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

Julian Hall\textsuperscript{1} Miles Lubin\textsuperscript{2}

\textsuperscript{1}School of Mathematics, University of Edinburgh

\textsuperscript{2}MIT

CSC14

Lyon

21 July 2014
Overview

- Background
- Stochastic LP problems
  - Structure
  - How to exploit data parallelism
- Results
- Conclusions
Linear programming (LP)

\[
\begin{align*}
\text{minimize} & \quad c^T x \\
\text{subject to} & \quad Ax = b \quad x \geq 0
\end{align*}
\]

Background

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

Example

STAIR: 356 rows, 467 columns and 3856 nonzeros
minimize \( f_P = c^T x \) \hspace{1cm} \text{maximize} \hspace{1cm} f_D = b^T y \\
subject to \hspace{1cm} A x = b \hspace{1cm} x \geq 0 \hspace{1cm} (P) \hspace{1cm} \text{subject to} \hspace{1cm} A^T y + s = c \hspace{1cm} s \geq 0 \hspace{1cm} (D)

### Optimality conditions

- For a partition \( B \cup N \) of the variable set with nonsingular basis matrix \( B \) in
  \[
  B x_B + N x_N = b \quad \text{for (P)} \quad \text{and} \quad \begin{bmatrix} B^T \\ N^T \end{bmatrix} y + \begin{bmatrix} s_B \\ s_N \end{bmatrix} = \begin{bmatrix} c_B \\ c_N \end{bmatrix} \quad \text{for (D)}
  \]

  with \( x_N = 0 \) and \( s_B = 0 \)

  - **Primal** basic variables \( x_B \) given by \( \hat{b} = B^{-1} b \)
  - **Dual** non-basic variables \( s_N \) given by \( \hat{c}_N^T = c_N^T - c_B^T B^{-1} N \)

- Partition is optimal if there is
  - **Primal feasibility** \( \hat{b} \geq 0 \)
  - **Dual feasibility** \( \hat{c}_N \geq 0 \)
Simplex algorithm: Each iteration

<table>
<thead>
<tr>
<th>B</th>
<th>N</th>
<th>RHS</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \hat{\mathbf{a}}_q )</td>
<td>( \hat{\mathbf{c}}_q )</td>
<td>( \hat{\mathbf{b}} )</td>
</tr>
<tr>
<td>( \hat{\mathbf{a}}_{pq} )</td>
<td>( \hat{\mathbf{c}}_{N} )</td>
<td>( \hat{\mathbf{b}}_p )</td>
</tr>
</tbody>
</table>

Dual algorithm: Assume \( \hat{\mathbf{c}}_N \geq 0 \) Seek \( \hat{\mathbf{b}} \geq 0 \)

Scan \( \hat{\mathbf{b}}_i, i \in \mathcal{B} \), for a good candidate \( p \) to leave \( \mathcal{B} \)  \( \text{CHUZR} \)

Scan \( \hat{\mathbf{c}}_j/\hat{\mathbf{a}}_p, j \in \mathcal{N} \), for a good candidate \( q \) to leave \( \mathcal{N} \)  \( \text{CHUZC} \)

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

Update \( \hat{\mathbf{b}} := \hat{\mathbf{b}} - \theta_p \hat{\mathbf{a}}_q \)  \( \theta_p = \hat{\mathbf{b}}_p/\hat{\mathbf{a}}_{pq} \)  \( \text{UPDATE-PRIMAL} \)

Update \( \hat{\mathbf{c}}_N^T := \hat{\mathbf{c}}_N^T - \theta_d \hat{\mathbf{a}}_p^T \)  \( \theta_d = \hat{\mathbf{c}}_q/\hat{\mathbf{a}}_{pq} \)  \( \text{UPDATE-DUAL} \)
Revised simplex method (RSM): Computation

Major computational components

\[
\pi_p^T = e_p^T B^{-1} \quad \text{BTRAN} \quad \hat{a}_p^T = \pi_p^T N \quad \text{PRICE}
\]

\[
\hat{a}_q = B^{-1} a_q \quad \text{FTRAN} \quad \text{Invert } B \quad \text{INVERT}
\]

Hyper-sparsity

- Vectors \( e_p \) and \( a_q \) are always sparse
- \( B \) may be highly reducible; \( B^{-1} \) may be sparse
- Vectors \( \pi_p, \hat{a}_p^T \) and \( \hat{a}_q \) may be sparse
- Efficient implementations must exploit these features

Stochastic MIP problems: General

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

\[
\begin{align*}
\text{minimize} & \quad c_0^T x_0 + c_1^T x_1 + c_2^T x_2 + \ldots + c_N^T x_N \\
\text{subject to} & \quad A x_0 = b_0 \\
& \quad T_1 x_0 + W_1 x_1 = b_1 \\
& \quad T_2 x_0 + W_2 x_2 = b_2 \\
& \quad \vdots \quad \vdots \quad \vdots \\
& \quad T_N x_0 + W_N x_N = b_N \\
& \quad x_0 \geq 0 \quad x_1 \geq 0 \quad x_2 \geq 0 \quad \ldots \quad x_N \geq 0
\end{align*}
\]

