Stochastic Mixed-Integer Programming: Models

Suvrajeet Sen

Data-Driven Decisions (“3d”) Lab
IWSE Department
The Ohio State University
Columbus, OH

Overview of the Presentation

- A (B-C-D) Notation for SMIP
- Two-stage SMIP with Recourse
  - Stochastic Server Location Problem
  - Network Design under Threat
- Disjunctive Decomposition (D2)
- D2-BAC (D2 with Branch-and-Cut)
- Computational Results
- Comments and Conclusions

A (B-C-D) Notation for SMIP

- Two Stage Stochastic Linear Programming

$$\text{Min } \mathrm{c}^T x + E[f(x, \tilde{\omega})]$$

$$Ax = b, x \geq 0$$

where,

$$f(x, \omega) = \text{Min } \mathrm{g}^T y$$

$$Wy \geq r(\omega) - T(\omega)x$$

$$y \geq 0$$

Stochastic MIP with First Stage Integers

$$\text{Min } c^T x + E[f(x, \tilde{\omega})]$$
$$Ax \geq b,\ x \in R^{n_1} \times Z^{n_2}$$

where,

$$f(x, \omega) = \text{Min } g^T y$$
$$Wy \geq r(\omega) - T(\omega)x$$
$$y \in R^{n_3}$$

Z^n denotes integer vectors of length n. With second-stage integers, extremely difficult!

Stochastic Combinatorial Optimization

$$\text{Min } \ c^T x + E[f(x, \tilde{\omega})]$$
$$Ax \geq b, \ x \in \boldsymbol{B}^{n_1}$$

where,

$$f(x, \omega) \ = \ \text{Min } \ g^T y$$
$$Wy \geq r(\omega) - T(\omega)x$$
$$y \in \boldsymbol{R}^{n_2} \times \boldsymbol{B}^{n_3}$$

Here B^n denotes binary vectors of length n.

Many different structures for SMIP!

- Describing SMIP Problems
- B = Set of stages with Binary Vars.
- C = Set of stages with Continuous Vars.
- D = Set of stages with Discrete Vars. (arbitrary integers, not just binary)

- Louveaux has proposed a notation that covers all SP problems (e.g. notation includes whether random variables are cont/discrete
- Above notation helps clarify domain of applicability of results/algorithms etc.

- SLP: B = {∅}, C={1,2}, D ={ ∅}
- Wollmer, Norkin et al, Poojari/Mitra: B = {1}, C={1,2}, D ={1}
- Simple Integer Recourse: B = {2}, C={1}, D ={2} + structure of second stage
- Ahmed et al: B = {2}, C={1,2}, D ={2} + Fixed Tenders
- Multi-stage SMIPs: Caroe/Schultz, Roemisch et al, Alonso-Ayuso et al, Lulli/Sen: B = {1,2,…N}, C={1,2…N}, D ={1,2…N}

