School of Mathematics

Interior Point Warmstarts Applied to Stochastic Programming

Marco Colombo, Jacek Gondzio, Andreas Grothey

Stochastic Programming Warmstarts: Overview

- Review of Interior Point Methods
- Warmstarting Interior Point Methods
- Stochastic Programming
- IPM warmstart applied to Stochastic Programming
- Numerical Results

Interior Point Methods (for LP)

$$\min c^\top x \qquad \text{s.t.} \quad \begin{aligned} Ax &= b \\ x &\geq 0 \end{aligned} \tag{LP}$$

Optimality conditions:

$$\begin{aligned} c - A^\top y - s &= 0 \\ Ax &= b \\ XSe &= 0 \\ x, s &\geq 0 \end{aligned} \tag{KKT}$$

Interior Point Methods (for LP)

$$\min c^\top x - \mu \sum \ln x_i \quad \text{s.t.} \quad \begin{aligned} Ax &= b \\ x &\geq 0 \end{aligned} \tag{LP}$$

Optimality conditions:

$$\begin{aligned} c - A^\top y - s &= 0 \\ Ax &= b \\ XSe &= \mu e \\ x, s &\geq 0 \end{aligned} \tag{KKT}$$

Interior Point Methods (for LP)

$$\min c^\top x - \mu \sum \ln x_i \quad \text{s.t.} \quad \begin{aligned} Ax &= b \\ x &\geq 0 \end{aligned} \tag{LP}$$

Optimality conditions:

$$\begin{aligned} c - A^\top y - s &= 0 \\ Ax &= b \\ XSe &= \mu e \\ x, s &\geq 0 \end{aligned} \tag{KKT}$$

Central Path:

The set of all solutions to the optimality conditions for $\mu > 0$.

The central path joins the analytic center (for $\mu = \infty$) with the LP solution (for $\mu = 0$).

Neighbourhoods (of the central path)

$$\begin{aligned} \mathcal{N}_2(\theta) &:= \{(x,y,s) \in \mathcal{F}^0 : \|XSe - \mu e\|_2 \leq \theta\mu\} \\ \mathcal{N}_{-\infty}(\gamma) &:= \{(x,y,s) \in \mathcal{F}^0 : x_i s_i \geq \gamma\mu\} \end{aligned}$$

where $\mathcal{F}^0 := \{(x,y,s) : c - A^\top y - s = 0, Ax = b, x, s > 0\}$.

Path Following Methods

- choose $x_0, y_0, s_0 > 0, \mu = x_0^\top s_0/n$
- compute Newton step $(\Delta x, \Delta s, \Delta y)$ for (KKT) and given $\mu^+ < \mu$.
- compute stepsizes

$$\alpha = \max_{\alpha > 0}\{\alpha : x + \alpha\Delta x \geq 0, s + \Delta s \geq 0, (x, s) \in \mathcal{N}_*(\tau)\}$$

- take step

$$\begin{aligned} x_+ &= x + 0.995\alpha\Delta x \\ y_+ &= y + 0.995\alpha\Delta y \\ s_+ &= z + 0.995\alpha\Delta s \end{aligned}$$

- update μ:

$$\mu_+ = \sigma\frac{x_+^\top s_+}{n}, \quad 0 < \sigma < 1$$

Warmstarting Interior Point Methods

- Many applications require the solution of a series of optimization problems (SQP, B&B, MPC, etc)
- Simplex/Active Set Methods are easy and efficient to warmstart
- IPM Warmstart is seen as difficult to impossible
- One of the main arguments against IPM

Warmstarting Interior Point Methods

- Many applications require the solution of a series of optimization problems (SQP, B&B, MPC, etc)
- Simplex/Active Set Methods are easy and efficient to warmstart
- IPM Warmstart is seen as difficult to impossible
- One of the main arguments against IPM

Some research (among others)

- Mitchell, Todd '92
- Hippolito '93
- Lustig, Marsten, Shanno '94
- Gondzio '98
- Gondzio, Vial '99
- Yıldırım, Wright '02
- Gondzio, G. '03
- John,Yıldırım '06
- Benson,Shanno '06

⇒ Message:

- Warmstarting IPMs is possible! (Can save around 50%-60% of iterations)

Warmstarting Interior Point Methods

Aim: Use information from solution process of

$$\min c^\top x \quad \text{s.t.} \quad \begin{aligned} Ax &= b \\ x &\geq 0 \end{aligned} \tag{LP}$$

to construct a starting point for (nearby problem)

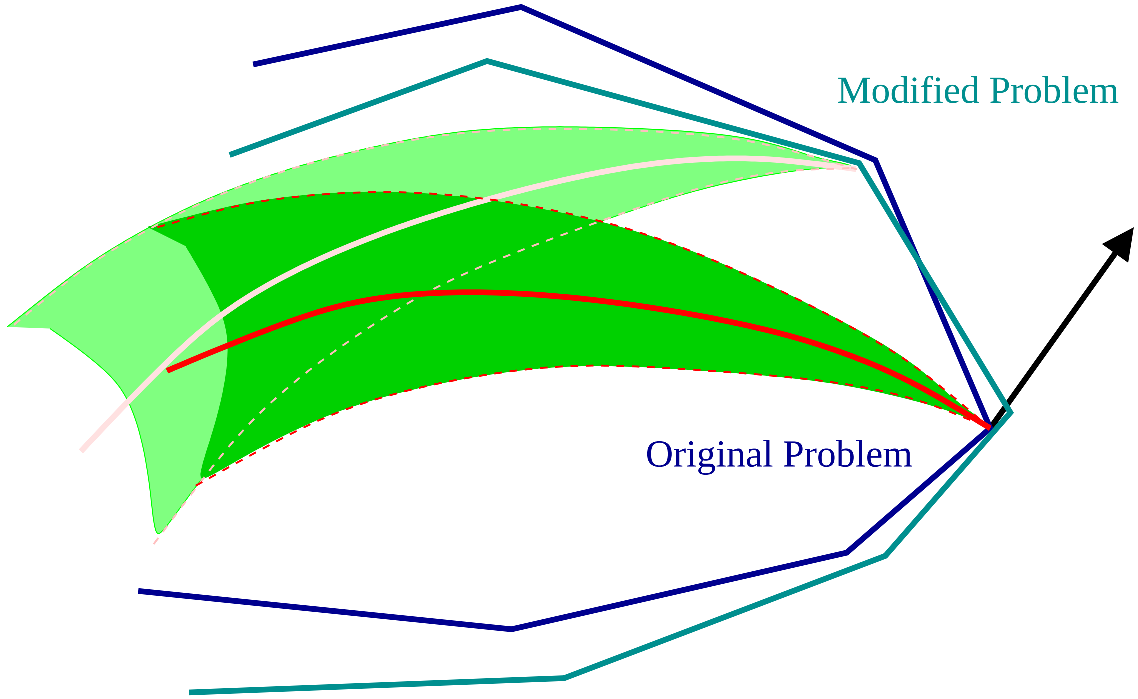
$$\min \tilde{c}^\top x \quad \text{s.t.} \quad \begin{aligned} \tilde{A}x &= \tilde{b} \\ x &\geq 0 \end{aligned} \tag{$\widetilde{\text{LP}}$}$$

where $\tilde{A} \approx A, \tilde{b} \approx b, \tilde{c} \approx c$

- It is **not** a good idea to use the solution of (LP) to start ($\widetilde{\text{LP}}$).
- *Unlike for the Simplex/Active Set Method!*

Why?

