Reformulation and Sampling to Solve a Stochastic Network Interdiction Problem

Udom Janjarassuk
Jeff Linderoth

ISE Department
COR@L Lab
Lehigh University
jtl3@lehigh.edu

COR@L
COMPUTATIONAL OPTIMIZATION RESEARCH AT LEHIGH

INFORMS Annual Meeting
San Francisco, CA
November 15, 2005

Network Interdiction

Problem Elements

Capacitated network, good guys, bad guys

Good Guy:

Network Interdiction

Problem Elements

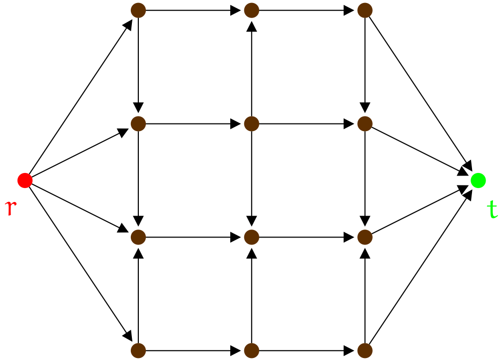
Capacitated network, good guys, bad guys

Bad Guy:

Network Interdiction

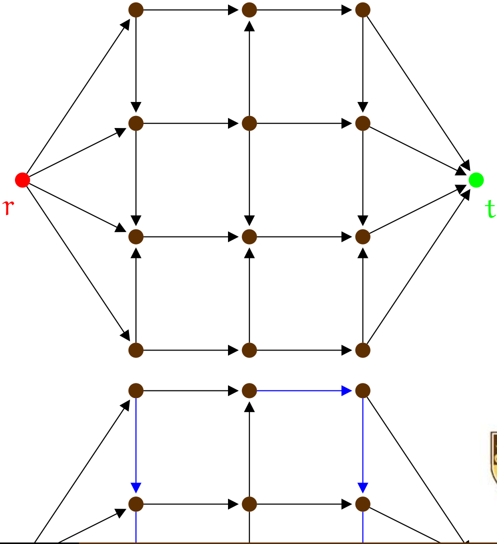
Bad Guy Says:

"I want to flow as much as possible from r to t"



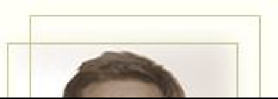
- Drugs, Enemy Supplies, Nuclear Material

Network Interdiction



Good Guy Says

"Not So Fast My Friend"



Mathematical Formulation

- Interdiction is a binary decision:

$$x_{ij} = \begin{cases} 1 & \text{if interdiction occurs on arc } (i,j) \in A, \\ 0 & \text{otherwise.} \end{cases}$$

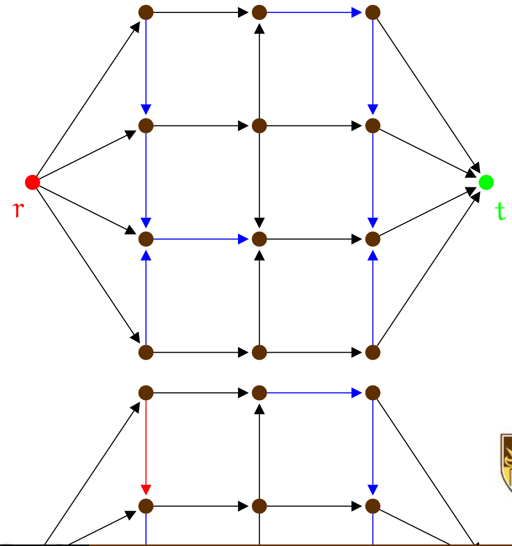
- $f(x)$: maximum flow in network if I intervene on arcs x
- Budget Constraint:

$$X = \left\{ x \in \{0,1\}^{|A|} \mid \sum_{(i,j)\in A} h_{ij} x_{ij} \le K \right\}$$

