} Eh(\boldsymbol{\gamma}, \tilde{\mathbf{I}}, \tilde{\mathbf{J}})$

(Model 8)

where:

$$h(\boldsymbol{\gamma}, \tilde{\mathbf{I}}, \tilde{\mathbf{J}}) = \max_{\mathbf{x}\in X} x_{ts},$$
$$\text{s.t. } 0 \leqslant x_{ij} \leqslant u_{ij}(\tilde{J}_{ij} - \tilde{I}_{ij}\tilde{J}_{ij}\gamma_{ij}) \qquad \forall\,(i,j) \in A',$$

and:

S-MIN-MAX-2(ICB) $\qquad w^* = \min_{\boldsymbol{\gamma}\in\Gamma} Eg(\boldsymbol{\gamma}, \tilde{\mathbf{I}}, \tilde{\mathbf{J}})$

(Model 9)

where:

$$g(\boldsymbol{\gamma}, \tilde{\mathbf{I}}, \tilde{\mathbf{J}}) = \max_{\mathrm{x}\in X} x_{ts} - \sum_{(i,j)\in A'} (1 - \tilde{J}_{ij} + \tilde{I}_{ij}\tilde{J}_{ij}\gamma_{ij}) x_{ij}.$$

In the “natural formulation” S-MIN-MAX-1(ICB), an arc (i, j) has its nominal capacity dropped to zero if it does not exist, i.e., if $\tilde{J}_{ij} = 0$; or if the arc is successfully interdicted, i.e., if $\tilde{I}_{ij}\gamma_{ij} = 1$. In the equivalent model S-MIN-MAX-2(ICB), we subtract any flow x_{ij} from the total flow if either of the above conditions hold. The correctness of the alternative formulation follows from Theorem 2. The development of bounds and an algorithm for SNIP(ICB) parallels the development for SNIP(IB) and is omitted. We remark only that the models may be transformed to a more standard form by viewing the random parameters as independent vectors of the form $\binom{\tilde{I}_{ij}}{\tilde{I}_{ij}, \tilde{I}_{ij}}$ with realizations $\binom{1}{1}, \binom{1}{0}, \binom{0}{0}$ and respective probabilities $\phi_{ij}p_{ij}$, $\phi_{ij}(1 - p_{ij})$, and $1 - \phi_{ij}$. Finally, note that SNIP(CB) (interdictions are certain but arc capacities are binary random variables) is the special case of SNIP(ICB) with $\text{Prob}(\tilde{I}_{ij} = 1) = 1$ for all $(i, j) \in A'$.

4.2. Discrete Random Arc Capacities, SNIP(CD)

We now generalize our models to handle random arc capacities that have a finite number of realizations. For notational simplicity and without loss of generality, we assume interdiction success is certain. This model is denoted “SNIP(CD).” Let $\tilde{\mathbf{u}}$ be the nonnegative random vector of arc capacities. Then the following model leads to upper bounds for SNIP(CD):

S-MIN-MAX-1(CG) $\quad w^* = \min_{\boldsymbol{\gamma}\in\Gamma} Eh(\boldsymbol{\gamma}, \tilde{\mathbf{u}})$

where:

$$h(\boldsymbol{\gamma}, \tilde{\mathbf{u}}) = \max_{\mathbf{x}\in X} x_{ts},$$

$$\text{s.t. } 0 \leq x_{ij} \leq \tilde{u}_{ij}(1-\gamma_{ij}) \qquad \forall\,(i,j)\in A'.$$

For fixed $\boldsymbol{\gamma}$, S-MIN-MAX-1(CG) is a stochastic maximum flow problem, and when $\tilde{\mathbf{u}}$ is replaced by $\bar{\mathbf{u}}$, we obtain the standard Jensen upper bound for this problem. The bound holds for more general nonnegative distributions in which the relevant expectations are finite. See Aneja and Nair (1980), Carey and Hendrickson (1984), and Wallace (1987a) for related methods for bounding the expected maximum flow in stochastic networks, i.e., bounding $Eh(\mathbf{0}, \tilde{\mathbf{u}})$.

In order to develop the lower bound, assume that each arc (i, j) can take on L different capacity values u_{ijl} such that $u_{ijl} \leq u_{ijl+1}$ for $l = 1, \ldots, L-1$. Let the "lth capacity increment" for arc (i, j) be u^{Δ}_{ijl}, i.e., $u^{\Delta}_{ijl} \equiv u_{ijl} - u_{ijl-1}$ for $l = 2, \ldots L$, and $u^{\Delta}_{ij1} \equiv u_{ij1}$. Let $\tilde{J}_{ijl}$ be the indicator random variable that is 1 if $\tilde{u}_{ij} \geq u_{ijl}$, and is 0 otherwise. For notational simplicity, assume that the return arc also has L capacity values, but that u_{tsL} is effectively infinite and $\tilde{J}_{tsL} \equiv 1$. Then S-MIN-MAX-1(CG) specializes to:

