Parallel distributed-memory simplex for large-scale stochastic LP problems

Julian Hall[1] Miles Lubin[2]

[1]School of Mathematics, University of Edinburgh

[2]MIT

CSC14

Lyon

21 July 2014

Overview

- Background
- Stochastic LP problems
  - Structure
  - How to exploit data parallelism
- Results
- Conclusions

Linear programming (LP)

$$\begin{array}{ll} \text{minimize} & \boldsymbol{c}^T\boldsymbol{x} \\ \text{subject to} & A\boldsymbol{x} = \boldsymbol{b} \quad \boldsymbol{x} \geq \boldsymbol{0} \end{array}$$

Background

- Fundamental model in optimal decision-making
- Solution techniques
  - Simplex method (1947)
  - Interior point methods (1984)
- Large problems have
  - 10^3–$10^{\cancel{7}8}$ variables
  - 10^3–$10^{\cancel{7}8}$ constraints
- Matrix A is (usually) sparse

Example

STAIR: 356 rows, 467 columns and 3856 nonzeros

Solving LP problems

$$\begin{array}{llll} \text{minimize} & f_P = \boldsymbol{c}^T\boldsymbol{x} & \qquad \text{maximize} & f_D = \boldsymbol{b}^T\boldsymbol{y} \\ \text{subject to} & A\boldsymbol{x} = \boldsymbol{b} \quad \boldsymbol{x} \geq \boldsymbol{0} \quad (P) & \qquad \text{subject to} & A^T\boldsymbol{y} + \boldsymbol{s} = \boldsymbol{c} \quad \boldsymbol{s} \geq \boldsymbol{0} \quad (D) \end{array}$$

Optimality conditions

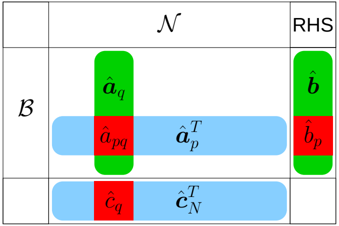
- For a partition $\mathcal{B} \cup \mathcal{N}$ of the variable set with nonsingular **basis matrix** B in

$$B\boldsymbol{x}_B + N\boldsymbol{x}_N = \boldsymbol{b} \text{ for (P)} \qquad \text{and} \qquad \begin{bmatrix} B^T \\ N^T \end{bmatrix} \boldsymbol{y} + \begin{bmatrix} \boldsymbol{s}_B \\ \boldsymbol{s}_N \end{bmatrix} = \begin{bmatrix} \boldsymbol{c}_B \\ \boldsymbol{c}_N \end{bmatrix} \text{ for (D)}$$

  with $\boldsymbol{x}_N = \boldsymbol{0}$ and $\boldsymbol{s}_B = \boldsymbol{0}$

  - Primal basic variables $\boldsymbol{x}_B$ given by $\widehat{\boldsymbol{b}} = B^{-1}\boldsymbol{b}$
  - Dual non-basic variables $\boldsymbol{s}_N$ given by $\widehat{\boldsymbol{c}}_N^T = \boldsymbol{c}_N^T - \boldsymbol{c}_B^T B^{-1} N$
- Partition is optimal if there is
  - Primal feasibility $\widehat{\boldsymbol{b}} \geq \boldsymbol{0}$
  - Dual feasibility $\widehat{\boldsymbol{c}}_N \geq \boldsymbol{0}$

Simplex algorithm: Each iteration



Dual algorithm: Assume $\widehat{\boldsymbol{c}}_N \geq \mathbf{0}$ Seek $\widehat{\boldsymbol{b}} \geq \mathbf{0}$

Scan $\widehat{b}_i$, $i \in \mathcal{B}$, for a good candidate p to leave $\mathcal{B}$ CHUZR

Scan $\widehat{c}_j/\widehat{a}_{pj}$, $j \in \mathcal{N}$, for a good candidate q to leave $\mathcal{N}$ CHUZC

Update: Exchange p and q between $\mathcal{B}$ and $\mathcal{N}$

Update $\widehat{\boldsymbol{b}} := \widehat{\boldsymbol{b}} - \theta_p \widehat{\boldsymbol{a}}_q$ $\theta_p = \widehat{b}_p/\widehat{a}_{pq}$ UPDATE-PRIMAL