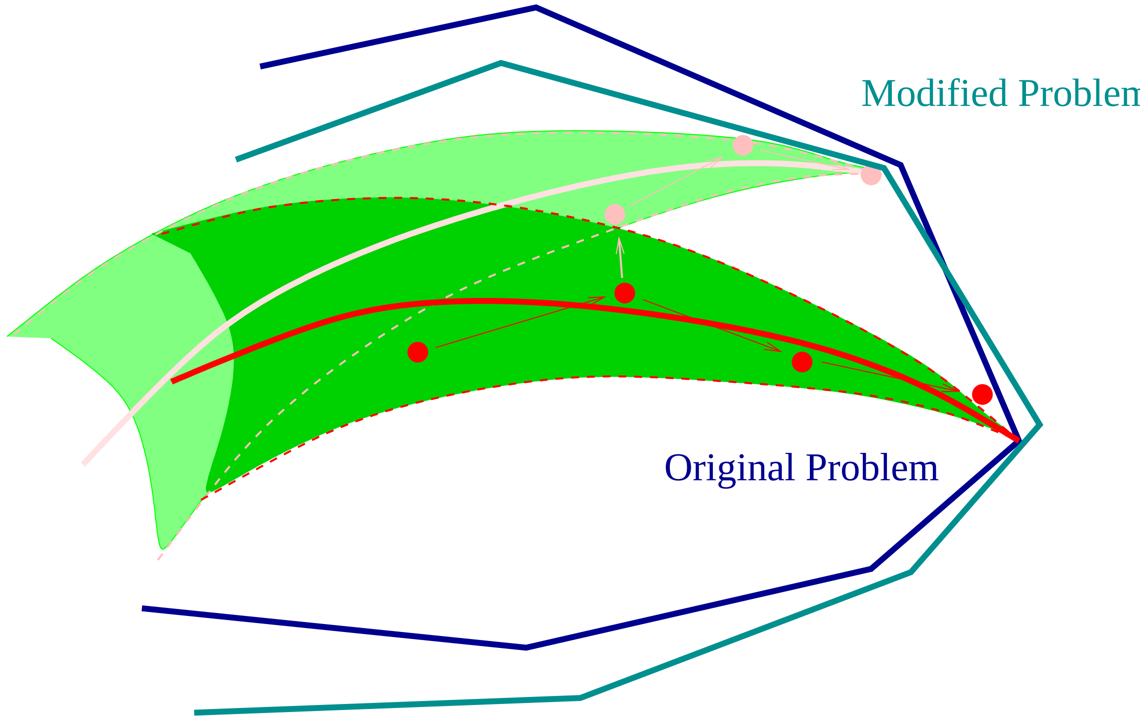
Hippolito (1993): Search direction is parallel to nearby constraints



⇒ only small step in search direction can be taken

Warmstarting Heuristics

Idea: Start close to the (new) central path, not close to the (old) solution



⇒ Start from a previous iterate and do additional *modification* step.

Interior Point Warmstarts: Theoretical Results

A typical warmstart results is (Assume $\tilde{A} = A$):

Lemma. Let $(x, y, s) \in \mathcal{N}_{-\infty}(\gamma_0)$ for problem (LP) then the full Newton step $(\Delta x, \Delta y, \Delta s)$ in the perturbed problem ($\tilde{\text{LP}}$) is feasible and

$$(x + \Delta x, y + \Delta y, s + \Delta s) \in \tilde{\mathcal{N}}_{-\infty}(\gamma)$$

provided that

$$\delta_{bc}^{GG} \leq \frac{\gamma_0}{B^{\infty}(1 + 1/\gamma)}\mu.$$

Here

$$\delta_{bc}^{GG} := \|\xi_c\|_2 + \|A^T(AA^T)^{-1}\xi_b\|_2, \qquad \xi_b = \tilde{b} - Ax, \quad \xi_c = \tilde{c} - A^T y - s$$

is the orthogonal distance of (x, y, s) from feasibility in the perturbed problem

Numerical Experiments: NETLIB

Testset proposed by Benson, Shanno ’06:

- Take small to mid-scale instances of NETLIB LP library
- Randomly perturb problem data in b, c or A
- Perturb 10% (at most 20) of components on average.
- Perturb by $0.001, 0.01, 0.1$.

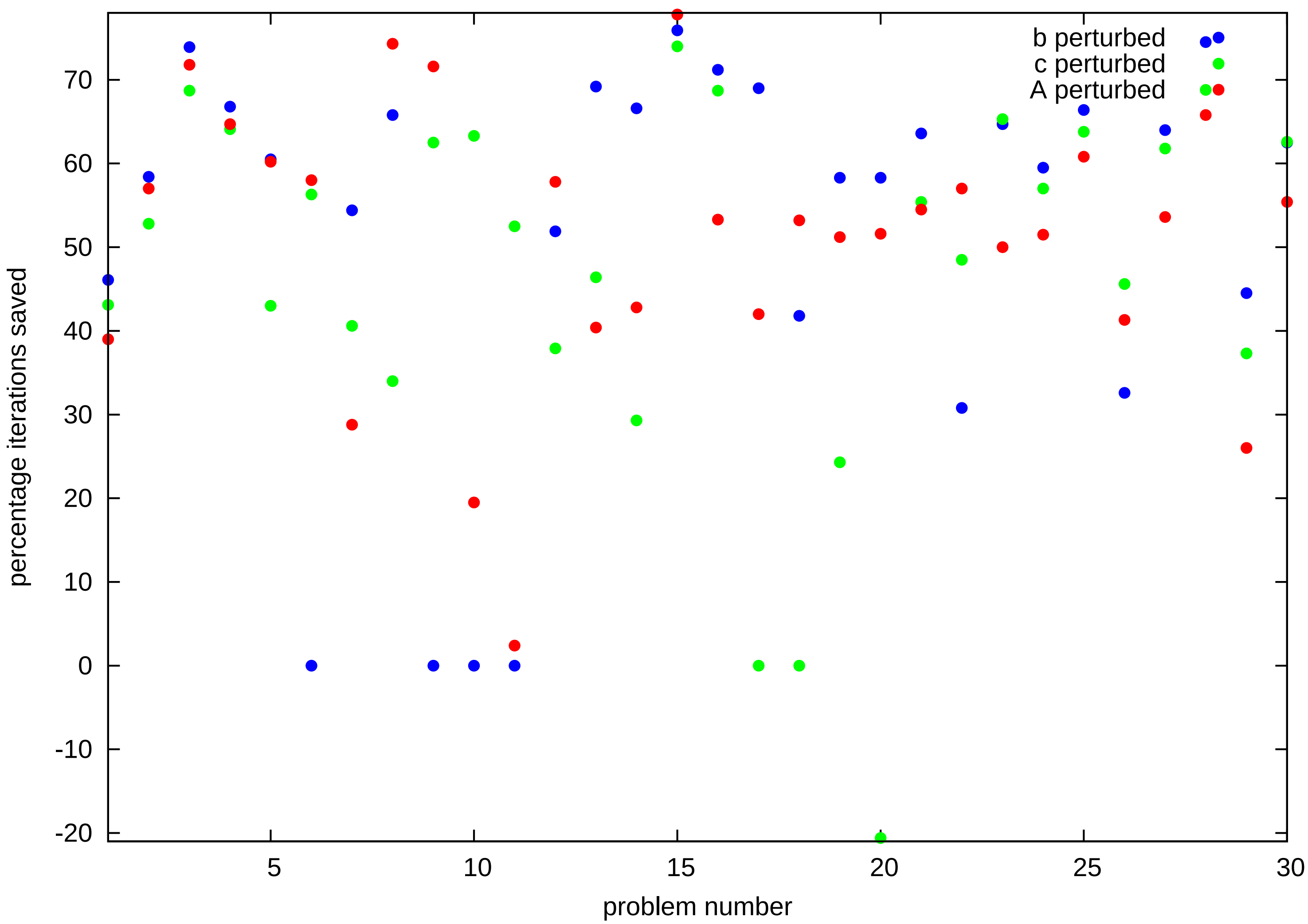
Our results:

- Choose 10 random instances (and take average)
- All problems warmstarted with $\hat{\mu} = 10^{-2}$.
- All test run within the OOPS/HOPDM IPM solver

Results (Best Warmstart) - all perturbations

Problem	b			c			A		
	cold	warm	per	cold	warm	perc	cold	warm	per
ADLITTLE	10.4	5.6	46.1	10.2	5.8	43.1	10.5	6.4	39.0
AFIRO	10.1	4.2	58.4	10.4	4.9	52.8	10.0	4.3	57.0
AGG2	16.1	4.2	73.9	16.3	5.1	68.7	16.0	4.5	71.8
AGG3	15.7	5.2	66.8	15.9	5.7	64.1	15.6	5.5	64.7
BANDM	13.7	5.4	60.5	13.7	7.8	43.0	13.6	5.4	60.2
BEACONFD	-	-	-	10.3	4.5	56.3	10.0	4.2	58.0
BLEND	9.0	4.1	54.4	9.1	5.4	40.6	9.0	6.4	28.8
BOEING1	19.9	6.8	65.8	19.4	12.8	34.0	19.5	5.0	74.3
BORE3D	-	-	-	13.1	4.9	62.5	14.1	4.0	71.6
BRANDY	-	-	-	15.3	5.6	63.3	20.0	16.1	19.5
DEGEN2	-	-	-	9.9	4.7	52.5	44.4	43.3	2.4
E226	15.6	7.5	51.9	15.3	9.5	37.9	15.2	6.4	57.8
GROW15	13.0	4.0	69.2	21.1	11.3	46.4	16.3	9.7	40.4
GROW7	12.0	4.0	66.6	20.8	14.7	29.3	18.2	10.4	42.8
ISRAEL	20.4	4.9	75.9	20.8	5.4	74.0	19.9	4.4	77.8
KB2	17.4	5.0	71.2	17.9	5.6	68.7	18.0	8.4	53.3
LOTFI	19.7	6.1	69.0	21.4	21.4	0.0	31.9	18.5	42.0
RECIPELP	14.1	8.2	41.8	-	-	-	13.9	6.5	53.2
SC105	12.0	5.0	58.3	11.5	8.7	24.3	11.7	5.7	51.2
SC205	12.0	5.0	58.3	12.6	15.2	-20.6	12.2	5.9	51.6
SC50A	11.0	4.0	63.6	11.0	4.9	55.4	11.0	5.0	54.5
SC50B	10.7	7.4	30.8	10.1	5.2	48.5	10.0	4.3	57.0
SCAGR25	12.2	4.3	64.7	12.4	4.3	65.3	12.0	6.0	50.0
SCAGR7	9.9	4.0	59.5	10.0	4.3	57.0	9.9	4.8	51.5
SCFXM1	14.6	4.9	66.4	14.1	5.1	63.8	14.8	5.8	60.8
SCSD1	10.1	6.8	32.6	9.2	5.0	45.6	9.2	5.4	41.3
SCTAP1	15.0	5.4	64.0	16.0	6.1	61.8	14.9	6.9	53.6
SHARE1B	21.2	5.4	74.5	21.8	6.8	68.8	21.4	7.3	65.8
SHARE2B	9.2	5.1	44.5	9.1	5.7	37.3	9.2	6.8	26.0
STOCFOR1	13.9	5.2	62.5	13.4	5.0	62.6	11.9	5.3	55.4
Average	13.8	5.3	59.6	14.2	7.3	48.4	15.5	8.0	50.9