S-MIN-MAX-1(CD) $\quad w^* = \min_{\boldsymbol{\gamma}\in\Gamma} Eh'(\boldsymbol{\gamma}, \tilde{\mathbf{J}})$

where:

$$h'(\boldsymbol{\gamma}, \tilde{\mathbf{J}}) = \max_{\mathbf{x}} \sum_{l=1}^{L} x_{tsl}$$

$$\text{s.t. } \sum_{(i,j)\in FS(i)} \sum_{l=1}^{L} x_{ijl} - \sum_{(j,i)\in RS(i)} \sum_{l=1}^{L} x_{jil} = 0 \quad \forall\, i\in N,$$

$$0 \leq x_{ijl} \leq u^{\Delta}_{ijl}\tilde{J}_{ijl}(1-\gamma_{ij}) \quad \forall\,(i,j)\in A',\ l = 1, \ldots, L.$$

This model replaces each original arc with L parallel arcs in order to use the indicator random variables $\tilde{J}_{ijl}$ rather than the original random variables $\tilde{u}_{ij}$. For each arc (i, j), the random variables $\tilde{J}_{ijl}$ for $l = 1, \ldots, L$ are dependent, but this does not compromise the validity of the bound since Jensen's inequality does not require independence of the relevant random variables.

Given that we have represented S-MIN-MAX-1(CD) as a variant of the binary model S-MIN-MAX-1(ICB) (Model 8) with $\tilde{I}_{ij} \equiv 1$, we can now write the corresponding variant of S-MIN-MAX-2(ICB) (Model 9):

S-MIN-MAX-2(CD) $\quad w^* = \min_{\boldsymbol{\gamma}\in\Gamma} Eg(\boldsymbol{\gamma}, \tilde{\mathbf{J}})$

where:

$$g(\boldsymbol{\gamma}, \tilde{\mathbf{J}}) = \max_{\mathbf{x}} \sum_{l=1}^{L} x_{tsl} - \sum_{(i,j)\in A'} \sum_{l=1}^{L} (1 - \tilde{J}_{ijl} + \tilde{J}_{ijl}\gamma_{ij})\, x_{ijl}$$

$$\text{s.t. } \sum_{(i,j)\in FS(i)} \sum_{l=1}^{L} x_{ijl} - \sum_{(j,i)\in RS(i)} \sum_{l=1}^{L} x_{jil} = 0 \quad \forall\, i\in N,$$

$$0 \leq x_{ijl} \leq u^{\Delta}_{ijl} \quad \forall\,(i,j)\in A',\ l = 1, \ldots, L.$$

As in SNIP(IB) and SNIP(ICB), conditional expectations of $g(\boldsymbol{\gamma}, \tilde{\mathbf{J}})$ and $h(\boldsymbol{\gamma}, \tilde{\mathbf{u}})$ with respect to partitions of the associated state spaces lead to lower and upper bounds, respectively. Partitioning schemes can be extended to discrete random arc capacities by conditioning on whether the capacity value is at or below a certain level *or* above the specified level.

4.3. Multiple Uncertain Interdiction Attempts, SNIP(IM)

When interdiction success probabilities are low, or when certain arcs are critical to maximizing flow, it may be desirable to make more than one attempt at interdicting certain arcs. We model this situation with "SNIP(IM)." This model assumes that the successes of interdiction attempts are independent and that no intelligence about the success or failure of an interdiction attempt can be gathered until all attempts are made. (A multistage stochastic programming model would be needed if the state of the network could be observed after each interdiction attempt, or group of attempts, and these observations could influence subsequent attempts.) For notational and expository simplicity, we describe the modeling extensions for the special case in which an interdiction on any arc (i, j) may be attempted at most twice. Let $\tilde{I}_{ij1}$ and $\tilde{I}_{ij2}$ be the indicator random variables for the success of the first and second interdiction attempts, respectively, with associated success probabilities p_{ij1} and p_{ij2}.

There are several ways to formulate models for SNIP(IM), but we have chosen the following for its ease of implementation. Let γ_{ij1} be the first-stage decision variable that is 1 if at least one interdiction attempt is made on arc (i, j) (0 otherwise), and let γ_{ij2} be 1 if two attempts are made (0 otherwise). Thus, $\gamma_{ij1} = 1$ and $\gamma_{ij2} = 0$ when exactly one interdiction attempt is made. In our formulation, we must add constraints requiring that the "first attempt" be made before the second. Therefore, we define:

$$\Gamma' \equiv \Bigg\{ \boldsymbol{\gamma} \,\big|\, \gamma_{ij1}, \gamma_{ij2} \in \{0, 1\},$$

$$\gamma_{ij1} - \gamma_{ij2} \geq 0,\ \sum_{(i,j)\in A'} \sum_{k=1}^{2} r_{ijk}\gamma_{ijk} \leq R \Bigg\}, \qquad (7)$$

where r_{ijk} units of resource are required for the kth interdiction attempt on arc (i, j). The equivalent formulations for this problem are:

S-MIN-MAX-1(IM) $\quad w^* = \min_{\boldsymbol{\gamma}\in\Gamma'} Eh(\boldsymbol{\gamma}, \tilde{\mathbf{I}})$

where:

Table I
Test Problem Characteristics

Problem	First Stage (rows × cols.)	Subproblem (nodes × arcs)	Number of Stochastic Parameters	Number of Scenarios
SNIP(IB).7 × 5	1 × 22	37 × 61	22	4.2×10^{6}
SNIP(ICB).7 × 5	1 × 22	37 × 61	44	3.1×10^{10}
SNIP(IB).4 × 9	1 × 24	38 × 70	24	1.7×10^{7}
SNIP(ICB).4 × 9	1 × 24	38 × 70	48	2.8×10^{11}
SNIP(IB).10 × 10	1 × 84	102 × 183	84	1.9×10^{25}
SNIP(ICB).10 × 10	1 × 84	102 × 183	168	1.2×10^{40}

$$h(\gamma, \tilde{\mathbf{I}}) = \max_{\mathbf{x}\in X} x_{ts},$$
$$\text{s.t. } 0 \leqslant x_{ij} \leqslant u_{ij} \cdot (1 - \tilde{I}_{ij1}\gamma_{ij1} - (1-\tilde{I}_{ij1})\tilde{I}_{ij2}\gamma_{ij2}) \quad \forall (i,j) \in A',$$

and:

S-MIN-MAX-2(IM) $\quad w^* = \min_{\boldsymbol{\gamma}\in\Gamma'} Eg(\boldsymbol{\gamma}, \tilde{\mathbf{I}})$

where:

$$g(\boldsymbol{\gamma}, \tilde{\mathbf{I}}) = \max_{\mathbf{x}\in X} x_{ts} - \sum_{(i,j)\in A'} (\tilde{I}_{ij1}\gamma_{ij1} + (1-\tilde{I}_{ij1})\tilde{I}_{ij2}\gamma_{ij2}) x_{ij}.$$

Devising a recursive partitioning scheme to exploit these formulations is straightforward if we condition first on whether or not $\tilde{I}_{ij1} = 1$ and only when $\tilde{I}_{ij1} \equiv 0$, condition on whether or not $\tilde{I}_{ij2} = 1$. As in subsection 4.1, implementation can be facilitated by transforming the random parameters into more standard, independent vectors of the form $\binom{\tilde{I}_{ij1}}{(1-\tilde{I}_{ij1})\tilde{I}_{ij2}}$ with realizations $\binom{0}{0}, \binom{0}{1}, \binom{1}{0}$ and respective probabilities $(1 - p_{ij1})(1 - p_{ij2})$, $(1 - p_{ij1})p_{ij2}$, and p_{ij1}.

5. COMPUTATIONAL RESULTS

The algorithm described in Section 3 has been implemented and tested on the set of stochastic network interdiction problems described here. We begin by giving an overview of the implementation and the test problems. Then, two examples are used to illustrate the value of the stochastic solution for the binary interdiction problem SNIP(IB). Computational results are then summarized for some modest-sized model instances that also include binary random arc capacities (SNIP(ICB)). We also illustrate how computational difficulty varies with certain model parameters including the size of the interdiction resource budget and the values of the arc existence probabilities. Finally, we describe computational experience on a larger network model.

The algorithm is coded in FORTRAN and uses IBM's Optimization Subroutine Library (OSL) (1991) to solve the integer master problems and linear subproblems. The code is adapted from an implementation (Morton 1995) designed to handle general LP subproblems in which the data are read in SMPS format (Birge et al. 1987) and first-stage decisions are passed to the right-hand side of the second-stage constraints. Because the code was not originally designed to solve network interdiction problems, it does not exploit many special structures that could improve its efficiency. Several examples are (a) the network flow subproblems are solved by a general LP optimizer rather than a more efficient network optimizer; (b) the dual of the lower-bounding subproblem (see S-MIN-MAX-2(IB), Model 4) is solved, rather than the primal, in order to accommodate decisions being passed to the right-hand side instead of the objective; and (c) bounds on variables in the upper-bounding subproblems are implemented as structural constraints rather than as simple upper bounds. So, this code is prototypic, but it still effectively illustrates the value of the proposed methodology. All CPU times reported here are from an IBM RS/6000 Model 590 with 512 megabytes of random access memory.

The test problems are summarized in Table I. SNIP(P).$a \times b$ denotes a SNIP problem of type P on a network with an $a \times b$ rectangular grid of nodes, i.e., with a horizontal layers of b nodes each. Horizontal arcs are oriented from "west to east" but vertical arcs are oriented randomly. There are no arcs in parallel or anti-parallel. All westernmost nodes are connected to a supersource with artificial, noninterdictable, infinite-capacity arcs and all easternmost nodes are connected to a supersink with artificial, noninterdictable, infinite-capacity arcs. (Because of this structure, the easternmost vertical arcs are superfluous and are not considered as interdictable.) The node and arc data in Table I counts the supersource, supersink and artificial arcs. Figures 3 and 4 show SNIP(IB).4 × 9 and SNIP(IB).7 × 5.

For each problem, only a subset of the network's arcs involve uncertainty. For example, the SNIP(IB).7 × 5 network contains 37 nodes and 61 arcs (including the artificial nodes and arcs), but only 22 of the original arcs are eligible for interdiction. In the SNIP(ICB).7 × 5 example, those same 22 arcs have both uncertain interdiction success and uncertain capacity while other arcs are noninterdictable and have deterministic capacities. As described in subsection 4.1, the SNIP(ICB) models implement stochastic arc existence combined with stochastic interdiction success as sets of independent random 2-vectors with three realizations each. All p_{ij} values (interdiction success probabilities) are identical for each interdictable arc in any test problem, as are the ϕ_{ij} values (arc existence probabilities) in the SNIP(ICB) examples. For simplicity, $r_{ij} = 1$ for all problems; i.e., one unit of resource is required to interdict each interdictable arc.

