Review of Linear Programming Software

Peter He, Yang Li, Zhenghue Nie, Nael El Shawwa

Instructor: Prof. Tamas Terlaky
School of Computational Engineering and Science
McMaster University

Jan. 12, 2007
Outline

1. Introduction to LP/QP/MIP
   - Linear Programming
   - (Mixed) Integer Programming
   - Quadratic Programming

2. LP Software
   - Introduction
   - CPLEX
   - XPRESS-MP
   - MOSEK
   - LPSOL
   - CLP

3. Analysis and Comparison
   - Benchmark Analysis
   - Comparison
Outline

1. Introduction to LP/QP/MIP
   - Linear Programming
   - (Mixed) Integer Programming
   - Quadratic Programming

2. LP Software
   - Introduction
   - CPLEX
   - XPRESS-MP
   - MOSEK
   - LPSOL
   - CLP

3. Analysis and Comparison
   - Benchmark Analysis
   - Comparison
What are LP?

- **Standard LP form:**
  \[
  \begin{align*}
  \min \quad & c^T x \\
  \text{s.t.} \quad & Ax = b \\
  \quad & x \geq 0
  \end{align*}
  \]

- **Practical form:**
  \[
  \begin{align*}
  \min \quad & c^T x \\
  \text{s.t.} \quad & b \leq Ax \leq \bar{b} \\
  \quad & l \leq x \leq u
  \end{align*}
  \]

- **Main Methods:**
  - Simplex Methods
  - Interior Point Methods
What are LP?

- Standard LP form:

\[
\begin{align*}
\text{min} & \quad c^T x \\
\text{s.t.} & \quad Ax = b \\
& \quad x \geq 0
\end{align*}
\]

- Practical form:

\[
\begin{align*}
\text{min} & \quad c^T x \\
\text{s.t.} & \quad b \leq Ax \leq \bar{b} \\
& \quad l \leq x \leq u
\end{align*}
\]

- Main Methods:
  - Simplex Methods
  - Interior Point Methods
What are LP?

- **Standard LP form:**

  \[
  \begin{align*}
  \min & \quad c^T x \\
  \text{s.t.} & \quad Ax = b \\
  & \quad x \geq 0
  \end{align*}
  \]

- **Practical form:**

  \[
  \begin{align*}
  \min & \quad c^T x \\
  \text{s.t.} & \quad \underline{b} \leq Ax \leq \bar{b} \\
  & \quad l \leq x \leq u
  \end{align*}
  \]

- **Main Methods:**
  - Simplex Methods
  - Interior Point Methods
What are LP?

- **Standard LP form:**
  \[
  \begin{align*}
  \text{min} & \quad c^T x \\
  \text{s.t.} & \quad Ax = b \\
  & \quad x \geq 0
  \end{align*}
  \]

- **Practical form:**
  \[
  \begin{align*}
  \text{min} & \quad c^T x \\
  \text{s.t.} & \quad b \leq Ax \leq \bar{b} \\
  & \quad l \leq x \leq u
  \end{align*}
  \]

- **Main Methods:**
  - Simplex Methods
  - Interior Point Methods
What are LP?

- **Standard LP form:**
  \[
  \min \quad c^T x \\
  s.t. \quad Ax = b \\
  \quad x \geq 0
  \]

- **Practical form:**
  \[
  \min \quad c^T x \\
  s.t. \quad b \leq Ax \leq \bar{b} \\
  \quad l \leq x \leq u
  \]

- **Main Methods:**
  - Simplex Methods
  - Interior Point Methods
Simplex Method (Pivot Method)

- $Ax = b$ defines a polytope, namely, Simplex
- Jump from vertex to vertex
- Perform well for small and medium scale, but bad for large scale problem, $O(2^n)$
- Efficient for MIP
- Degeneracy

Source: www.cas.mcmaster.ca/terlaky/htm/talks/pivot-vs-IPM.pdf
Ax = b defines a polytope, namely, Simplex

- Jump from vertex to vertex
- Perform well for small and medium scale, but bad for large scale problem, \( O(2^n) \)
- Efficient for MIP
- Degeneracy

Source:
www.cas.mcmaster.ca/terlaky/htm/talks/pivot-vs-IPM.pdf
Simplex Method (Pivot Method)