Results for LP problems (NETLIB)



Stochastic Programming:

A Stochastic Programming Problem is given by

$$\begin{array}{rl} \min & c_1^T x_1 + \mathbb{E}_\xi[c_2(\xi)^T x_2(\xi)] \\ \text{s.t.} & W_1 x_1 = h_1, \\ & T_2(\xi)x_1 + W_2(\xi)x_2(\xi) = h_2(\xi) \quad \text{a.s.} \\ & x_1 \geq 0, x_2(\xi) \geq 0 \end{array}$$

This models a decision process $x_1 \rightarrow \xi \rightarrow x_2(\xi)$

Stochastic Programming:

A Stochastic Programming Problem is given by

$$\begin{array}{rl} \min & c_1^T x_1 + \mathbb{E}_\xi[c_2(\xi)^T x_2(\xi)] \\ \text{s.t.} & W_1 x_1 = h_1, \\ & T_2(\xi) x_1 + W_2(\xi) x_2(\xi) = h_2(\xi) \quad \text{a.s.} \\ & x_1 \geq 0, x_2(\xi) \geq 0 \end{array}$$

This models a decision process $x_1 \rightarrow \xi \rightarrow x_2(\xi)$

Multistage Stochastic Programming:

$$\begin{array}{rl} \min & c_1^T x_1 + \sum_{t=2}^T \mathbb{E}_{\xi_t}[c_t(\xi_t)^T x_t(\xi_t)] \\ \text{s.t.} & W_1 x_1 = h_1, \\ & T_t(\xi_t) x_{t-1}(\xi_{t-1}) + W_t(\xi_t) x_t(\xi_t) = h_t(\xi_t) \quad \text{a.s.} \\ & x_1 \geq 0, x_2(\xi) \geq 0 \end{array} \tag{P(ξ)}$$

Multistage stochastic programming models a decision process:

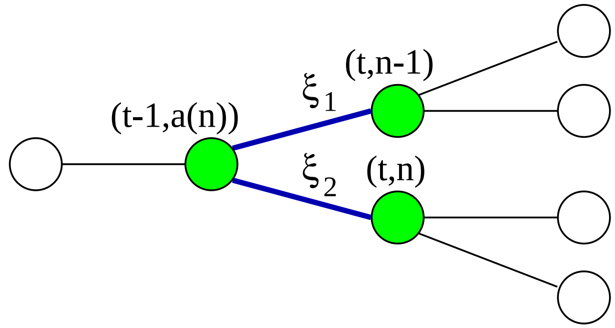
$$x_1 \rightarrow \xi_2 \rightarrow x_2(\xi_2) \rightarrow \ldots \rightarrow \xi_T \rightarrow x_T(\xi_T)$$

Stochastic Programming: Deterministic equivalent

Introduce discretizations:

$$\xi_t = \{\xi_t^l : P(\xi_t = \xi_t^l) = p_t^l\}$$

corresponding to a scenario tree $\mathcal{T}$.



$$\Rightarrow$$

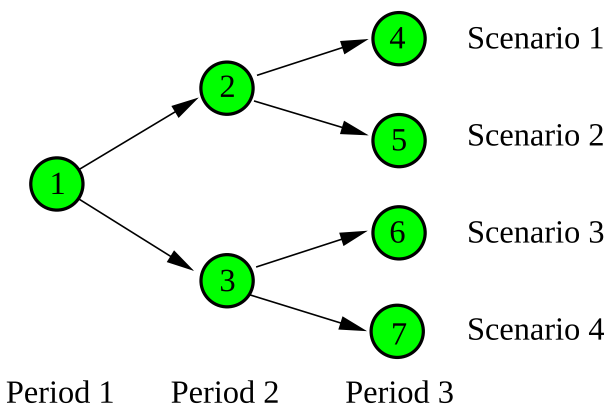
Deterministic equivalent

$$\begin{array}{ll} \min & c_1^T x_1 + \sum_{t=2}^{T} \sum_{l \in \mathcal{L}_t} p_t^l (c_t^l)^T x_t^l \\ \text{s.t.} & W_1 x_1 = h_1, \\ & T_t^l x_{t-1}^{a(l)} + W_t^l x_t^l = h_t^l \\ & x_t^l \geq 0 \end{array}$$

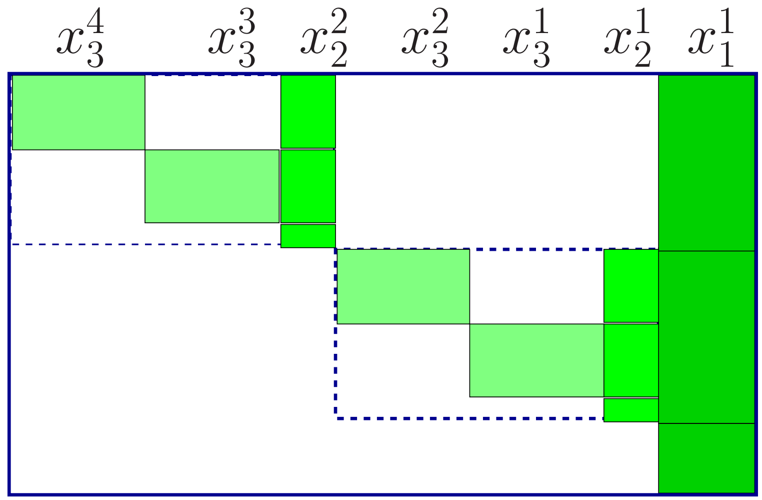
where

$$W_t^l = W_t(\xi_t^l), T_t^l = T_t(\xi_t^l), c_t^l = c_t(\xi_t^l), h_t^l = h_t(\xi_t^l),$$

Multistage Stochastic Programming



Scenario Tree



Constraint Matrix

Symmetrical event tree with K realizations/node and T periods corresponds to

$$K^{T-1} \text{ scenarios} \qquad \frac{K^T - 1}{K - 1} \text{ nodes (blocks)}$$

Applications

(Multistage) Stochastic Programming has many applications

- Portfolio Optimization
- Planning under uncertainty
- Electricity Generation Planning (involving hydro or wind)
- etc

Solving Stochastic Programming Problems:

Problem: size of deterministic equivalent quickly becomes very large
⇒Difficult for standard solvers. Suitable approaches are:

- Decomposition (Benders, L-shaped method)
- Interior Point Method
  (*Gondzio, G. (2005): solved multistage SP problem with* 10^9 *variables on 1280 processors in under 2h*)