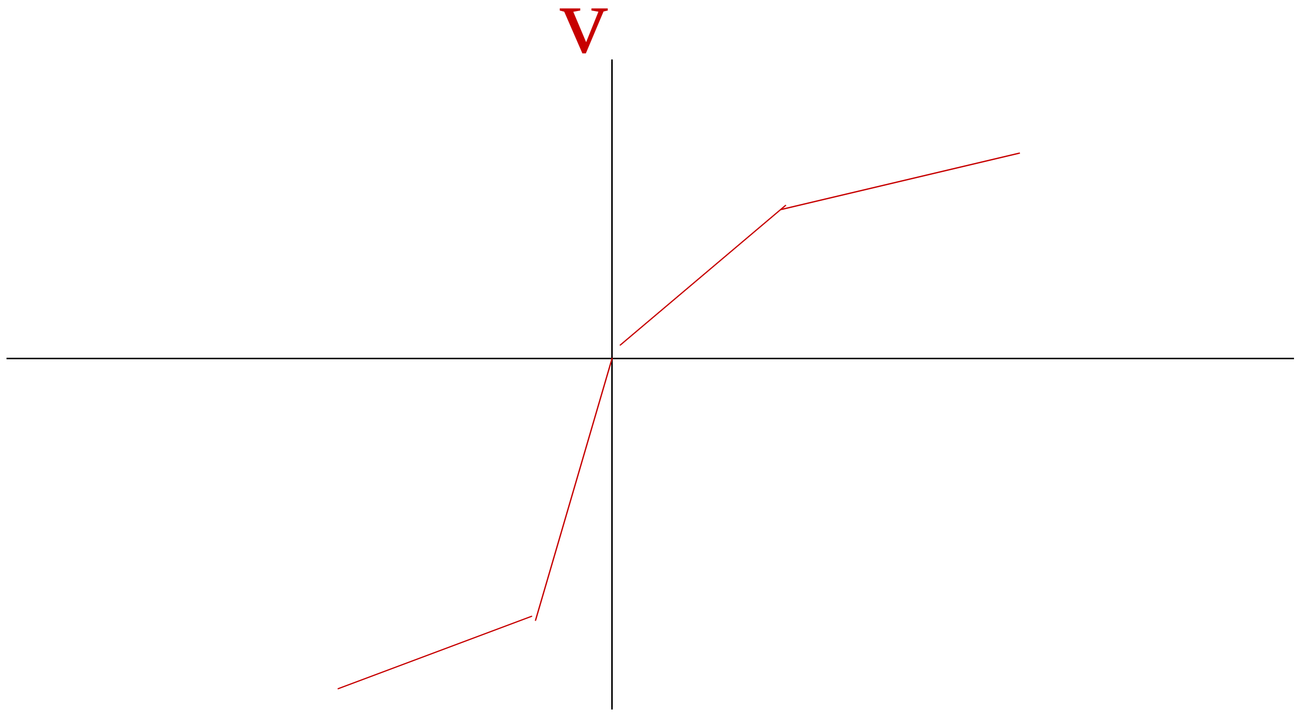
- Sherali/Fraticelli, Sen/Higle, Sen/Sherali
  $B = \{1,2\}$, $C=\{2\}$, $D =\{ \varnothing \}$
- Ntaimo/Sen $B = \{1,2\}$, $C=\{1,2\}$, $D =\{ \varnothing \}$

Two-stage SMIP with Recourse

- We have only stated models via "Expected Values"
- Is the reliance on "Expectation" a handi-cap?

- Of course! But many risk measures (e.g. down-side risk, mean absolute deviation etc.) can be re-formulated using expectation of a slightly modified, though mathemati-cally similar function.