Update $\widehat{\boldsymbol{c}}_N^T := \widehat{\boldsymbol{c}}_N^T - \theta_d \widehat{\boldsymbol{a}}_p^T$ $\theta_d = \widehat{c}_q/\widehat{a}_{pq}$ UPDATE-DUAL

Revised simplex method (RSM): Computation

Major computational components

$\pi_p^T = \mathbf{e}_p^T B^{-1}$	BTRAN	$\widehat{\mathbf{a}}_p^T = \pi_p^T N$	PRICE
$\widehat{\mathbf{a}}_q = B^{-1}\mathbf{a}_q$	FTRAN	Invert B	INVERT

Hyper-sparsity

- Vectors $\mathbf{e}_p$ and $\mathbf{a}_q$ are always sparse
- B may be highly reducible; B^{-1} may be sparse
- Vectors π_p, $\widehat{\mathbf{a}}_p^T$ and $\widehat{\mathbf{a}}_q$ may be sparse
- Efficient implementations must exploit these features

H and McKinnon (1998–2005), Bixby (1999)
Clp, Koberstein and Suhl (2005–2008)

Stochastic MIP problems: General

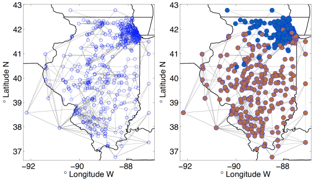
Two-stage stochastic LPs have column-linked block angular structure

$$
\begin{array}{rccccccccccc}
\text{minimize} & \boldsymbol{c}_0^T\boldsymbol{x}_0 & + & \boldsymbol{c}_1^T\boldsymbol{x}_1 & + & \boldsymbol{c}_2^T\boldsymbol{x}_2 & + & \dots & + & \boldsymbol{c}_N^T\boldsymbol{x}_N & & \\
\text{subject to} & A\boldsymbol{x}_0 & & & & & & & & & = & \boldsymbol{b}_0 \\
 & T_1\boldsymbol{x}_0 & + & W_1\boldsymbol{x}_1 & & & & & & & = & \boldsymbol{b}_1 \\
 & T_2\boldsymbol{x}_0 & & & + & W_2\boldsymbol{x}_2 & & & & & = & \boldsymbol{b}_2 \\
 & \vdots & & & & & & \ddots & & & & \vdots \\
 & T_N\boldsymbol{x}_0 & & & & & & & + & W_N\boldsymbol{x}_N & = & \boldsymbol{b}_N \\
 & \boldsymbol{x}_0 \ge \boldsymbol{0} & & \boldsymbol{x}_1 \ge \boldsymbol{0} & & \boldsymbol{x}_2 \ge \boldsymbol{0} & & \dots & & \boldsymbol{x}_N \ge \boldsymbol{0} & &
\end{array}
$$

- Variables $\boldsymbol{x}_0 \in \mathbb{R}^{n_0}$ are **first stage** decisions
- Variables $\boldsymbol{x}_i \in \mathbb{R}^{n_i}$ for $i = 1, \dots, N$ are **second stage** decisions
  Each corresponds to a **scenario** which occurs with modelled probability
- The objective is the expected cost of the decisions
- In stochastic MIP problems, some/all decisions are discrete

Stochastic MIP problems: For Argonne

- Power systems optimization project at Argonne
- Integer second-stage decisions
- Stochasticity comes from availability of wind-generated electricity
- Initial experiments carried out using model problem
- Number of scenarios increases with refinement of probability distribution sampling
- Solution via branch-and-bound
  - Solve root node using parallel IPM solver `PIPS` Lubin, Petra *et al.* (2011)
  - Solve subsequent nodes using parallel dual simplex solver `PIPS-S` Lubin, H *et al.* (2013)



Illinois power network (2010)