Stochastic Programming Warmstarts

Stochastic Programming Warmstarts

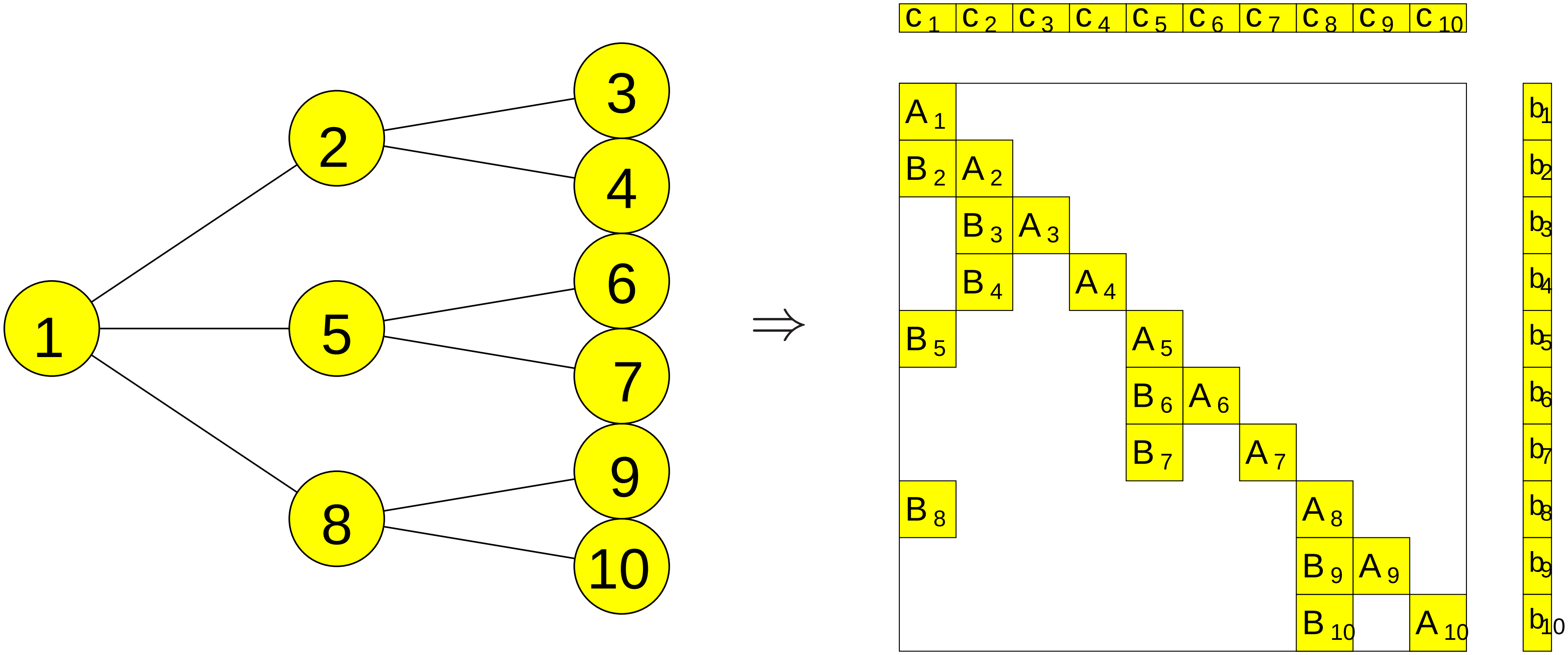
- **Idea:** speed up solution process by crash-starting from a smaller tree

Stochastic Programming Warmstarts

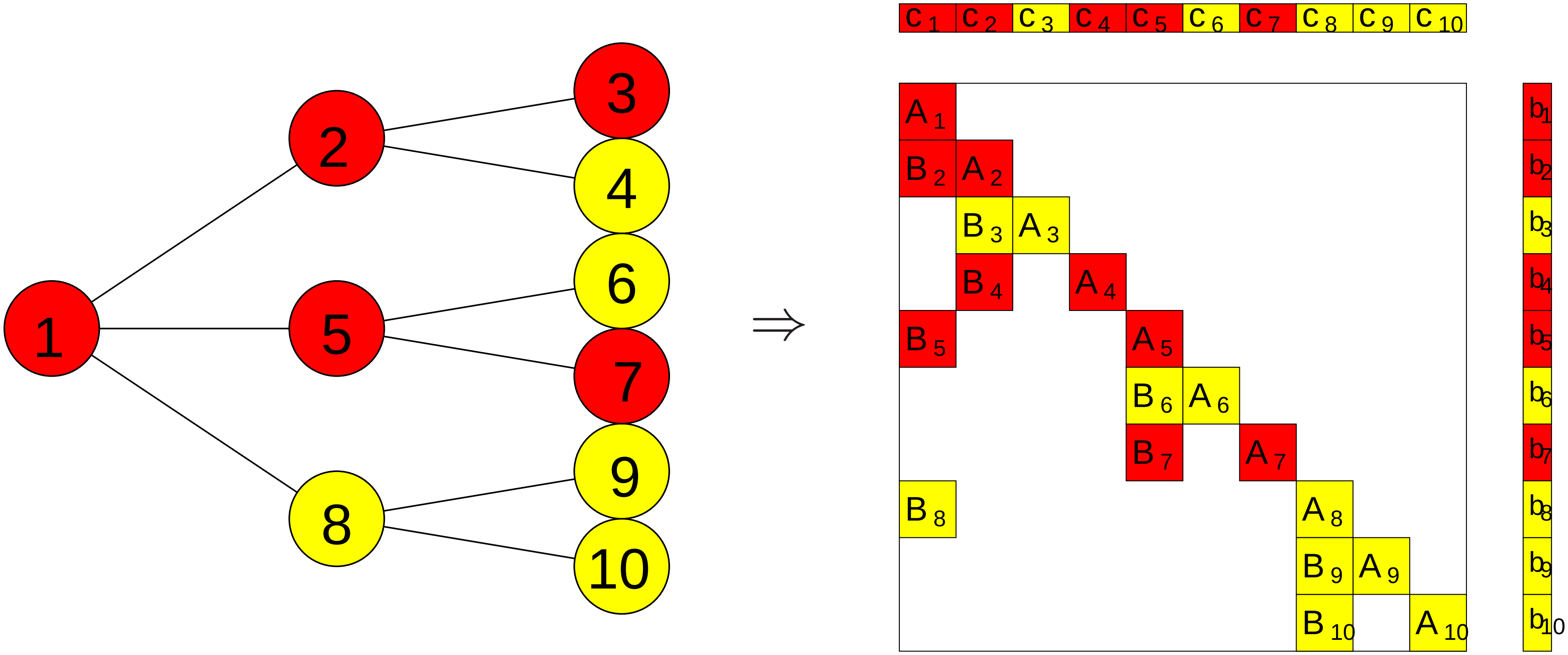
- **Idea:** speed up solution process by crash-starting from a smaller tree