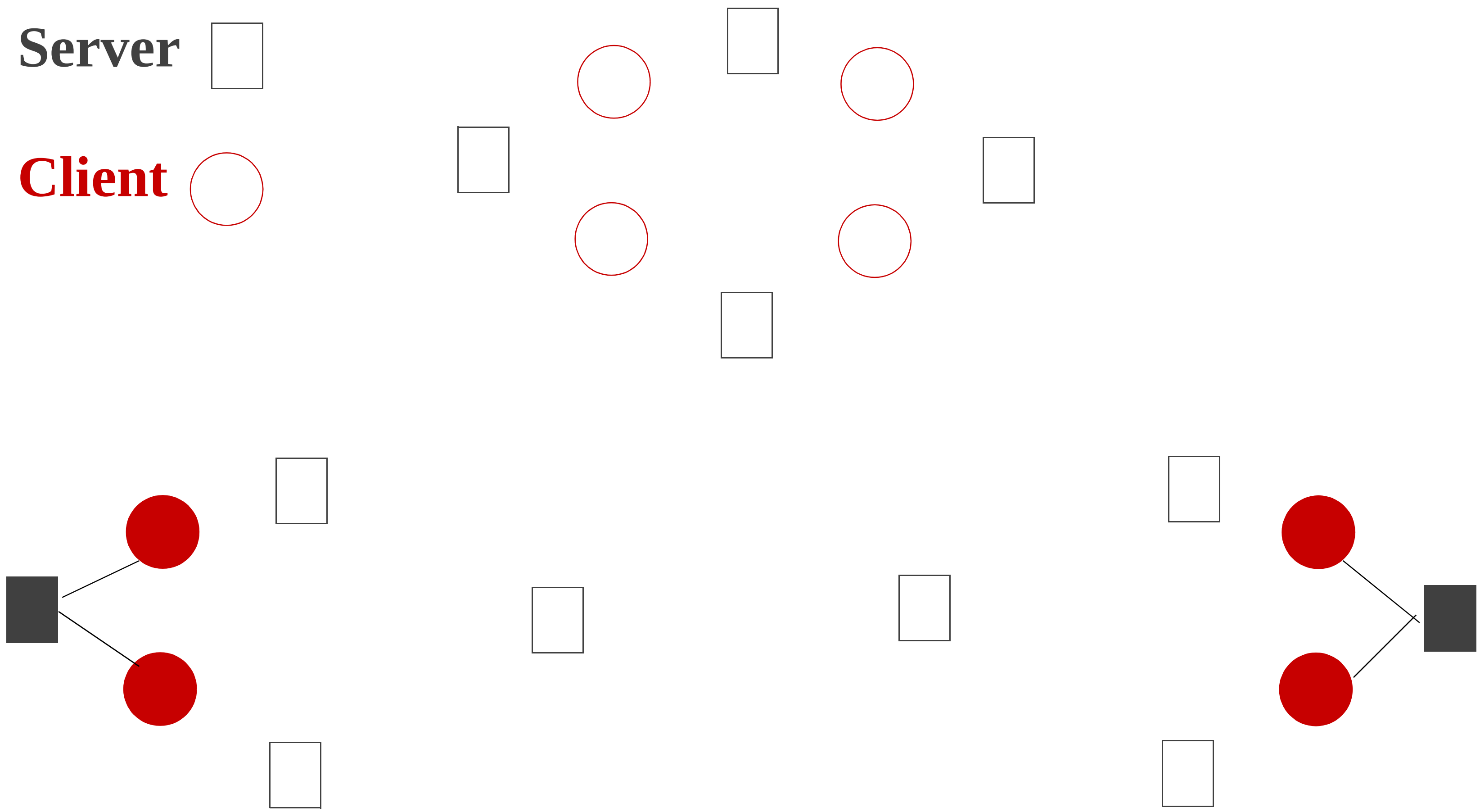
- Example: Kahneman/Tversky “S” curve for risk-aversion: Make first stage choice to Maximize weighted second-stage value

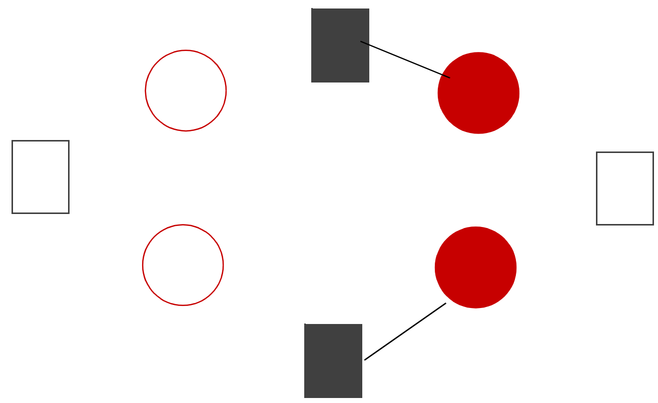


- Each piece has its own 0-1 variable

Stochastic Server Location Problem

Potential Server Locations (Choice) Client Sites (Random)