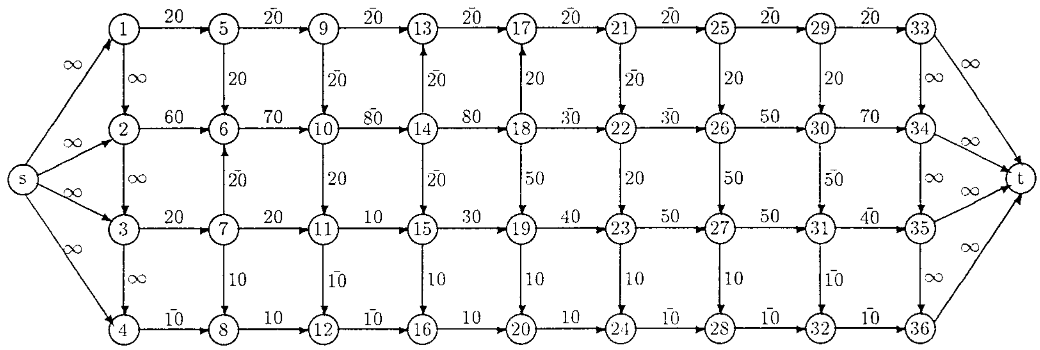

Figure 3. Sequential Approximation Algorithm for Solving SNIP(IB)

The number of scenarios and/or stochastic parameters is not always a good measure of the difficulty of a stochastic program. The values in Table I should be read with the following in mind: given a fixed first-stage decision, evaluating the recourse function for SNIP(ICB) requires dealing with one or two stochastic parameters for each interdictable arc, in particular, the arc-existence parameter and possibly the interdiction parameter if the first-stage decision interdicts the arc. However, for SNIP(IB), this evaluation only involves a subset of the stochastic interdiction parameters due the integral first-stage decision $\boldsymbol{\gamma}$. So, the number of stochastic interdiction parameters that is "active" at any one time corresponds to the budget available, rather than the total number of stochastic parameters. On the other hand, SNIP(IB) with $100R$ arcs, all interdictable, and an interdiction budget of R, is certainly harder to solve than a simple stochastic maximum flow problem with $99R$ deterministic arcs and R stochastic arcs, because the interdiction problem must decide where to "place the uncertainty."

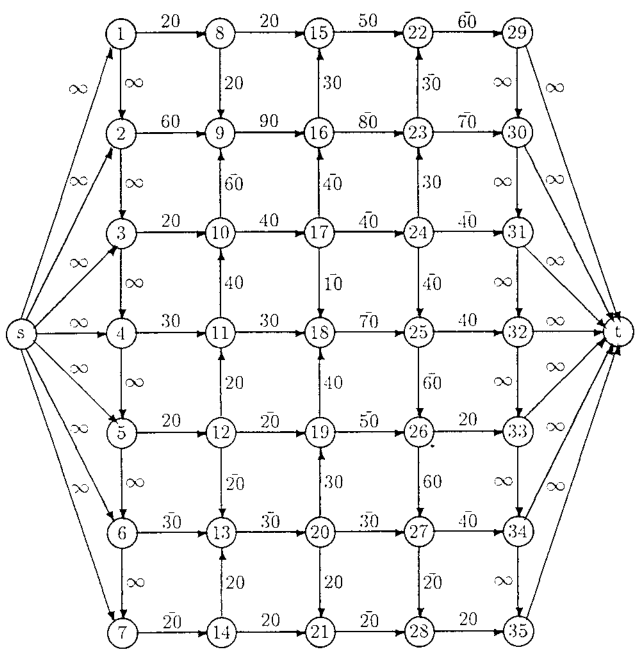

Figure 4. SNIP(IB).7 × 5 example. Node s is the supersource and node t the supersink. Arcs are labeled with their capacities and only arcs with bars over their capacities are interdictable.

In contrast to interdiction uncertainties, random arc existence parameters in SNIP(ICB) do not depend on $\boldsymbol{\gamma}$ and hence are always active.

5.1. Value of the Stochastic Solution

Table I indicates that the deterministic equivalents of our stochastic programming problems can be very large. This is typical of stochastic programs, and so, such problems are often approximately solved by solving expected-value models in which random parameters are replaced with their means. That this can lead to poor decisions has been demonstrated in the introduction. However, more can be learned about the pitfalls of the expected-value approach by investigating the differences in the expected-value solution and the true solution for more realistically sized examples. We perform this investigation here with two examples of SNIP(IB). All solution values we report in this section are computed with a relative error of no more than 0.01, where relative error is defined as (upper bound–lower bound)/(lower bound).

To perform this analysis, we first solve the expected-value model of the "natural formulation" S-MIN-MAX-1(IB) (Model 3), and denote the optimal solution $\boldsymbol{\gamma}_{EV}$. The performance of $\boldsymbol{\gamma}_{EV}$ is then evaluated in the stochastic environment; i.e., we calculate $Eh(\boldsymbol{\gamma}_{EV}, \tilde{\mathbf{I}})$. This objective value can be compared with the optimal objective $Eh(\boldsymbol{\gamma}^*, \tilde{\mathbf{I}})$; the difference between these values is referred to as "the value of the stochastic solution" (Birge 1982).