- Variables \( x_0 \in \mathbb{R}^{n_0} \) are \textbf{first stage} decisions
- Variables \( x_i \in \mathbb{R}^{n_i} \) for \( i = 1, \ldots, N \) are \textbf{second stage} decisions
  - Each corresponds to a \textbf{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)
Convenient to permute the LP thus:

\[
\begin{align*}
\text{minimize} & \quad c_1^T x_1 + c_2^T x_2 + \ldots + c_N^T x_N + c_0^T x_0 \\
\text{subject to} & \quad W_1 x_1 + T_1 x_0 = b_1 \\
& \quad W_2 x_2 + T_2 x_0 = b_2 \\
& \quad \vdots \quad \vdots \quad \vdots \\
& \quad W_N x_N + T_N x_0 = b_N \\
& \quad A x_0 = b_0 \\
& \quad x_1 \geq 0 \quad x_2 \geq 0 \quad \ldots \quad x_N \geq 0 \quad x_0 \geq 0
\end{align*}
\]
Exploiting problem structure: Basis matrix inversion

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

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

- $W_i^B$ are columns corresponding to $n_i^B$ basic variables in scenario $i$
- $T_i^B$ are columns corresponding to $n_0^B$ basic first stage decisions
Inversion of the basis matrix $B$ is key to revised simplex efficiency

For column-linked BALP problems

\[
B = \begin{bmatrix}
W_1^B & T_1^B \\
\vdots & \vdots \\
W_N^B & T_N^B \\
\end{bmatrix}
\]

$B$ is nonsingular so

- $W_i^B$ are “tall”: full column rank
- $[W_i^B \quad T_i^B]$ 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

\[
B = \begin{bmatrix}
S_1 & & C_1 \\
. & \ddots & . \\
S_N & & C_N \\
R_1 & \ldots & R_N \\
\end{bmatrix}
= \begin{bmatrix}
S & \\
C \\
\end{bmatrix}
= \begin{bmatrix}
S & \\
R \\
\end{bmatrix}
\begin{bmatrix}
C \\
V \\
\end{bmatrix}
\]

Specifically

- \( L_i U_i = S_i \) of dimension \( n_i^B \)
- \( \hat{C}_i = L_i^{-1} C_i \)
- \( \hat{R}_i = R_i U_i^{-1} \)
- LU factors of the Schur complement \( M = V - RS^{-1} C \) of dimension \( n_0^B \)

Scope for parallelism since each GE applied to \( [W_i^B \mid T_i^B] \) is independent
Exploiting problem structure: Solving $Bx = b$

FTRAN for $Bx = b$

Solve

$$\begin{bmatrix} S & C \\ R & V \end{bmatrix} \begin{bmatrix} x^* \\ x_0 \end{bmatrix} = \begin{bmatrix} b^* \\ b_0 \end{bmatrix}$$

as

1. $L_i y_i = b_i, \ i = 1, \ldots, N$
2. $z_i = \hat{R}_i y_i, \ i = 1, \ldots, N$
3. $z = b_0 - \sum_{i=1}^{N} z_i$
4. $M x_0 = z$
5. $U_i x_i = y_i - \hat{C}_i x_0, \ i = 1, \ldots, N$

- Appears to be dominated by parallelizable
  - Solves $L_i y_i = b_i$ and $U_i x_i = y_i - \hat{C}_i x_0$
  - Products $\hat{R}_i y_i$ and $\hat{C}_i x_0$
- Curse of exploiting hyper-sparsity
  - In simplex, $b^*$ is from constraint column
    $$\begin{bmatrix} t_{1q} \\ \vdots \\ t_{Nq} \end{bmatrix}$$
    Either $w_{iq}$ or, more likely,
    $$\begin{bmatrix} 0 \\ w_{iq} \\ 0 \end{bmatrix}$$
  - In latter case, the $y_i$ inherit structure
    - Only one $L_i y_i = w_{iq}$
    - Only one $\hat{R}_i y_i$
- Less scope for parallelism than anticipated
Exploiting problem structure: Solving $B^T x = b$

**BTRAN for $B^T x = b$**

Solve

\[
\begin{bmatrix}
S^T & R^T \\
C^T & V^T
\end{bmatrix}
\begin{bmatrix}
x_* \\
x_0
\end{bmatrix}
= \begin{bmatrix} b_* \\ b_0 \end{bmatrix}
\]

as

1. $U_i^T y_i = b_i$, $i = 1, \ldots, N$
2. $z_i = \hat{C}_i^T y_i$, $i = 1, \ldots, N$
3. $z = b_0 - \sum_{i=1}^{N} z_i$
4. $M^T x_0 = z$
5. $L_i^T x_i = y_i - \hat{R}_i^T x_0$, $i = 1, \ldots, N$

- Appears to be dominated by parallelizable
  - Solves $U_i^T y_i = b_i$ and $L_i^T x_i = y_i - \hat{R}_i^T x_0$
  - Products $\hat{C}_i^T y_i$ and $\hat{R}_i^T x_0$
- Curse of exploiting hyper-sparsity
  - In simplex, $b = e_p$
  - At most one solve $U_i^T y_i = b_i$
  - At most one $\hat{C}_i^T y_i$
- Less scope for parallelism than anticipated
Exploiting problem structure: Forming $\pi_p^T N$