Network Interdiction Problem

$$\min_{x \in X} f(x)$$

Stochastic Network Interdiction



Jeff Is An Idiot

Jeff's interdictions are not always successful

Stochastic Network Interdiction

- S: Set of scenarios
- ξ_{ijs}: Bernoulli random variable if interdiction on arc (i, j) would be successful in scenario s
- $f_s(x)$: maximum flow if Dudley intervenes on arcs x and scenario $s \in S$ occurs

Stochastic Network Interdiction Problem (SNIP)

$$\min_{x \in X} \mathbb{E} f_s(x) = \min_{x \in X} \sum_{s \in S} p_s f_s(x)$$

Max Flow

- Formulate Max Flow $f_s(x)$ as an LP: $A' = A \cup \{(t, r)\}$

Primal

$$f_s(x) = \max_{y \in \mathbb{R}_+^{|A'|}} y_{tr}$$

$$y_{ij} \leq u_{ij}(1 - \xi_{ijs} x_{ij}) \quad \forall (i,j) \in A$$

$$Ny = 0$$

Dual

$$\min \sum_{(i,j) \in A} u_{ij}(1 - \xi_{ijs} x_{ij}) \rho_{ij}$$

$$\begin{aligned} \pi_r - \pi_t &\geq 1 \\ \rho_{ij} - \pi_i + \pi_j &\geq 0 \quad \forall (i,j) \in A \\ \rho_{ij} &\geq 0 \quad \forall (i,j) \in A \end{aligned}$$

Strong Duality

- For fixed $\hat{x}$, if Dudley can find (primal) feasible y^* and (dual) feasible (π^*, ρ^*) such that

$$y^*_{tr} = \sum_{(i,j)\in A} u_{ij}(1 - \xi_{ijs}\hat{x}_{ij})\rho_{ij} \tag{1}$$

then max flow $f_s(\hat{x}) = y^*_{tr}$

Formulation Idea

1. Duplicate (primal and dual) flow variables (y, π, ρ) for each scenario
2. Enforce primal feasibility, dual feasibility, and equality (1) for each scenario

A $\min\min$ SNIP Formulation

$$\min \sum_{s \in S} p_s y_{trs}$$

subject to

$$\begin{aligned}
y_{trs} - \sum_{(i,j)\in A} u_{ij}(1-\xi_{ijs}x_{ij})\rho_{ijs} &= 0 && \forall s \in S \\
\sum_{(i,j)\in A} h_{ij}x_{ij} &\leq K \\
y_{ijs} - u_{ij}(1-\xi_{ijs}x_{ij}) &\leq 0 && \forall (i,j) \in A, \forall s \in S \\
Ny_s &= 0 && \forall s \in S \\
\pi_{rs} - \pi_{ts} &\geq 1 && \forall s \in S \\
\rho_{ijs} - \pi_{is} + \pi_{js} &\geq 0 && \forall (i,j) \in A, \forall s \in S \\
(x, y, \pi, \rho) &\in \mathbb{B}^{|A|} \times \mathbb{R}_+^{|A'||S|} \times \mathbb{R}^{|N||S|} \times \mathbb{R}_+^{|A||S|}
\end{aligned}$$

Good and Bad

Some nice things about the formulation

- It's a pure minimization problem
- There are not "too many" integer variables: $(x \in \mathbb{B}^{|A|})$
- If x is fixed, it is decomposible by scenario
- Integer variables appear only in the first stage!

A BAD thing about the formulation

$$y_{trs} - \sum_{(i,j) \in A} u_{ij}(1 - \xi_{ijs} x_{ij}) \rho_{ijs} = 0$$

- $x_{ij}\rho_{ijs}$ are nonlinear terms

Linearization Trick

- Introduce auxiliary variables z_{ijs}
- Let M be an upper bound on ρ_{ijs}
- Then $z_{ijs} = x_{ij}\rho_{ijs}$ if and only if
  1. $z_{ijs} \leq Mx_{ij}$
  2. $z_{ijs} \leq \rho_{ijs}$
  3. $z_{ijs} \geq \rho_{ijs} + M(x_{ij} - 1)$