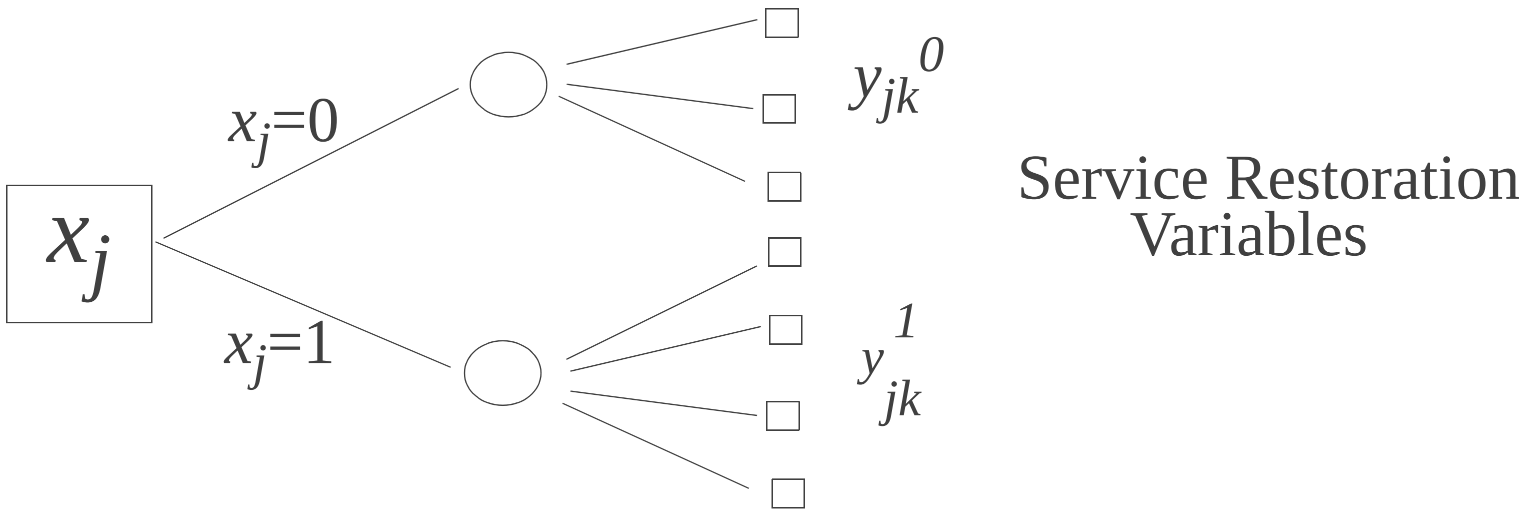
This SCO has two sets of decisions:

- choose server locations (e.g. bases)
- once demand nodes (e.g threats) appear, then assign servers to demand nodes

Network Reinforcement under Threat

- Setting of the Problem
  - Reinforce link of a network
  - Opponent threatens implemented network and some elements lose functionality
  - Design and implement a revised plan
- Opponent actions modeled by random variables predicting opponent's behavior
- Main distinction with previous model: Opponent actions (random variables) depend on implemented network

- Decision tree in SCO



- Logical conditions are as follows

$$y_{jk}^{0} \leq 1 - x_j$$
$$y_{jk}^{1} \leq x_j$$

Comments about Modeling

- Integer variables expand the scope of Stochastic Programming
- New applications in location, security, and many more such as
  - design optimization under uncertainty
  - supply chain
  - hybrid systems
- New decomposition algorithms are necessary because deterministic equivalent IP is too large in realistic models

Major Issue: Second Stage Integer Restrictions

- The function $F(x) = E[f(x, \tilde{\omega})]$ (i.e the recourse function) is no longer convex. Essentially, there aren't too many positive things to say about these functions ... they are (in general) nonconvex, discontinuous

Algorithmic strategies include

- Methods based on Computational Algebra (Schultz, Stougie and van der Vlerk)
- Lagrangian Relaxation (Caroe and Schultz)
- Lift-and-Project Cuts for the Deterministic Equivalent IP (Caroe and Tind)
- Global optimization (Ahmed, Tawarmalani, Sahinidis)

- Ranking/Selection (Statistics) (Sanchez and Wood)

- Our Work at Arizona/OSU
  - Set Convexification (Sen and Higle)
  - Branch -and-Cut (Sen and Sherali)
  - Branch-and-Price (Lulli and Sen)

Set Convexification of the Second Stage

Common Cut Coefficients (C^3) Theorem

Let $Y(x;\omega)$ denote the set of integer feasible solutions of the second stage MILP. Suppose that for some finite index set H we have

$$Y(x;\omega) \subseteq \cup_{h \in H} \{y \mid C_h y \geq d_h, \; y \geq 0\}$$

Let

$$S_h(x;\omega) = \{y \mid Wy \geq \omega - T(\omega)x, \; C_h y \geq d_h, \; y \geq 0\}$$

For a given pair $(\bar{x};\bar{\omega})$, suppose that

$$\pi^T y \geq \pi_0(\bar{x};\bar{\omega})$$

is a valid inequality for the following

$$S(\bar{x};\bar{\omega}) = \bigcup_{h \in H} S_h(\bar{x};\bar{\omega})$$

Then for any other pair $(x;\omega)$ there exists a finite scalar $\pi_0(x;\omega)$ such that the inequality

$$\pi^T y \geq \pi_0(x;\omega)$$

is valid for $S(x;\omega)$.