We begin with SNIP(IB).4 × 9 with an interdiction budget of $R = 6$ and $p_{ij} = 0.75$ for all interdictable arcs. (See Figure 3.) In the expected-value problem, the minimum capacity cut consists of four interdicted arcs, arcs (17, 21), (22, 26), (31, 35) and (32, 36). When the budget resource constraint is implemented as an inequality, only these four arcs are interdicted in our solution, even though there is an interdiction budget of six. For this model we obtain, $Eh(\boldsymbol{\gamma}_{EV}, \tilde{\mathbf{I}}) = 25.0$ while $Eh(\boldsymbol{\gamma}^*, \tilde{\mathbf{I}}) = 10.9$. That is, the stochastic solution improves the stochastic objective by more than a factor of two. When the budget constraint is implemented as an equality, two additional arcs (arcs (12,

Table II
Relative Errors vs. CPU Time for SNIP(IB).7 × 5 with $p_{ij} = 0.75$ and $R = 6$, and for SNIP(ICB).7 × 5 with $p_{ij} = 0.75$, $\phi_{ij} = 0.9$, and $R = 6$

	SNIP(IB).7 × 5 Relative Error				SNIP(ICB).7 × 5 Relative Error		
Number of Cells	(Local)	(Global)	Cumulative CPU sec.	Number of Cells	(Local)	(Global)	Cumulative CPU sec.
1		0.42	1.1	1		0.95	1.5
2	0.17	0.17	2.5	12	0.43	0.64	6.3
4	0.058	0.11	7.3	20	0.30	0.34	21.3
7	0.046	0.058	11.8	55	0.17	0.22	46.0
10	0.022	0.037	17.8	126	0.11	0.11	104.9
12	0.017	0.017	20.3	327	0.055	0.068	228.9
14	0.0052	0.0052	29.2	742	0.034	0.038	433.1
				1648	0.019	0.019	848.0
				3735	0.0099	0.0099	1532.4

16) and (31, 32)) are interdicted in the expected-value problem, but poor choices are made by the model and we still obtain $Eh(\boldsymbol{\gamma}_{EV}, \tilde{\mathbf{I}}) = 25.0$. The stochastic solution selects the same four arcs as the expected-value model, but an intelligent choice of the additional two arcs (arcs (5, 9) and (10, 14)) significantly improves the quality of the solution. So, in this example, one of the expected-value solutions is an optimal stochastic solution, but the expected-value model has no way of differentiating among its alternatives and chooses a wrong one.

The second example, shown in Figure 4, is SNIP(IB).7 × 5. Once again, we use an interdiction budget of $R = 6$ and $p_{ij} = 0.75$ for all interdictable arcs. In this model, the budget and problem structure allow the expected-value model to interdict only a subset of the arcs on a minimum capacity cut. Unlike the previous example, the expected value and stochastic solutions share only two arcs in common. (The expected-value solution interdicts arcs (12, 19), (13, 20), (17, 24), (18, 25), (22, 29), and (23, 30). The stochastic solution interdicts arcs (6, 13), (7, 14), (17, 16), (17, 24), (18, 25), and (23, 22)). In fact, the total nominal capacity of interdicted arcs in the expected-value solution is 290 while the corresponding capacity for the stochastic solution is 230. The stochastic program sacrifices 60 units of nominal capacity in order to improve the solution. For this model, $Eh(\boldsymbol{\gamma}_{EV}, \tilde{\mathbf{I}}) = 119.6$ while $Eh(\boldsymbol{\gamma}^*, \tilde{\mathbf{I}}) = 80.4$.

5.2. Small Test Problems and Computational Enhancements

As noted in subsection 3.2.1, every time the first-stage decision changes, the Jensen bounds on each cell of the partition must be reevaluated. However, solution speed can be increased significantly by performing a number of cell refinements prior to re-solving the lower-bounding problem. This idea is implemented as follows: After having performed Steps 1 and 3 of the algorithm (see Figure 2), a relative error is calculated from the global upper and lower bounds associated with the current solution $\hat{\boldsymbol{\gamma}}$. We then refine the partition with respect to the fixed $\hat{\boldsymbol{\gamma}}$ until the local relative error is cut in half. Note that during this procedure, the recourse function need only be evaluated on new cells. We then perform the next iteration of the algorithm. The upper bound produced for fixed $\hat{\boldsymbol{\gamma}}$ is globally valid but a global lower bound is obtained only by reoptimizing the lower-bounding problem.

Tables II and III display local and global relative errors as the algorithm converges for four test problems. They also reveal a number of interesting features regarding the problems, bounds, and algorithm. The initial Jensen bounds are considerably worse for the "longer" 4 × 9 network than for the 7 × 5 network. This is consistent with the presence of redundant penalty terms in the lower-bounding objective. For instance, consider the initial lower-bounding problem for SNIP(IB). If flow moves along a path with two interdicted arcs (i, j) and (i', j') in this model, a penalty of $p_{ij} + p_{i'j'}$ is subtracted in the objective of the recourse function. However, this flow will be stopped if either interdiction is successful and the penalty should only be $p_{ij} + p_{i'j'} - p_{ij}p_{i'j'}$. One may be able to reduce this effect by using a path (or partial-path) formulation of the model (Aneja and Nair 1980).