Lemma

In any optimal solution to the dual max flow problem,

$$\rho_{ijs} \leq 1, \ \forall (i,j) \in A.$$

SNIP: MILP Formulation

$$\min \sum_{s \in S} p_s y_{trs}$$

subject to

$$\begin{aligned}
y_{trs} - \sum_{(i,j)\in A} u_{ij}\rho_{ijs} + \sum_{(i,j)\in A} u_{ij}\xi_{ijs}z_{ijs} &= 0 \quad \forall s \in S \\
z_{ijs} - x_{ij} &\leq 0 \quad \forall (i,j) \in A, \forall s \in S \\
z_{ijs} - \rho_{ijs} &\leq 0 \quad \forall (i,j) \in A, \forall s \in S \\
\rho_{ijs} - z_{ijs} + x_{ij} &\leq 1 \quad \forall (i,j) \in A, \forall s \in S \\
\text{Primal Feasibility} & \\
\text{Dual Feasibility} &
\end{aligned}$$

$$(x, y, \pi, \rho, z) \in \mathbb{B}^{|A|} \times \mathbb{R}_+^{|A'||S|} \times \mathbb{R}^{|N||S|} \times \mathbb{R}_+^{|A||S|} \times \mathbb{R}_+^{|A||S|}$$

Good News and Bad News

- Problem is "just" an IP
- But it's big

Name	K	$\lvert N \rvert$	$\lvert A \rvert$	$\lvert B \rvert$	$\lvert S \rvert$
SNIP4x4	4	18	32	9	512
SNIP7x5	6	37	72	22	4.2×10^6
SNIP4x9	6	38	67	24	1.7×10^7
SNIP10x10	10	102	200	65	3.7×10^{19}
SNIP20x20	20	402	800	253	1.4×10^{76}

- SNIP10x10 has a mere XX variables and XX constraints

Monte Carlo Methods

$$\min_{x \in S} \{ f(x) \equiv \mathbb{E}_P g(x; \xi) \equiv \int_\Omega g(x; \xi) dP(\xi) \}$$

- Draw $\xi^1, \xi^2, \ldots \xi^N$ from P
- Sample Average Approximation:

$$\widehat{f}_N(x) \equiv N^{-1} \sum_{j=1}^{N} g(x, \xi^j)$$

- $\widehat{f}_N(x)$ is an unbiased estimator of $f(x)$ $(\mathbb{E}[\widehat{f}_N(x)] = f(x))$.
- We instead minimize the Sample Average Approximation:

$$\min_{x \in S} \{ \widehat{f}_N(x) \}$$

Lower Bound on the Optimal Objective Function Value

$$v^* = \min_{x \in S}\{f(x)\}$$

$$\hat{v}_N = \min_{x \in S}\{\widehat{f}_N(x)\}$$

Thm:

$$\mathbb{E}[\hat{v}_N] \le v^*$$

- The expected optimal solution value for a sampled problem of size N is $\le$ the optimal solution value.

Estimating $\mathbb{E}[\hat{v}_N]$

- Generate M independent SAA problems of size N.
- Solve each to get $\widehat{v}_N^j$

$$L_{N,M} \equiv \frac{1}{M} \sum_{j=1}^{M} \widehat{v}_N^j$$

- The estimate $L_{N,M}$ is an unbiased estimate of $\mathbb{E}[\widehat{v}_N]$.

$$\sqrt{M}\,[L_{N,M} - \mathbb{E}(\widehat{v}_N)] \to \mathcal{N}(0, \sigma_L^2)$$

- $\sigma_L^2 \equiv \text{Var}(\widehat{v}_N)$
- This variance depends on the sample!

Confidence Interval

$$s_L^2(M) \equiv \frac{1}{M-1} \sum_{j=1}^{M} \left(\widehat{v}_N^j - L_{N,M} \right)^2$$

$$\left[L_{N,M} - \frac{z_\alpha s_L(M)}{\sqrt{M}}, L_{N,M} + \frac{z_\alpha s_L(M)}{\sqrt{M}} \right]$$

- These only apply if the $\widehat{v}_N^j$ are i.i.d. random variables.