- **But:** IPMs are notoriously bad at exploiting a known starting point

Stochastic Programming Warmstarts

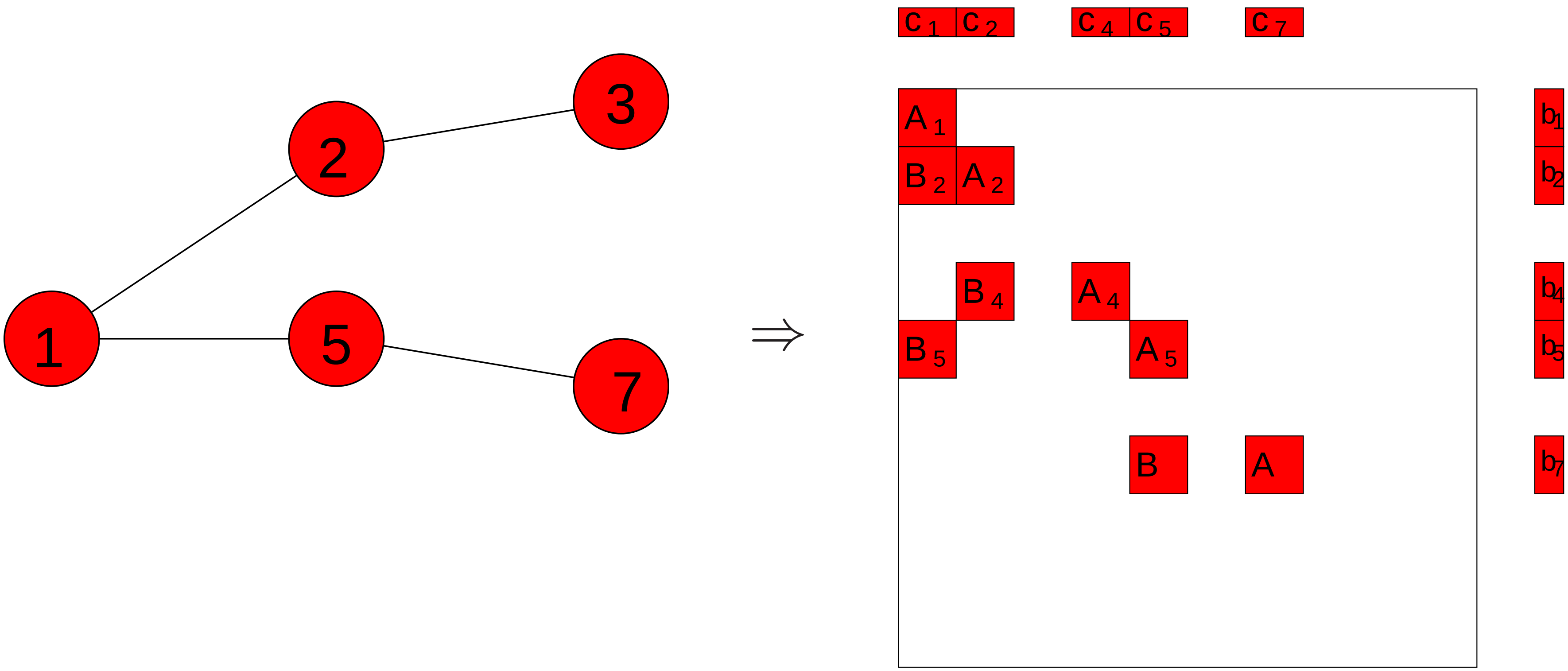


Stochastic Programming Warmstarts



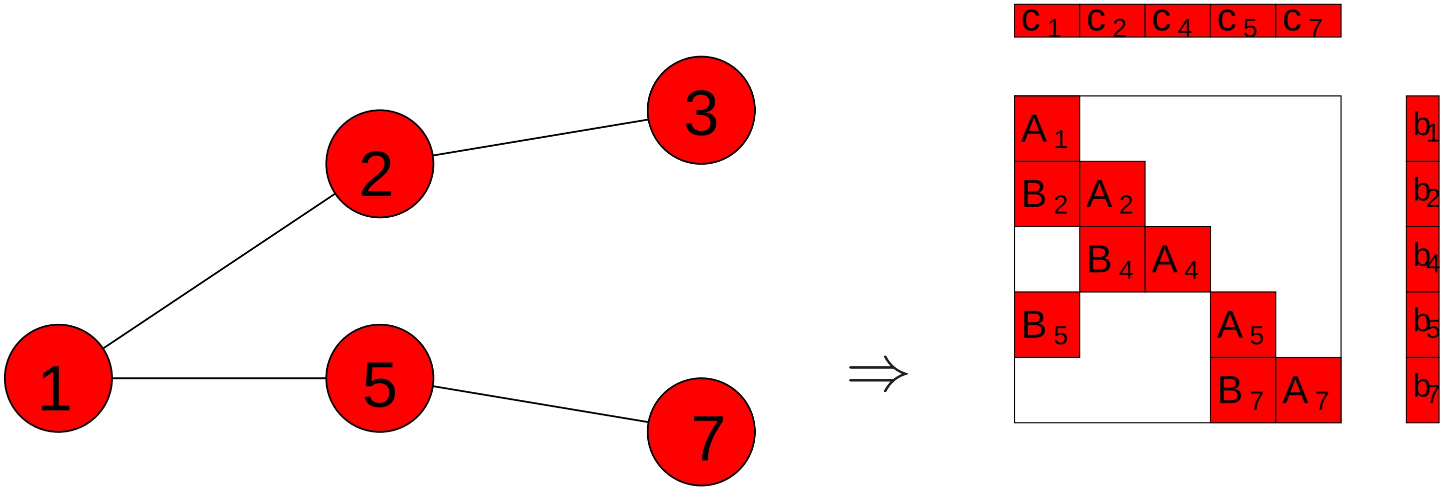
- Select sample scenarios

Stochastic Programming Warmstarts



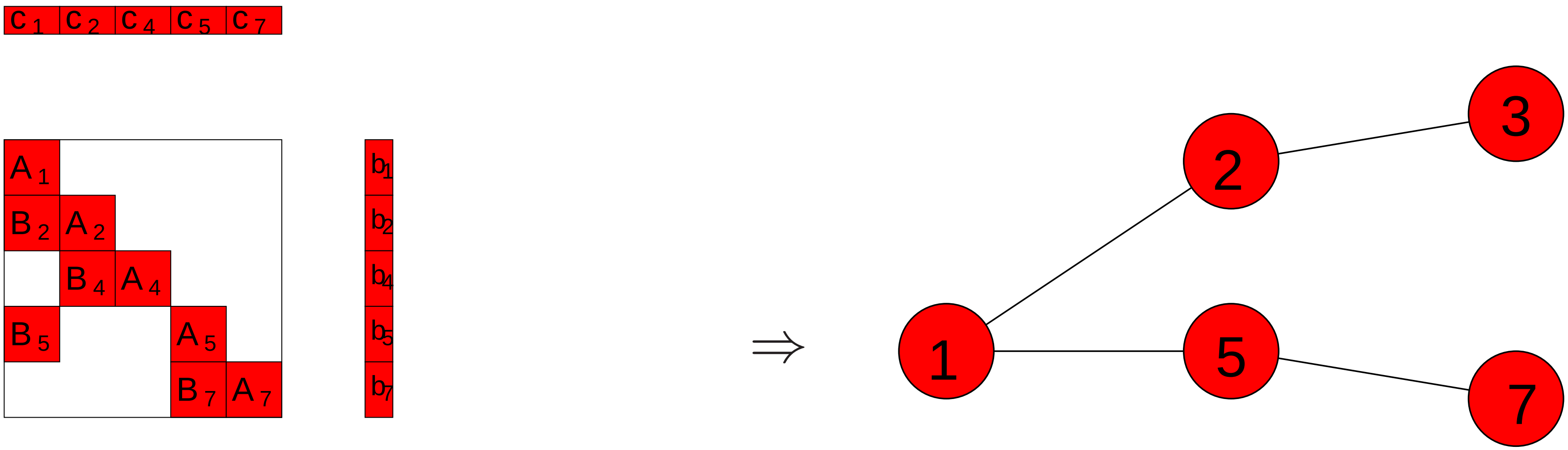
- Select sample scenarios
- Aggregate Scenarios/Reduce Problem

Stochastic Programming Warmstarts



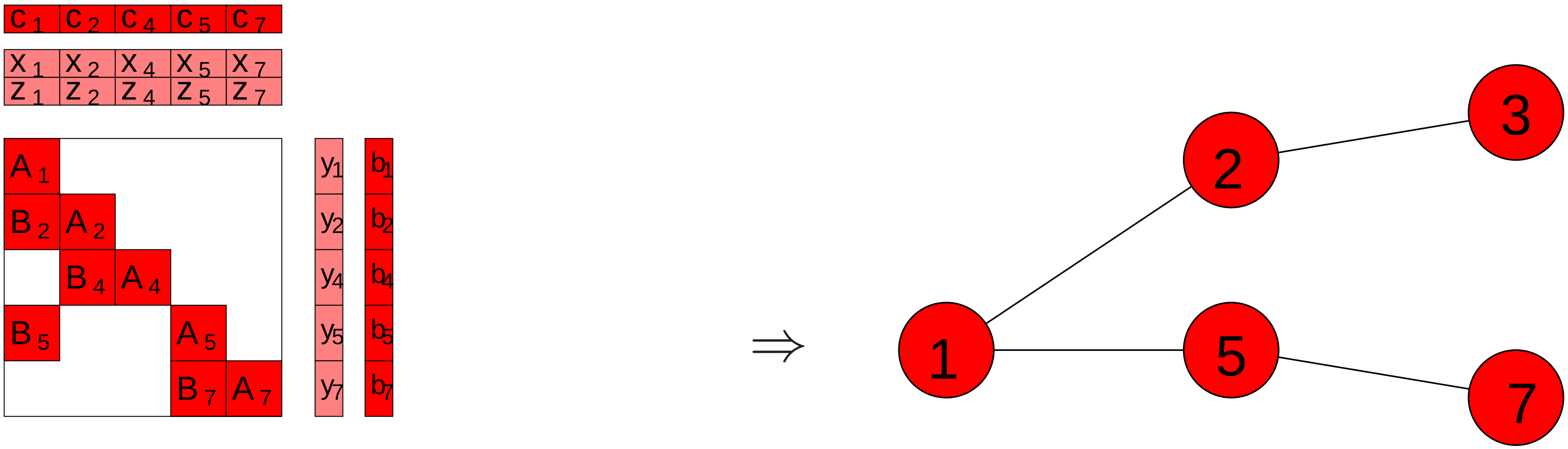
- Select sample scenarios
- Aggregate Scenarios/Reduce Problem