- PRICE forms

$$\begin{bmatrix}
\pi_1^T & \pi_2^T & \ldots & \pi_N^T & \pi_0^T
\end{bmatrix}
\begin{bmatrix}
W_1^N & T_1^N \\
W_2^N & T_2^N \\
\vdots & \vdots \\
W_N^N & T_N^N \\
A^N & \sum_{i=1}^{N} \pi_i^T T_i^N
\end{bmatrix}
= \begin{bmatrix}
\pi_1^T W_1^N & \pi_2^T W_2^N & \ldots & \pi_N^T W_N^N & \pi_0^T A^N + \sum_{i=1}^{N} \pi_i^T T_i^N
\end{bmatrix}$$

- Dominated by parallelizable products $\pi_i^T W_i^N$ and $\pi_i^T T_i^N$
## Results: Stochastic LP test problems

<table>
<thead>
<tr>
<th>Test Problem</th>
<th>1st Stage</th>
<th>2nd-Stage Scenario</th>
<th>Nonzero Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$n_0$</td>
<td>$m_0$</td>
<td>$n_i$</td>
</tr>
<tr>
<td>Storm</td>
<td>121</td>
<td>185</td>
<td>1,259</td>
</tr>
<tr>
<td>SSN</td>
<td>89</td>
<td>1</td>
<td>706</td>
</tr>
<tr>
<td>UC12</td>
<td>3,132</td>
<td>0</td>
<td>56,532</td>
</tr>
<tr>
<td>UC24</td>
<td>6,264</td>
<td>0</td>
<td>113,064</td>
</tr>
</tbody>
</table>

- 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

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

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

<table>
<thead>
<tr>
<th></th>
<th>Storm</th>
<th>SSN</th>
<th>UC12</th>
<th>UC24</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>4</td>
<td>3.6</td>
<td>3.5</td>
<td>2.7</td>
<td>3.0</td>
</tr>
<tr>
<td>8</td>
<td>7.3</td>
<td>7.5</td>
<td>6.1</td>
<td>5.3</td>
</tr>
<tr>
<td>16</td>
<td>13.6</td>
<td>15.1</td>
<td>8.5</td>
<td>8.9</td>
</tr>
<tr>
<td>32</td>
<td>24.6</td>
<td>30.3</td>
<td>14.5</td>
<td></td>
</tr>
<tr>
<td>clp</td>
<td>8.5</td>
<td>6.5</td>
<td>2.4</td>
<td>0.7</td>
</tr>
</tbody>
</table>
Results: On Fusion cluster - larger instances

<table>
<thead>
<tr>
<th></th>
<th>Storm</th>
<th>SSN</th>
<th>UC12</th>
<th>UC24</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scenarios</td>
<td>32,768</td>
<td>32,768</td>
<td>512</td>
<td>256</td>
</tr>
<tr>
<td>Variables</td>
<td>41,255,033</td>
<td>23,134,297</td>
<td>28,947,516</td>
<td>28,950,648</td>
</tr>
<tr>
<td>Constraints</td>
<td>17,301,689</td>
<td>5,734,401</td>
<td>30,431,232</td>
<td>30,431,232</td>
</tr>
</tbody>
</table>
### Speed-up of PIPS-S relative to 1-core PIPS-S and 1-core clp

<table>
<thead>
<tr>
<th></th>
<th>storm</th>
<th>ssn</th>
<th>UC12</th>
<th>UC24</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>15</td>
<td>19</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>16</td>
<td>52</td>
<td>45</td>
<td>14</td>
<td>12</td>
</tr>
<tr>
<td>32</td>
<td>117</td>
<td>103</td>
<td>26</td>
<td>22</td>
</tr>
<tr>
<td>64</td>
<td>152</td>
<td>181</td>
<td>44</td>
<td>41</td>
</tr>
<tr>
<td>128</td>
<td>202</td>
<td>289</td>
<td>60</td>
<td>64</td>
</tr>
<tr>
<td>256</td>
<td>285</td>
<td>383</td>
<td>70</td>
<td>80</td>
</tr>
<tr>
<td>clp</td>
<td>299</td>
<td>45</td>
<td>67</td>
<td>68</td>
</tr>
</tbody>
</table>
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 (≥ 1024 Blue Gene cores)
- Runs from an advanced basis

<table>
<thead>
<tr>
<th>Cores</th>
<th>Iterations</th>
<th>Time (h)</th>
<th>Iter/sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>1024</td>
<td>Exceeded execution time limit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2048</td>
<td>82,638</td>
<td>6.14</td>
<td>3.74</td>
</tr>
<tr>
<td>4096</td>
<td>75,732</td>
<td>5.03</td>
<td>4.18</td>
</tr>
<tr>
<td>8192</td>
<td>86,439</td>
<td>4.67</td>
<td>5.14</td>
</tr>
</tbody>
</table>
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