The results indicate that the higher dimensionality introduced by including random arc capacities makes computation significantly more expensive. The tables display relative errors associated with the current solution $\hat{\boldsymbol{\gamma}}$, in order to convey some of the volatility of the quality of the decisions generated by the sequence of lower-bounding problems. In particular, see Table III for the 4 × 9 problems. The global relative error is typically larger than the local error for the same partition (because the global lower bound is no larger than the local lower bound). However, within the local/global refinement procedure, the lower-bounding problem occasionally generates a new first-stage decision that significantly improves the upper bound and the reverse occurs. For example, see the sixth row of the results for SNIP(ICB).4 × 9.

To illustrate the value of the local/global refinement procedure over naive refinement, we performed experiments in which the lower-bounding problem is reoptimized every time a cell is subdivided. For this strategy, SNIP(IB).7 × 5 requires 12 cells and 53.5 seconds, while

Table III

Relative Errors vs. CPU Time for SNIP(IB).4 × 9 with $p_{ij} = 0.75$, $R = 6$, and SNIP(ICB).4 × 9 with $p_{ij} = 0.75$, $\phi = 0.9$, and $R = 6$

	SNIP(IB).4 × 9 Relative Error				SNIP(ICB).4 × 9 Relative Error		
Number of Cells	(Local)	(Global)	Cumulative CPU Sec.	Number of Cells	(Local)	(Global)	Cumulative CPU Sec.
1		4.0	2.5	1		9.0	1.0
2	1.1	1.2	9.7	3	4.0	4.0	2.2
3	0.42	0.47	15.4	4	1.5	1.7	3.8
4	0.16	0.42	17.7	9	0.72	1.0	8.2
6	0.14	0.14	31.7	16	0.50	0.87	14.8
7	0.064	0.16	32.9	31	0.42	0.39	25.1
8	0.064	0.16	34.3	49	0.19	0.33	42.6
10	0.022	0.10	43.3	75	0.16	0.47	65.8
12	0.031	0.031	48.3	98	0.23	0.19	91.3
13	0.015	0.065	52.2	140	0.10	0.16	129.1
14	0.016	0.015	56.3	207	0.078	0.088	189.1
15	0.0036	0.015	59.8	269	0.044	0.049	239.9
16	0.0063	0.0063	66.1	351	0.025	0.037	308.7
				439	0.018	0.020	373.5
				535	0.0099	0.011	465.0
				543	0.0099	0.0099	481.6

SNIP(IB).4 × 9 requires 16 cells and 84.0 seconds to achieve a global relative error of 1%. SNIP(ICB).7 × 5 requires 15855.2 seconds to form 1116 cells with a relative error of 2.1% (at which point the algorithm was prematurely terminated), and SNIP(ICB).4 × 9 achieves a relative error of 1% with 358 cells in 4134.5 seconds. Contrasting these values with those in Tables II and III, we see that the local/global refinement procedure, which provides a formal mechanism for deciding how many cells to subdivide prior to resolving the lower bounding problem, can lead to significant computational savings.

Table IV reveals how problem difficulty varies with the size of the interdiction budget. When $R = 0$ or $R \geqslant 24$, SNIP(ICB).4 × 9 reduces to a stochastic maximum flow problem with unreliable arcs (e.g., Carey and Hendrickson (1984), Evans (1976)) because the optimal first-stage decision is trivial. However, in this case, computing tight bounds on the expected maximum flow still requires a fairly fine partition with corresponding computational effort.

For general R, the master problem is a difficult integer knapsack problem with a piecewise linear objective function formed by optimality cuts. (Of course, for our test problems, the knapsack problem by itself is quite simple since $r_{ij} = 1$ for all interdictable arcs.) The results show that for moderate values of R the problem is most difficult. A finer partition is required to close the gap between bounds when these values are moderate, so there are more master problems and subproblems to solve. Also, the more partitions there are, the more master-problem cuts there tend to be and the integer master problems become harder and harder to solve. Our implementation solves each master problem from scratch and increased efficiency could probably be achieved by using information from previous iterations for initial solution and bounding purposes.

Table V shows that the bounds are the weakest, and hence require finer partitions, when the uncertainty regarding arc existence is greatest, i.e., when $\phi_{ij} \approx 0.5$. Even small deviations from 0.5, particularly to larger values, make the problem significantly easier. The asymmetry in

Table IV

SNIP(ICB).4 × 9 with $\phi_{ij} = 0.9$ and $p_{ij} = 0.75$ for Various Interdiction Resource Budgets

		Master Probs. Solved		Subproblems Solved			
Budget Value R	Num. of Cells	(Number)	(CPU Sec.)	(Number for Lower Bnd.)	(Number for Upper Bnd.)	(CPU Sec.)	Total CPU Sec.
0	56	0	0.0	2290	2145	18.6	22.6
3	93	37	3.5	4681	4068	37.9	45.8
6	543	151	102.8	47205	26100	417.3	581.6
9	725	195	226.4	80440	37030	731.1	995.4
12	768	106	24.4	74335	43089	492.3	595.9
15	474	47	0.5	36097	28823	237.6	295.2
18	362	25	0.11	25447	24240	174.3	271.6
21	422	27	0.13	33176	31345	224.4	281.6
24	287	0	0.0	23911	23033	161.6	202.4

Note: all problems are solved to within a relative error of 0.01.