Stochastic MIP problems: General

Convenient to permute the LP thus:

$$\begin{array}{rcccccccccccc}
\text{minimize} & \mathbf{c}_1^T\mathbf{x}_1 & + & \mathbf{c}_2^T\mathbf{x}_2 & + & \dots & + & \mathbf{c}_N^T\mathbf{x}_N & + & \mathbf{c}_0^T\mathbf{x}_0 & & \\
\text{subject to} & W_1\mathbf{x}_1 & & & & & & & + & T_1\mathbf{x}_0 & = & \mathbf{b}_1 \\
& & & W_2\mathbf{x}_2 & & & & & + & T_2\mathbf{x}_0 & = & \mathbf{b}_2 \\
& & & & & \ddots & & & \vdots & & & \vdots \\
& & & & & & & W_N\mathbf{x}_N & + & T_N\mathbf{x}_0 & = & \mathbf{b}_N \\
& & & & & & & & & A\mathbf{x}_0 & = & \mathbf{b}_0 \\
& \mathbf{x}_1 \ge \mathbf{0} & & \mathbf{x}_2 \ge \mathbf{0} & & \dots & & \mathbf{x}_N \ge \mathbf{0} & & \mathbf{x}_0 \ge \mathbf{0} & &
\end{array}$$

Exploiting problem structure: Basis matrix inversion

- Inversion of the basis matrix B is key to revised simplex efficiency
- For column-linked BALP problems

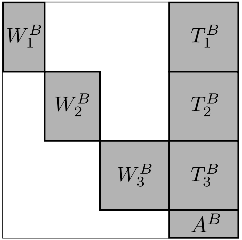
$$B = \begin{bmatrix} W_1^B & & & T_1^B \\ & \ddots & & \vdots \\ & & W_N^B & T_N^B \\ & & & A^B \end{bmatrix}$$

- W_i^B are columns corresponding to n_i^B basic variables in scenario i
- $\begin{bmatrix} T_1^B \\ \vdots \\ T_N^B \\ A^B \end{bmatrix}$ are columns corresponding to n_0^B basic first stage decisions

Exploiting problem structure: Basis matrix inversion

- Inversion of the basis matrix B is key to revised simplex efficiency
- For column-linked BALP problems

$$B = \begin{bmatrix} W_1^B & & & T_1^B \\ & \ddots & & \vdots \\ & & W_N^B & T_N^B \\ & & & A^B \end{bmatrix}$$

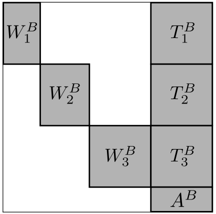


- B is nonsingular so
  - W_i^B are "tall": full column rank
  - $\begin{bmatrix} W_i^B & T_i^B \end{bmatrix}$ are "wide": full row rank
  - A^B is "wide": full row rank

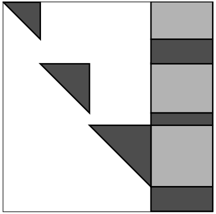
- Scope for parallel inversion is immediate and well known

Duff and Scott (2004)

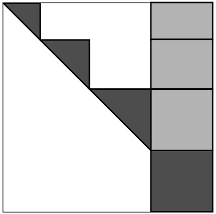
Exploiting problem structure: Basis matrix inversion



- Eliminate sub-diagonal entries in each W_i^B (independently)



- Apply elimination operations to each T_i^B (independently)



- Accumulate non-pivoted rows from the W_i^B with A^B and complete elimination

Exploiting problem structure: Basis matrix inversion

- After Gaussian elimination, have invertible representation of

$$B = \left[\begin{array}{ccc|c} S_1 & & & C_1 \\ & \ddots & & \vdots \\ & & S_N & C_N \\ \hline R_1 & \dots & R_N & V \end{array}\right] = \left[\begin{array}{c|c} S & C \\ \hline R & V \end{array}\right]$$

- Specifically
  - $L_i U_i = S_i$ of dimension n_i^B
  - $\widehat{C}_i = L_i^{-1} C_i$
  - $\widehat{R}_i = R_i U_i^{-1}$
  - LU factors of the Schur complement $M = V - RS^{-1}C$ of dimension n_0^B
- Scope for parallelism since each GE applied to $\left[W_i^B \mid T_i^B \right]$ is independent

Exploiting problem structure: Solving $Bx = \boldsymbol{b}$

FTRAN for $B\boldsymbol{x} = \boldsymbol{b}$

Solve $\begin{bmatrix} S & C \\ R & V \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_\bullet \\ \boldsymbol{x}_0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{b}_\bullet \\ \boldsymbol{b}_0 \end{bmatrix}$ as