Stochastic Programming Warmstarts



To solve the problem by warmstarting, reverse the process

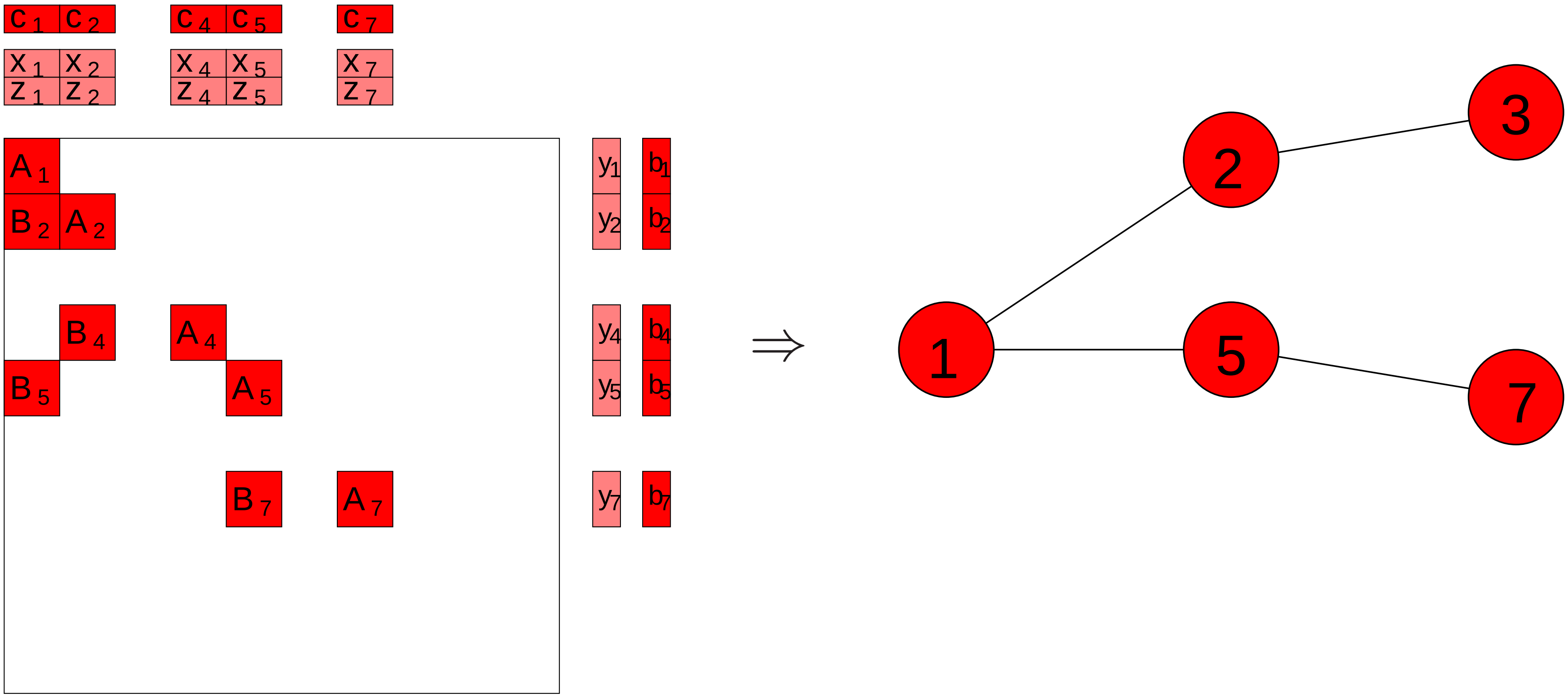
Stochastic Programming Warmstarts



To solve the problem by warmstarting, reverse the process

- Solve reduced problem (to low accuracy)