Table V
SNIP(ICB).7 $\times$ 5 with $R = 0$ for Various Arc Existence Probabilities

ϕ_{ij}	Number of Cells	CPU Sec.	Objective Bounds
0.9	53	20.0	[181.5, 183.3]
0.7	1063	342.8	[138.1, 139.5]
0.55	2217	697.3	[101.4, 102.4]
0.5	6496	1954.9	[89.10, 89.99]
0.45	5033	1527.7	[77.02, 77.78]
0.3	1112	359.0	[43.59, 44.02]
0.1	185	65.1	[10.32, 10.42]

Note: all problems are solved to within a relative error of 0.01.

Table V is due, in part, to the fact that a 1% relative error is more difficult to achieve when the objective function values are smaller, i.e., when ϕ_{ij} is smaller. However, an absolute error termination criterion would reverse the asymmetry.

5.3. Larger Test Problems

Table I indicates that SNIP(IB).10 $\times$ 10 and SNIP(ICB).10 $\times$ 10 are significantly larger than the problems analyzed above, particularly with respect to the number of the random parameters. Certain special cases of the 10 $\times$ 10 models are tractable with respect to the proposed solution technique. Table VI shows the effort required to achieve various relative errors for a "highly" reliable stochastic maximum flow problem ($R = 0$) with arc existence probabilities of 0.95. The total number of upper- and lower-bounding subproblems solved to achieve a relative error of 1% is just over 1×10^6. Table VII illustrates that SNIP(IB).10 $\times$ 10 is tractable when the interdiction budget is relatively small. As expected, the computational effort to solve the integer master problem grows exponentially with R and quickly becomes unmanageable.

6. CONCLUSIONS

This paper has studied stochastic variants of a network interdiction problem where an interdictor attempts to destroy arcs in a capacitated network through which an adversary will subsequently maximize flow. The problem is formulated as a mixed-integer stochastic program with a "min-max" objective, although it is possible to convert this to a more standard stochastic integer program with a minimizing objective.

The stochastic variants include cases where one or more uncertain interdiction attempts may be made, cases where the arc capacities are random variables, and certain combinations and extensions of these cases. The objective of the problem is to plan interdictions so that the expected maximum flow, after interdiction attempts are made, is minimized. The models for these problems are two-stage stochastic programs with recourse.

We develop a sequential approximation algorithm for solving these problems that recursively refines a partition of the state-space of the random variables to improve upper and lower bounds on the objective. Jensen's inequality applied in the usual fashion to the "natural formulation" of the models leads to upper bounds. A key result allows us to reformulate each model so that Jensen's inequality can also be used to compute lower bounds. An important computational advantage of our bounding models is that their size increases only with the number of elements in the state-space partition, not with the number of stochastic parameters.

The natural formulation of the model minimizes a concave function with respect to the convex hull of the integer first-stage feasible region. An equivalent reformulation yields a convex recourse function over the same space and thus allows direct application of an integer L-shaped method. This reformulation makes the problem tractable. Two examples showed that the value of the stochastic solution can be quite large for stochastic network interdiction problems (greater than 50% for one example). Computational results for a number of test problems demonstrated the merit of the proposed solution technique.

It is clear that the key reformulation technique of this paper leads to a new method for evaluating the expected maximum flow in a stochastic network. The reader may also see that the technique can be applied to other stochastic network problems dealing with reliability, shortest path length, project completion date in a PERT problem, etc. We are already programming algorithms to solve these problems and exploiting the reformulation technique in a number of other stochastic network models. We are also developing analogous bounds for more general stochastic programs.

Table VI
Relative Error vs. CPU Time for SNIP(ICB).10 $\times$ 10 with $\phi_{ij} = 0.95$ and $R = 0$.

Number of Cells	Relative Error	Cumulative CPU Sec.
115	0.048	329.5
625	0.024	1623.5
2993	0.012	7598.2
4201	0.0099	10684.4

ACKNOWLEDGMENTS

This research has been supported by grants from the Office of Naval Research and the Air Force Office of Scientific Research. David Morton's research was partially supported by the National Research Council under a Research Associateship at the Naval Postgraduate School and by a Summer Research Assignment Grant from the University of Texas at Austin. Kevin Wood's research was performed, in part, while he was Visiting Associate Professor in the Operations Research Department at Stanford University, and he thanks the university and department for its support. We are also grateful for the comments of two anonymous referees that helped improve the presentation of this paper.

Table VII
SNIP(IB).10 $\times$ 10 with p_{ij} = 0.75 for Various Interdiction Resource Budget Values

		Master Problems		Subproblems		
Budget Value R	Number of Cells	(Number)	(CPU Sec.)	(Total Num.)	(CPU Sec.)	Total CPU Sec.
5	15	26	34.2	325	2.9	38.5
6	11	43	295.8	327	3.5	300.7
7	14	64	1668.5	401	4.2	1674.2
8	63	145	23927.7	3114	32.6	23970.5

Note: all problems are solved to within a relative error of 0.01.

REFERENCES

AHUJA, R. K., T. L. MAGNANTI, AND J. B. ORLIN. 1993. *Network Flows*. Prentice Hall, Englewood Cliffs, NJ.

ANEJA, Y. P., AND K. P. K. NAIR. 1980. Maximal Expected Flow in a Network Subject to Arc Failures. *Networks* **10,** 45–57.