1. $L_i \boldsymbol{y}_i = \boldsymbol{b}_i,\ i = 1, \ldots, N$
2. $\boldsymbol{z}_i = \widehat{R}_i \boldsymbol{y}_i,\ i = 1, \ldots, N$
3. $\boldsymbol{z} = \boldsymbol{b}_0 - \sum_{i=1}^{N} \boldsymbol{z}_i$
4. $M\boldsymbol{x}_0 = \boldsymbol{z}$
5. $U_i \boldsymbol{x}_i = \boldsymbol{y}_i - \widehat{C}_i \boldsymbol{x}_0,\ i = 1, \ldots, N$

- Appears to be dominated by parallelizable
  - Solves $L_i \boldsymbol{y}_i = \boldsymbol{b}_i$ and $U_i \boldsymbol{x}_i = \boldsymbol{y}_i - \widehat{C}_i \boldsymbol{x}_0$
  - Products $\widehat{R}_i \boldsymbol{y}_i$ and $\widehat{C}_i \boldsymbol{x}_0$
- Curse of exploiting hyper-sparsity
  - In simplex, $\boldsymbol{b}_\bullet$ is from constraint column

    Either $\begin{bmatrix} \boldsymbol{t}_{1q} \\ \vdots \\ \boldsymbol{t}_{Nq} \end{bmatrix}$ or, more likely, $\begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{w}_{iq} \\ \boldsymbol{0} \end{bmatrix}$
  - In latter case, the $\boldsymbol{y}_i$ inherit structure
    - Only one $L_i \boldsymbol{y}_i = \boldsymbol{w}_{iq}$
    - Only one $\widehat{R}_i \boldsymbol{y}_i$
- Less scope for parallelism than anticipated

Exploiting problem structure: Solving $B^Tx = \boldsymbol{b}$

BTRAN for $B^T\boldsymbol{x} = \boldsymbol{b}$

Solve $\begin{bmatrix} S^T & R^T \\ C^T & V^T \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_\bullet \\ \boldsymbol{x}_0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{b}_\bullet \\ \boldsymbol{b}_0 \end{bmatrix}$ as

1. $U_i^T\boldsymbol{y}_i = \boldsymbol{b}_i,\ i = 1, \ldots, N$
2. $\boldsymbol{z}_i = \widehat{C}_i^T\boldsymbol{y}_i,\ i = 1, \ldots, N$
3. $\boldsymbol{z} = \boldsymbol{b}_0 - \sum_{i=1}^{N} \boldsymbol{z}_i$
4. $M^T\boldsymbol{x}_0 = \boldsymbol{z}$
5. $L_i^T\boldsymbol{x}_i = \boldsymbol{y}_i - \widehat{R}_i^T\boldsymbol{x}_0,$
   $i = 1, \ldots, N$

- Appears to be dominated by parallelizable
  - Solves $U_i^T\boldsymbol{y}_i = \boldsymbol{b}_i$ and $L_i^T\boldsymbol{x}_i = \boldsymbol{y}_i - \widehat{R}_i^T\boldsymbol{x}_0$
  - Products $\widehat{C}_i^T\boldsymbol{y}_i$ and $\widehat{R}_i^T\boldsymbol{x}_0$
- Curse of exploiting hyper-sparsity
  - In simplex, $\boldsymbol{b} = \boldsymbol{e}_p$
  - At most one solve $U_i^T\boldsymbol{y}_i = \boldsymbol{b}_i$
  - At most one $\widehat{C}_i^T\boldsymbol{y}_i$
- Less scope for parallelism than anticipated

Exploiting problem structure: Forming $\pi_p^T N$

- PRICE forms

$$
\begin{bmatrix} \pi_1^T & \pi_2^T & \dots & \pi_N^T & \pi_0^T \end{bmatrix}
\begin{bmatrix} W_1^N & & & & T_1^N \\ & W_2^N & & & T_2^N \\ & & \ddots & & \vdots \\ & & & W_N^N & T_N^N \\ & & & & A^N \end{bmatrix}
$$

$$
= \begin{bmatrix} \pi_1^T W_1^N & \pi_2^T W_2^N & \dots & \pi_N^T W_N^N & \pi_0^T A^N + \sum_{i=1}^{N} \pi_i^T T_i^N \end{bmatrix}
$$