- Because of the C^3 theorem, adding cut coefficients to the second stage maintains the property that the technology matrix (of the second stage) remains independent of $(x;\omega)$. Hence in iteration k, we can represent the matrix as W^k.

- The right hand side, will have to be modified, depending upon the pair $(x;\omega)$.

- It turns out that $\pi_0(x;\omega)$ has the following form.

$$\pi_0(x;\omega) = \text{Min}_{h \in H}\{\bar{\nu}_h(\omega) - \bar{\gamma}_h(\omega)^T x\}$$

- Since this is a piecewise linear concave function, the following value function is nonconvex.

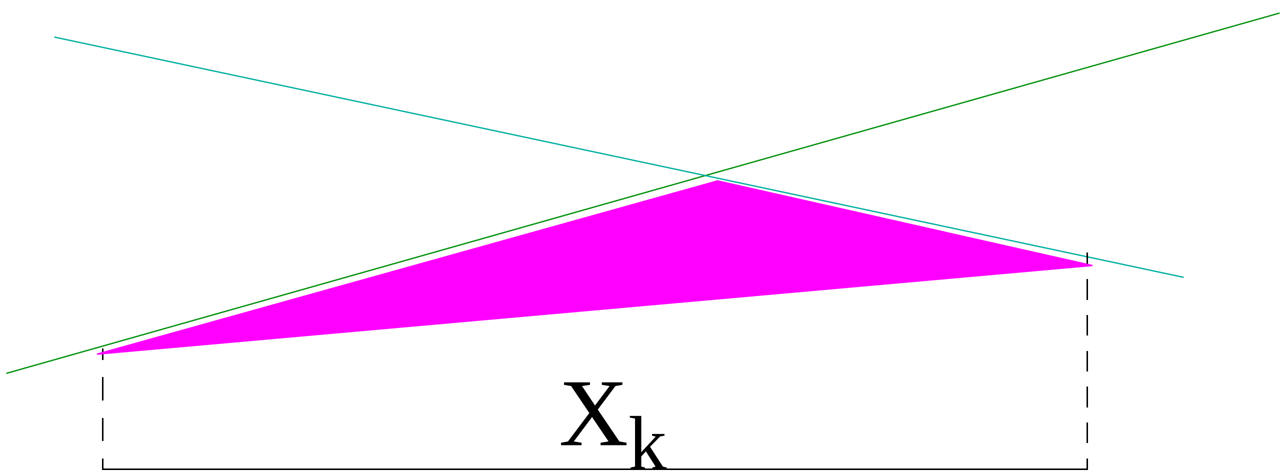
$$f_0(x, \omega) = \text{Min } g^T y$$

$$Wy \geq \omega - T(\omega)x$$

$$\pi^T y \geq \pi_0(x;\omega)$$

$$y \geq 0$$

We will convexify $\pi_0(x;\omega)$ by viewing its epigraph as a following disjunctive set such as the one shown below.



- Thus, we will write the constraints of second stage problem in the form

$$W^k y \geq \rho_c^k(x;\omega)$$

In summary Disjunctive Programming is used on both sides of the above inequality.

Hence, it makes sense to refer to it as the D^2 method

Convergence for 0-1 Stochastic MILP

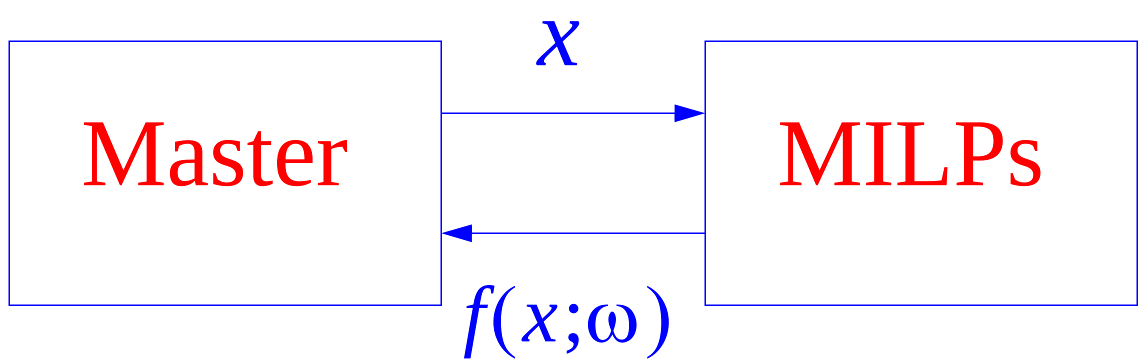
Assumptions

- Complete recourse
- All integer variables are 0-1
- Maintain all cuts in W^k
- Certain rules of order hold (a la lexico-graphic dual simplex in Gomory's proof)

Under these assumptions, the D^2 method results in a convergent algorithm.