BIRGE, J. R. 1982. The Value of the Stochastic Solution in Stochastic Linear Programs with Fixed Recourse. *Math. Prog.* **24,** 314–325.

BIRGE, J. R., M. A. H. DEMPSTER, H. I. GASSMANN, E. A. GUNN, A. J. KING, AND S. W. WALLACE. 1987. A Standard Input Format for Multiperiod Stochastic Linear Programs. *COAL: Committee on Algorithms of the Mathematical Programming Society* **17,** 1–19.

BIRGE, J. R., AND S. W. WALLACE. 1988. A Separable Piecewise Linear Upper Bound for Stochastic Linear Programs. *SIAM J. Control Optim*. **26,** 725–739.

BIRGE, J. R., AND R. J.-B. WETS. 1989. Sublinear Upper Bounds for Stochastic Programs with Recourse. *Math. Prog.* **43,** 131–149.

BIRGE, J. R., AND R. J.-B. WETS. 1986. Designing Approximation Schemes for Stochastic Optimization Problems, in Particular, for Stochastic Programs with Recourse. *Math Prog. Study* **27,** 54–102.

CACCETTA, L. 1984. Vulnerability of Communication Networks. *Networks* **14,** 141–146.

CAREY, M., AND C. HENDRICKSON. 1984. Bounds on Expected Performance of Networks with Links Subject to Failure. *Networks* **14,** 439–456.

DONOHUE, C. J., AND J. R. BIRGE. 1995a. An Upper Bound on the Expected Value of a Non-Increasing Convex Function with Convex Marginal Return Functions. Working Paper, Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI.

DONOHUE, C. J., AND J. R. BIRGE. 1995b. An Upper Bound on the Network Recourse Function. Working Paper, Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI.

EVANS, J. R. 1976. Maximal Flow in Probabilistic Graphs: The Discrete Case. *Networks* **6,** 161–183.

FRAUENDORFER, K. 1992. *Stochastic Two-Stage Programming*. Lecture Notes in Economics and Mathematical Systems 392, Springer-Verlag, Berlin.

GODDARD, W. 1994. Measures of Vulnerability—The Integrity Family. *Networks* **24,** 207–213.

HAUSCH, D. B., AND W. T. ZIEMBA. 1983. Bounds on the Value of Information in Uncertain Decision Problems II. *Stochastics* **10,** 181–217.

HUANG, C. C., W. T. ZIEMBA, AND A. BEN-TAL. 1977. Bounds on the Expectation of a Convex Function of a Random Variable with Applications to Stochastic Programming. *Opns. Res.* **25,** 315–325.

IBM CORPORATION. 1991. *Optimization Subroutine Library Guide and Reference, Release 2*. Kingston, NY.

KALL, P., A. RUSZCZYŃSKI, AND K. FRAUENDORFER. 1988. Approximation Techniques in Stochastic Programming. In *Numerical Techniques for Stochastic Optimization*, Y. Ermoliev and R. J.-B. Wets (eds.). Springer Verlag, Berlin.

LAPORTE, G., AND F. V. LOUVEAUX. 1993. The Integer L-shaped Method for Stochastic Integer Programs with Complete Recourse. *O. R. Letts.* **13,** 133–142.

MADANSKY, A. 1959. Bounds on the Expectation of a Convex Function of a Multivariate Random Variable. *Ann. Math. Stat.* **30,** 743–746.

MORTON, D. P. 1995. A Sampling-Based Algorithm for Two-Stage Stochastic Linear Programming. Working Paper, Department of Mechanical Engineering, The University of Texas at Austin.

MCMASTERS, A. W., AND T. M. MUSTIN. 1970. Optimal Interdiction of a Supply Network. *Naval Res. Logist.* **17,** 261–268.

PHILLIPS, C. A. 1992. The Network Destruction Problem. SAND-92-0186C, Sandia National Laboratories, Albuquerque, NM.

STEINRAUF, R. L. 1991. A Network Interdiction Model. M.S. Thesis, Naval Postgraduate School, Monterey, CA.

VAN SLYKE, R. M., AND R. J.-B. WETS. 1969. L-Shaped Linear Programs with Applications to Optimal Control and Stochastic Programming. *SIAM J. Appl. Math.* **17,** 638–663.

WALLACE, S. W. 1987a. Investing in Arcs in a Network to Maximize the Expected Max Flow. *Networks* **17,** 87–103.

WALLACE, S. W. 1987b. A Piecewise Linear Upper Bound on the Network Recourse Function. *Math Prog.* **38,** 133–146.

WASHBURN, A. R., AND R. K. WOOD. 1995. Deterministic Network Interdiction. *Opns. Res.* **43,** 243–251.

WETS, R. J.-B. 1966. Programming Under Uncertainty: The Equivalent Convex Program. *SIAM J. Appl. Math.* **14,** 89–105.

WOLLMER, R. D. 1964. Removing Arcs from a Network. *J. Opns. Soc. Amer.* **12,** 934–940.

WOLLMER, R. D. 1980. Two-Stage Linear Programming Under Uncertainty with 0-1 Integer First-Stage Variables. *Math. Prog.* **19,** 279–288.

WOOD, R. K. 1993. Deterministic Network Interdiction. *Math. Comput. Modelling* **17,** 1–18.