- Dominated by parallelizable products $\pi_i^T W_i^N$ and $\pi_i^T T_i^N$

Results: Stochastic LP test problems

Test Problem	1st Stage n_0	m_0	2nd-Stage Scenario n_i	m_i	Nonzero Elements A	W_i	T_i
`Storm`	121	185	1,259	528	696	3,220	121
`SSN`	89	1	706	175	89	2,284	89
`UC12`	3,132	0	56,532	59,436	0	163,839	3,132
`UC24`	6,264	0	113,064	118,872	0	327,939	6,264

- `Storm` and `SSN` are publicly available
- `UC12` and `UC24` are stochastic unit commitment problems developed at Argonne
  - Aim to choose optimal on/off schedules for generators on the power grid of the state of Illinois over a 12-hour and 24-hour horizon
  - In practice each scenario corresponds to a weather simulation
    Model problem generates scenarios by normal perturbations

Zavala (2011)

Results: Baseline serial performance for large instances

Serial performance of PIPS-S and clp

Problem	Dimensions	Solver	Iterations	Time (s)	Iter/sec
Storm	$n =$10,313,849	PIPS-S	6,353,593	385,825	16.5
8,192 scen.	$m = 4,325,561$	clp	6,706,401	133,047	50.4
SSN	$n = 5,783,651$	PIPS-S	1,025,279	58,425	17.5
8,192 scen.	$m = 1,433,601$	clp	1,175,282	12,619	93.1
UC12	$n = 1,812,156$	PIPS-S	1,968,400	236,219	8.3
32 scen.	$m = 1,901,952$	clp	2,474,175	39,722	62.3
UC24	$n = 1,815,288$	PIPS-S	2,142,962	543,272	3.9
16 scen.	$m = 1,901,952$	clp	2,441,374	41,708	58.5

Results: On Fusion cluster

Speed-up of PIPS-S relative to 1-core PIPS-S and 1-core clp

Cores	Storm	SSN	UC12	UC24
1	1.0	1.0	1.0	1.0
4	3.6	3.5	2.7	3.0
8	7.3	7.5	6.1	5.3
16	13.6	15.1	8.5	8.9
32	24.6	30.3	14.5	
clp	8.5	6.5	2.4	0.7

Results: On Fusion cluster - larger instances

	Storm	SSN	UC12	UC24
Scenarios	32,768	32,768	512	256
Variables	41,255,033	23,134,297	28,947,516	28,950,648
Constraints	17,301,689	5,734,401	30,431,232	30,431,232

Results: On Fusion cluster - larger instances, from an advanced basis

Speed-up of PIPS-S relative to 1-core PIPS-S and 1-core clp

Cores	Storm	SSN	UC12	UC24
1	1	1	1	1
8	15	19	7	6
16	52	45	14	12
32	117	103	26	22
64	152	181	44	41
128	202	289	60	64
256	285	383	70	80
clp	299	45	67	68

Results: On Blue Gene supercomputer - very large instance

- Instance of `UC12`
  - 8,192 scenarios
  - 463,113,276 variables
  - 486,899,712 constraints
- Requires 1 TB of RAM ($\geq$ 1024 Blue Gene cores)
- Runs from an advanced basis

Cores	Iterations	Time (h)	Iter/sec
1024	Exceeded execution time limit		
2048	82,638	6.14	3.74
4096	75,732	5.03	4.18
8192	86,439	4.67	5.14

Conclusions

- Developed a distributed dual revised simplex solver for column linked BALP
- Demonstrated scalable parallel performance
  - For highly specialised problems
  - On highly specialised machines
- Solved problems which would be intractable using commercial serial solvers

Slides: http://www.maths.ed.ac.uk/hall/CSC14/

Paper: M. Lubin, J. A. J. Hall, C. G. Petra, and M. Anitescu
Parallel distributed-memory simplex for large-scale stochastic LP problems
Computational Optimization and Applications, 55(3):571–596, 2013

COIN|OR

Cup winners: 2013