Decomposition and Convexification

Both
Stages are
Hard



- Two Issues in Algorithm Design
  - Cuts for MILP (Disjunctive Cuts)
  - Approximation of f (also convexification)

Decomposition for SMIP

- Deterministic Equivalent Problem (DEP) solved with CPLEX
  - ILOG Corporation [2002]
- Disjunctive Decomposition (D^2) Algorithm
  - Sen and Higle [2005]
- Disjunctive Decomposition with Branch-and-Cut (D2-BAC)
  - Sen and Sherali [2006]
- For computational results
  - Thanks to Lewis Ntaimo and Yang Yuan

Computational Results

- **Instances**
  - Stochastic Server Location Problems (SSLP)
    - Ntaimo and Sen [2005]
  - Strategic Supply Chain Planning (SSCh)
    - Alonso-Ayuso et al [2003]
- **Platform:**
  - CPLEX 7.0 running on Sun 280R with 2 UltraSPARC-III+ CPUs running at 900MHz
- **CPLEX Settings for DEP using CPLEX:** (Best CPU Time)
  - "set mip emphasis 1": emphasizes looking for feasible solutions
  - "set mip strategy start 4": uses barrier at the root
  - "branching priority order on x"

The Case for Decomposition

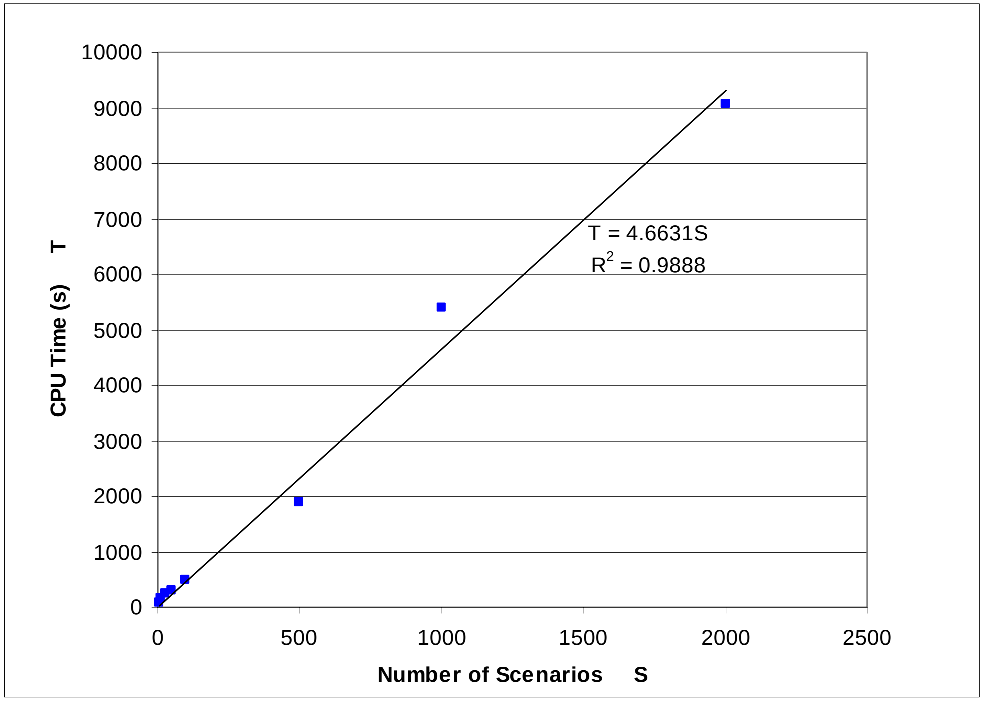
Comparison of D2 with CPLEX (3 replications for each instance)