Upper Bounds

$$f(\hat{x}) \geq v^* \quad \forall \hat{x} \in S$$

- Generate T independent batches of samples of size $\bar{N}$

$$\mathbb{E}\left[\widehat{f}^j_{\bar{N}}(x) := \bar{N}^{-1} \sum_{i=1}^{\bar{N}} g(x, \xi^{i,j})\right] = f(x), \quad \text{for all } x \in X.$$

$$U_{\bar{N},T}(\hat{x}) := T^{-1} \sum_{j=1}^{T} \widehat{f}^j_{\bar{N}}(\hat{x})$$

More Confidence Intervals

$$\sqrt{T}\left[U_{\bar{N},T}(\hat{x}) - f(\hat{x})\right] \Rightarrow N(0, \sigma_U^2(\hat{x})), \quad \text{as } T \to \infty,$$

- $\sigma_U^2(\hat{x}) \equiv \text{Var}\left[\hat{f}_{\bar{N}}(\hat{x})\right]$
- Estimate $\sigma_U^2(\hat{x})$ by the sample variance estimator $s_U^2(\hat{x}, T)$

$$s_U^2(\hat{x}, T) \equiv \frac{1}{T-1}\sum_{j=1}^{T}\left[\hat{f}_{\bar{N}}^{j}(\hat{x}) - U_{\bar{N},T}(\hat{x})\right]^2.$$

$$\left[U_{\bar{N},T}(\hat{x}) - \frac{z_\alpha s_U(\hat{x};T)}{\sqrt{T}}, U_{\bar{N},T}(\hat{x}) + \frac{z_\alpha s_U(\hat{x};T)}{\sqrt{T}}\right]$$

Solution Times

Table of sizes

Table of CPLEX times

Use a Decomposition Approach

- ATR: A parallel solver for two-stage stochastic linear programs, engineered to run a collection of (Condor provided) non-dedicated CPUs
- Uses MW: Master-Worker framework for parallelization
- Master: Solves master problem (to determine $\hat{x}$)
- Workers: Evaluate $f_s(\hat{x}\ \forall s \in S$ by solving (indepedent) linear programs
- Since $f_s(x)$ is still convex in x, the same cutting plane-based decomposition approach works to solve the problem, but we need just solve the master problem as an integer program
- For many small SNIP instances, the solution of the linear relaxation comes out to be nearly integer

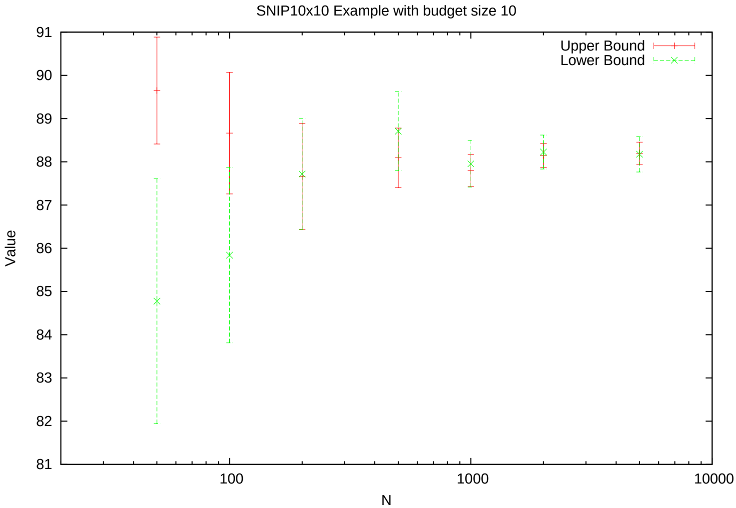
Solution Approach

1. Solve LP Relaxation of SNIP (using ATR)
2. Round (or keep track of relaxed iterations) to get an UB
3. Remove all (most) inactive optimality cuts
4. Solve IP using ATR