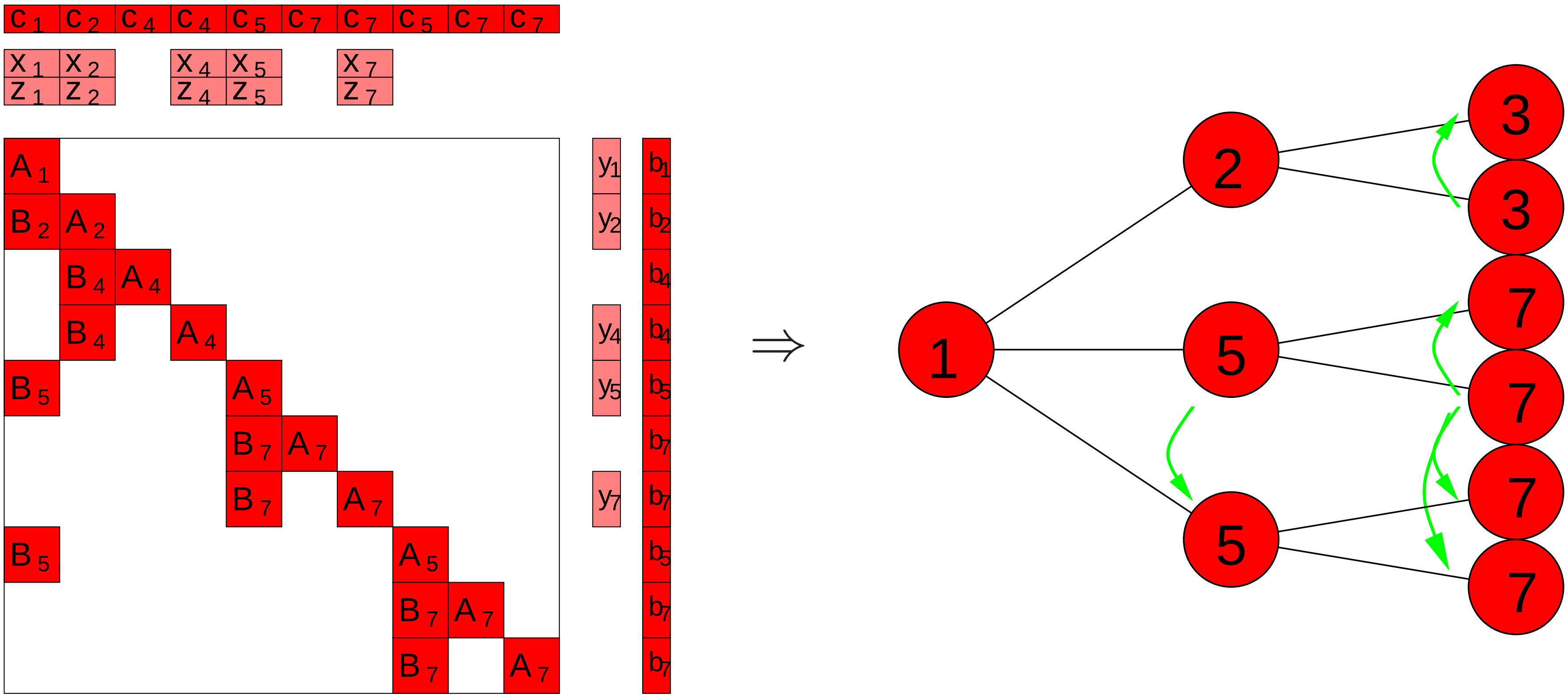
Stochastic Programming Warmstarts



To solve the problem by warmstarting, reverse the process

- Solve reduced problem (to low accuracy)
- Expand the problem to original size

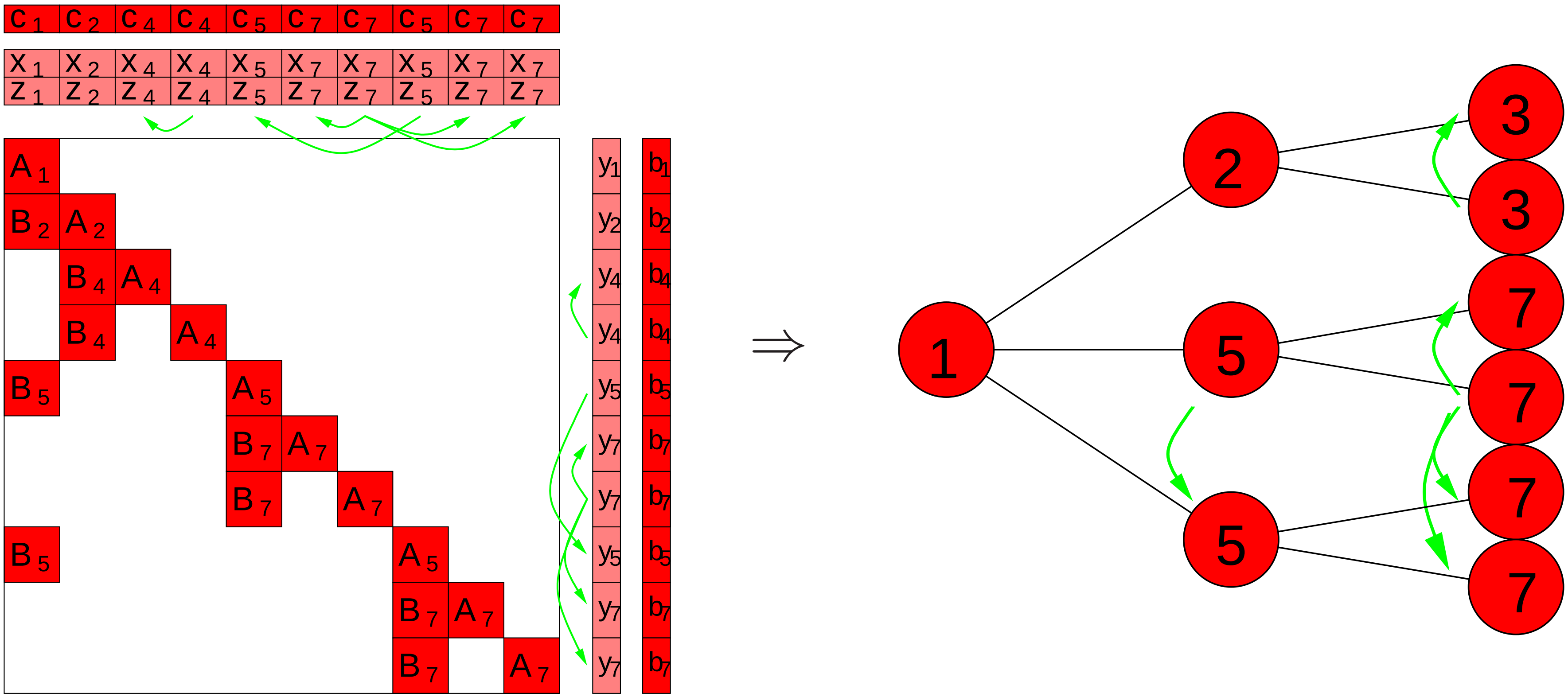
Stochastic Programming Warmstarts



To solve the problem by warmstarting, reverse the process

- Solve reduced problem (to low accuracy)
- Expand the problem to original size (by duplicating scenarios)

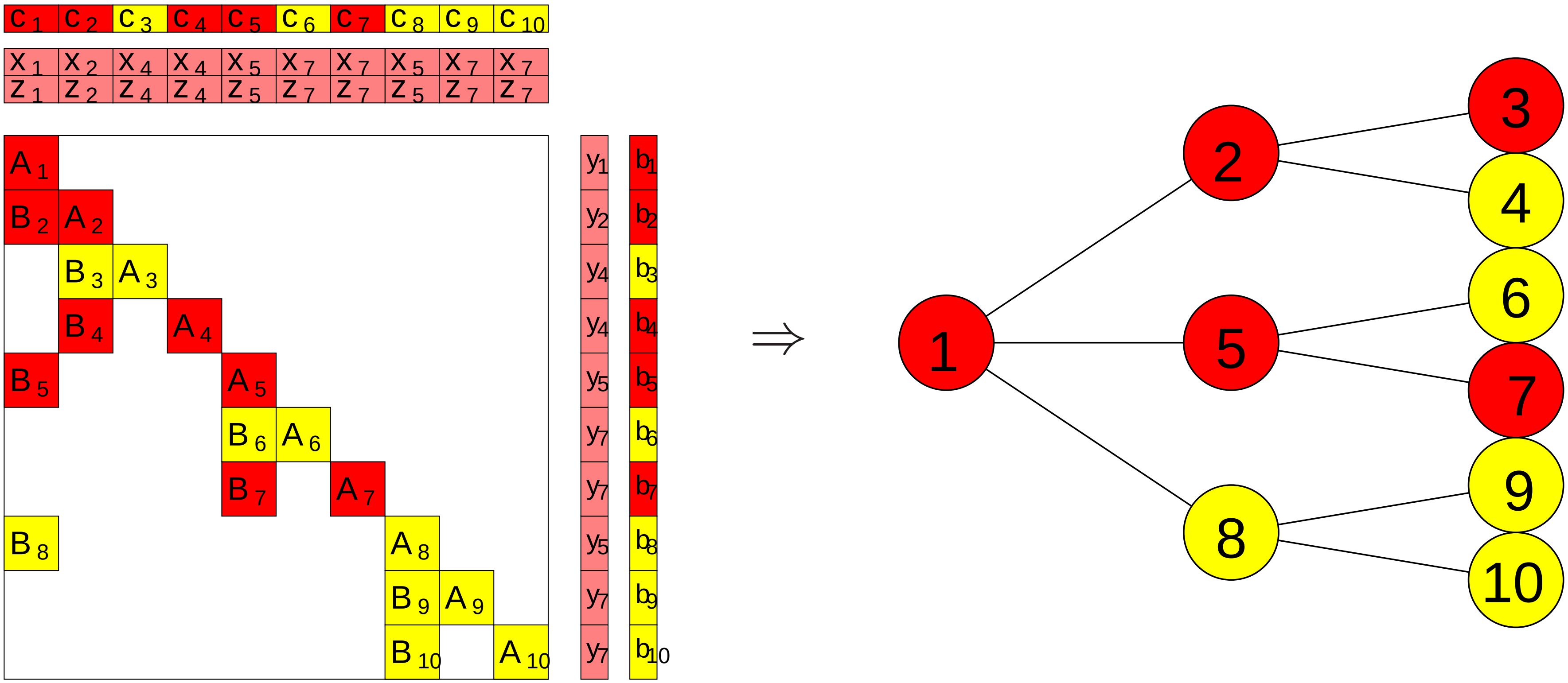
Stochastic Programming Warmstarts



To solve the problem by warmstarting, reverse the process

- Solve reduced problem (to low accuracy)
- Expand the problem to original size (by duplicating scenarios)
- Expand solution to primal/dual feasible solution to expanded problem

Stochastic Programming Warmstarts

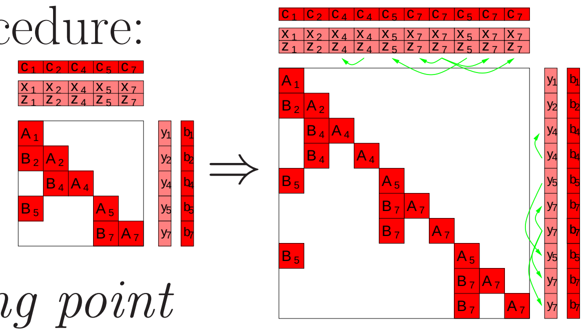


To solve the problem by warmstarting, reverse the process

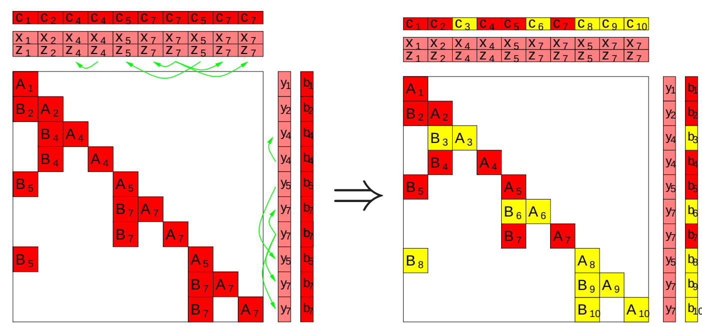
- Solve reduced problem (to low accuracy)
- Expand the problem to original size (by duplicating scenarios)
- Expand solution to primal/dual feasible solution to expanded problem
- Use this to warmstart full problem

Stochastic Programming Warmstarts

The proposed warmstart procedure is a **two** stage procedure:



- Reduced Problem ⇒ Expanded Problem
  - *Can construct primal/dual feasible starting point*
  - *Although point is not central (We are duplicating constraints!)*



- Expanded Problem ⇒ Full Problem
  - *Can bound changes in problem data (differences in scenarios)*

Outline of stochastic warmstart procedure

1. Choose reduced scenario tree $\mathcal{T}_R$.

2. Solve problem $P_R = P(\mathcal{T}_R)$ on reduced tree to low accuracy:

$$(x_R, y_R, s_R) \in \mathcal{N}_{-\infty}(\gamma_0), \quad x_R^T s_R/n \approx \mu_0$$

3. Construct expanded tree $\hat{\mathcal{T}}$
(by duplicating nodes in reduced tree to obtain shape of full tree $\mathcal{T}$)

4. Construct primal-dual feasible solution $(\hat{x}, \hat{y}, \hat{s})$ to $\hat{P} = P(\hat{\mathcal{T}})$ from (x_R, y_R, s_R):

$$(\hat{x}, \hat{y}, \hat{s}) \in \hat{\mathcal{N}}_{-\infty}(\rho\gamma_0), \quad \hat{x}^T\hat{s}/\hat{n} \approx \mu_0, 0 < \rho < 1$$