Instance	*D2* Iters Avg	*D2* Cuts Avg	*D2* CPU (secs) Min	Max	Avg	DEP using CPLEX 7. CPU (secs)	Avg Gap
5.25.50	20.60	10.00	0.53	0.82	0.73	4.58	
5.25.100	20.60	13.00	1.03	1.84	1.48	14.69	
5.50.50	25.00	5.80	0.68	1.64	0.98	10.35	
5.50.100	25.60	10.60	1.25	3.95	1.89	33.25	
10.50.50	233.40	229.40	138.71	295.95	228.89	>10,000	0.44%
10.50.100	243.00	240.60	228.68	480.00	318.12	>10,000	9.02%
10.50.500	298.40	297.20	1616.12	1902.20	1753.88	>10,000	38.17%
10.50.1000	298.40	297.20	3307.67	5410.10	3948.13	>10,000	99.60%
10.50.2000	308.60	307.60	8530.37	9571.04	8975.60	>10,000	46.24%
15.45.5	147.00	136.50	58.94	181.53	119.19	>10,000	1.19%
15.45.10	303.40	297.20	1306.46	2988.65	1930.00	>10,000	0.27%
15.45.15	739.33	738.33	5244.14	7210.63	6208.23	>10,000	0.72%

m.n.s = servers.customers.scenarios

Scalability

- Linear in the number of scenarios



*D*2 CPU time for SSLP_10_50 with increasing scenarios

Strategic Supply Chain Problems

- Alonso-Ayuso et al [2003]
- Approx. 1000 binary & 65,000 continuous
- Maximization Problems

Instance	D2 Value	B&F Value	D2 Iters	D2 Cuts	D2 CPU secs	Gap
C1	184439	178366	184	177	4558	4.139%
C2	0	0	68	57	1342	0%
C3	230268	224564	92	85	1179	4.461%
C4	201454	197487	160	149	3265	4.07%
C6	231368	226578	114	109	1642	4.65%
C8	100523	89607	186	180	9650	3.234%
C10	139738	139738	87	81	1083	0%

Performance of D2-BAC

Instance	*BAC* Iters Avg	*BAC* Cuts Avg	Nodes Avg	*BAC* CPU Time (secs) Min	Max	Avg
5.25.50	21.4	11.6	601.6	0.68	1.08	0.81
5.25.100	21.8	14.4	1506.4	1.13	1.96	1.70
5.50.50	24.0	4.2	210.4	0.59	0.96	0.75
5.50.100	23.6	5.6	560.4	1.15	1.52	1.34
10.50.50	212.7	201.3	14053.3	169.66	413.25	327.25
10.50.100	229.5	225.3	25989.5	232.65	861.01	418.88
10.50.500	257.5	251.5	169859.5	1077.14	5960.12	2325.21
10.50.1000	236.8	234.8	235140.5	2356.96	2584.76	2473.09
10.50.2000	147.0	147.0	617898.0	>10,000	>10,000	>10,000
15.45.5	233.5	197.5	2267.5	162.53	544.54	353.54
15.45.10	211.5	193.0	3343.0	172.83	2217.00	1194.92
15.45.15	325.3	316.7	5414.0	157.85	3327.16	1609.73
15.45.5	147.00	136.50		58.94	181.53	119.19
15.45.10	303.40	297.20		1306.46	2988.65	1930.00
15.45.15	739.33	738.33		5244.14	7210.63	6208.23

m.n.s = servers.customers.scenarios

Most Recent Implementation

(D2-BAC++ uses special Cut Generation LP)

Instance	D2	D2-BAC	D2-BAC++
5.25.50	1.64	0.70	0.36
5.25.100	2.15	1.73	0.89
5.50.100	7.10	3.70	1.56
5.50.500	34.50	23.05	12.36
5.50.1000	140.47	64.17	22.77
5.50.2000	603.37	274.40	42.74
10.50.50	295.95	373.98	262.13
10.50.100	396.76	452.31	486.99
10.50.500	1902.2	2772.22	1313.38
10.50.1000	5410.1	5677.80	2139.47
10.50.2000	9055.29	> 10800	3916.47
15.45.5	110.34	232.30	211.79
15.45.10	1494.89	222.41	153.41
15.45.15	7210.63	1988.26	803.56