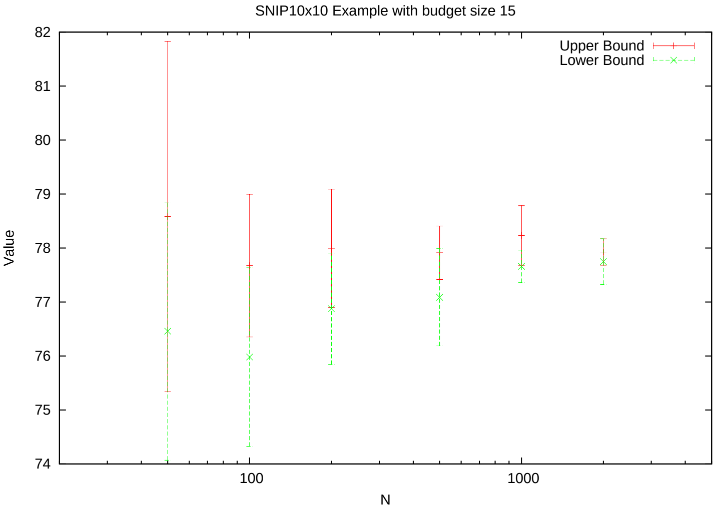
Computational Details

- Run an very small configuration of < 50 workstations in COR@L lab and Grid lab at Lehigh
- Run each instance $M = 10$ times
- Upper Bound: $N' = ?$
- Did not use any variance reduction techniques in the sampling, which can greatly speed the convergence rate!

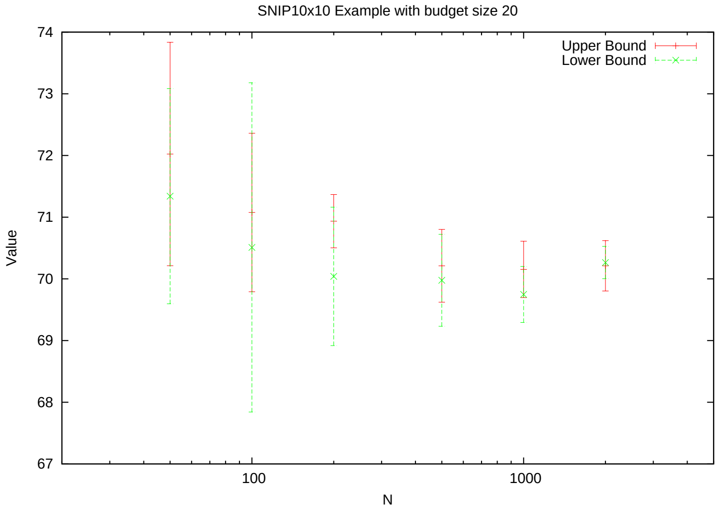
SNIP10x10 $K = 10$



SNIP10x10 $K = 15$



SNIP10x10 $K = 20$



CPU Times: SNIP10x10, $K = 10$

Sample size	Master (avg)	Stdev	Wall Clk	Stdev
50	6.76	7.00	133.44	96.90
100	12.04	17.50	140.34	29.78
200	8.92	1.12	141.34	6.00
500	43.82	8.09	210.54	22.22
1000	140.09	9.19	362.42	13.71
2000	636.77	120.50	960.40	142.62

Solution Times: SNIP10x10 K = 15

Sample size	Master avg	Stdev	Wall Clk	Stdev
50	14.27	9.85	120.63	16.02
100	41.97	27.80	192.32	95.78
200	94.87	38.54	219.90	38.66
500	461.90	289.87	658.39	289.67
1000	2758.29	1572.40	3023.90	1560.25
2000	22882.00	9674.37	23587.25	9502.51

Solution Times: SNIP10x10 $K = 20$

Sample size	Master avrg	Stdev	Wall Clk	Stdev
50	4.19	5.79	84.50	4.63
100	10.39	7.50	102.87	8.44
200	20.38	12.24	129.07	17.34
500	79.90	37.77	222.20	40.32
1000	232.96	59.24	445.44	72.14
2000	1615.82	531.72	2044.28	559.89

Conclusions

- Jeff didn't finish
- Continuing work: Solve bigger instances!
- Parallelize (grid-ify) IP master solve
- Thanks to XPRESS-SP
- Instances available
- Acknowledge grants