5. Use as warmstart for the full problem $P = P(\mathcal{T})$.

1. How to choose reduced tree scenarios

Select sample scenarios $\eta_R^l = \{W_t^l, T_t^l, c_t^l, h_t^l\}_{t=1}^T \in \mathcal{T}$ by minimizing scenario distances:

$$d(\eta^l, \eta^k) := \sum_{t=1}^{T} \|W_t^l - W_t^k\|_2 + \|T_t^l - T_t^k\|_2 + \|c_t^l - c_t^k\|_2 + \|h_t^l - h_t^k\|_2$$

Denote the scenario distance between two trees by

$$d(\mathcal{T}_R, \mathcal{T}) = \max_{\eta^k \in \mathcal{T}} \min_{\eta_R^l \in \mathcal{T}_R} d(\eta^k, \eta_R^l)$$

For a given size $n_R = |\mathcal{T}_R|$ of the reduced tree $\mathcal{T}_R$, choose scenarios $\eta_R \in \mathcal{T}_R$ such that

$$\mathcal{T}_R = \arg \min_{\tilde{\mathcal{T}} : |\tilde{\mathcal{T}}| = n_R} d(\tilde{\mathcal{T}}, \mathcal{T}\}$$

1. How to choose reduced tree scenarios(assign probabilities)

Given the reduced tree $\mathcal{T}_R \subset \mathcal{T}$. We have a mapping

$$\begin{aligned} r : \mathcal{T} &\to \mathcal{T}_R \\ r(\eta) &= \arg\min_{\eta_R \in \mathcal{T}_R} d(\eta, \eta_R), \qquad \textit{(closest node in } \mathcal{T}_R\textit{)} \end{aligned}$$

And its inverse

$$\mathcal{I}_k = I(\eta_R^k) := \{\eta^l \in \mathcal{T} : r(\eta^l) = \eta_R^k\}$$

then

$$p_R^k = \sum_{\eta^l \in \mathcal{I}_k} p^l$$

$\Rightarrow$ Scenario η_R^k *aggregates* the scenarios in $I(\eta_R^k)$.

$$\rho = \min_{\eta^l \in \mathcal{T}} \left\{ \frac{p^l}{p_R^{r(l)}} \frac{n}{n_R}, \frac{p_R^{r(l)}}{p^l} \frac{n_R}{n} \right\}$$

$\Rightarrow \rho \leq 1$. For a well balanced selection of nodes $\mathcal{T}_R \subset \mathcal{T}$ we expect $\rho \approx 1$

2. Solve problem on reduced tree

3. Construct expanded tree

4. Construct solution to expanded tree $\hat{\mathcal{T}}$

Given $(x_R, y_R, s_R) \in \mathcal{N}_{R,-\infty}(\gamma)$ construct primal-dual feasible point for expanded problem by

$$\hat{x}^i = x_R^{r(i)}, \qquad (\hat{y}^i, \hat{s}^i) = \frac{p^i}{p_R^{r(i)}}(y_R^{r(i)}, s_R^{r(i)})$$

(Copy primal solution, scale dual solution according to probabilities)

$$\Rightarrow$$

Lemma.

$$(x_R, y_R, s_R) \in \mathcal{N}_{R,-\infty}(\gamma) \Rightarrow (\hat{x}, \hat{y}, \hat{s}) \in \hat{\mathcal{N}}_{-\infty}(\rho\gamma)$$

5. Warmstart the full problem

Change scenario data to that of the full problem

- induces change $\Delta b, \Delta c$ to problem data
- induces residuals ξ_b, ξ_c (and thus δ_{bc}^{GG}).

We can bound

$$\delta_{bc}^{GG} \leq C\sqrt{|\mathcal{T}|}d(\mathcal{T}, \mathcal{T}_R)$$

and therefore using the earlier condition for succesful warmstart

$$\delta_{bc}^{GG} \leq \frac{\gamma}{B^\infty(1 + 1/\gamma)}$$

we get:

Lemma. The stochastic programming warmstart is succesful if

$$d(\mathcal{T}, \mathcal{T}_R) \leq \frac{1}{(1+\gamma)B^\infty C}\frac{\mu_0}{\sqrt{|\mathcal{T}|}}$$

(i.e. μ_0 large enough compared to size of full tree - in practice use fairly small μ_0)

Test Problems

Three sources of test problems:

- Collection of **standard SMPS problems**.
- **Capacity assignment problem** with uncertain demand
- Pricing electricity **swing options**

Setup (for the first two test problem sets)

- 2 scenarios in the reduced tree
- Reduced problem optimality tolerance: 5.0×10^{-1}
- Complete problem optimality tolerance: 5.0×10^{-8}

Swing problem uses

- 16 scenarios in the reduced tree
- Reduced problem optimality tolerance: 5.0×10^{-1}
- Complete problem optimality tolerance: 1.0×10^{-3}

Numerical results: Standard SP test problems

Problem data			Cold start		Warm start	
Name	Stgs	Scens	Iters	Seconds	Iters	Seconds
fxm2-16	2	16	22	1.2	13	1.0
fxm3-6	3	36	30	1.5	17	1.3
fxm3-16	3	256	40	31.1	20	20.7
fxm4-6	4	216	30	8.2	22	8.3
fxm4-16	4	4096	41	218.3	27	182.6
pltexpA3-16	3	256	26	153.8	14	87.8
pltexpA4-6	4	216	36	55.8	16	27.5
pltexpA5-6	5	1296	81	772.0	30	311.5
storm27	2	27	41	95.4	22	53.2
storm125	2	125	73	107.3	36	69.1
storm1000	2	1000	107	1498.3	45	831.5

Numerical results: Capacity Assignment Problems:

Problem data			Cold start		Warm start	
Name	Stgs	Scens	Iters	Seconds	Iters	Seconds
mnx-200	2	200	13	12.9	7	7.3
mnx-800	2	800	17	58.8	10	39.5
mnx-1600	2	1600	19	131.1	10	68.8
jlg-200	2	200	45	164.9	17	39.5
jlg-800	2	800	27	353.4	10	152.9
jlg-1600	2	1600	32	855.3	13	360.6
mgntA-100	2	100	28	260.0	14	156.2
mgntA-200	2	200	50	877.1	35	690.6
mgntA-400	2	400	40	1470.3	14	572.5
mgntB-100	2	100	23	511.1	14	318.0
mgntB-200	2	200	25	909.4	8	332.4
mgntB-400	2	400	29	2154.5	7	538.1

Test Problem: Pricing Electricity Swing Options

"An electricity swing option is a contract that allows the holder to buy between $\underline{e}$ and $\bar{e}$ units of electricity up to time T for a price of K (per unit)."

Pricing model is a (multistage) stochastic program (typically $T > 500$).

$$\begin{array}{ll} \min_{e,p} & I\!\!E_\xi \left[\sum_{t=1}^T (S_t(\xi_t) - K) p_t(\xi_t)\right] \\ \text{s.t.} & \underline{e} \le e_T(\xi_T) \le \bar{e} \\ & e_t(\xi_t) - e_{t-1}(\xi_{t-1}) = p_t(\xi_{t-1}) \\ & \underline{p}_t \le p_t(\xi_t) \le \bar{p}_t, \qquad t = 1, \dots, T \\ & |p_t(\xi_t) - p_{t-1}(\xi_{t-1}| \le \rho_t, \qquad t = 1, \dots, T \end{array}$$

where

- S_t: spot price for electricity at time t
- K: strike price of the option
- p_t: electricity used in period t
- e_t: total usage of option up to period t

Numerical results: Swing Options:

Problem data			Cold start		Warm start	
Name	Stgs	Scens	Iters	Seconds	Iters	Seconds
sw6-4	6	1024	10	0.7	4	0.3
sw8-4	8	65536	14	114	4	34
sw9-4	9	262144	24	481	5	90

Here the reduced tree has 16 scenarios

Results are **not** sensitive to the size of the reduced tree!

Numerical results: Swing Options:

Problem sw9-4: $m = 1.048.574, n = 1.398.098$. $\sharp$scenarios = 262.144.

$\|\mathcal{T}_R\|$	Iter ($\mathcal{T}_R$)	Iter ($\mathcal{T}$)	Time (s)
cold	-	24	481
1	4	8	173
2	4	6	148
3	5	5	126
4	5	5	90
16	5	5	103
64	6	5	93
256	8	5	92
1024	10	4	101
4096	10	5	111
16384	14	4	124
65536	20	5	259

Reduced tree is solved to a value of $\mu = 0.5$.

Conclusions

- IPM **can** be warmstarted
- IPM warmstarts can save 50%-60% of iterations (on all problem sizes)
- Can exploit stochastic programming structure to significantly speed-up solution.

Future Work:

- Multi-Step scheme
- Complexity of such a scheme
- Carry over to other structures (PDE constrained optimization)
- Integrate into structured modelling language