- $Ax = b$ defines a polytope, namely, Simplex
- Jump from vertex to vertex
- Perform well for small and medium scale, but bad for large scale problem, $O(2^n)$
- Efficient for MIP
- Degeneracy

Source:
www.cas.mcmaster.ca/terlaky/htm/talks/pivot-vs-IPM.pdf
Simplex Method (Pivot Method)

- $Ax = b$ defines a polytope, namely, Simplex
- Jump from vertex to vertex
- Perform well for small and medium scale, but bad for large scale problem, $O(2^n)$
- Efficient for MIP
- Degeneracy

Source: www.cas.mcmaster.ca/terlaky/htm/talks/pivot-vs-IPM.pdf
Simplex Method (Pivot Method)

- \( Ax = b \) defines a polytope, namely, Simplex
- Jump from vertex to vertex
- Perform well for small and medium scale, but bad for large scale problem, \( O(2^n) \)
- Efficient for MIP
- Degeneracy

Source:
www.cas.mcmaster.ca/terlaky/htm/talks/pivot-vs-IPM.pdf
Interior Point Methods

\[ \begin{align*}
Ax & = b \\
A^T y + s & = c \\
x^T s & = \mu e
\end{align*} \]

follow the central path in the feasible region

Source: www.cas.mcmaster.ca/terlaky/htm/talks/pivot-vs-IPM.pdf
Interior Point Methods

$Ax = b$

$A^T y + s = c$

$x^T s = \mu e$

follow the central path in the feasible region

Source:
www.cas.mcmaster.ca/terlaky/htm/talks/pivot-vs-IPM.pdf
Interior Point Methods

- Very efficient for large-scale problem, total computational complexity $O(n^3L)$
- Bad performance in MIP
- No troubles caused by degeneracy

Source:
www.cas.mcmaster.ca/terlaky/htm/talks/pivot-vs-IPM.pdf
Interior Point Methods

- Very efficient for large-scale problem, total computational complexity $O(n^3 L)$
- Bad performance in MIP
- No troubles caused by degeneracy

Source:
www.cas.mcmaster.ca/terlaky/htm/talks/pivot-vs-IPM.pdf
Interior Point Methods

- Very efficient for large-scale problem, total computational complexity $O(n^3 L)$
- Bad performance in MIP
- No troubles caused by degeneracy

Source:
www.cas.mcmaster.ca/terlaky/htm/talks/pivot-vs-IPM.pdf
Outline

1 Introduction to LP/QP/MIP
   - Linear Programming
   - (Mixed) Integer Programming
   - Quadratic Programming

2 LP Software
   - Introduction
   - CPLEX
   - XPRESS-MP
   - MOSEK
   - LPSOL
   - CLP

3 Analysis and Comparison
   - Benchmark Analysis
   - Comparison
What is (Mixed) Integer Programming

Integer Programming:

\[
\begin{align*}
\min & \quad c^T x \\
\text{s.t.} & \quad Ax = b \\
\quad & \quad x_i \in \mathbb{Z} \text{ or subset of } \mathbb{Z} \text{ for all } i
\end{align*}
\]

- Branch and Bound & Cutting Plane.
What is (Mixed) Integer Programming

Mixed Integer Programming:

\[
\begin{align*}
\text{min } & \quad c^T x \\
\text{s.t. } & \quad Ax = b \\
& \quad x_i \in \mathbb{Z} \text{ or subset of } \mathbb{Z} \text{ for some } i
\end{align*}
\]

- Branch and Bound & Cutting Plane.
Outline

1. Introduction to LP/QP/MIP
   - Linear Programming
   - (Mixed) Integer Programming
   - Quadratic Programming

2. LP Software
   - Introduction
   - CPLEX
   - XPRESS-MP
   - MOSEK
   - LPSOL
   - CLP

3. Analysis and Comparison
   - Benchmark Analysis
   - Comparison
What is Quadratic Programming?

Standard QP form:

$$\min c^T x + \frac{1}{2} x^T Q x$$
\[\text{s.t. } Ax = b\]
\[x \geq 0\]

- If $Q$ is semi-definite, the problem is convex. Otherwise, local convergence.
What is Quadratic Programming?

Standard QP form:

$$\min c^T x + \frac{1}{2} x^T Q x$$

s.t. $Ax = b$

$x \geq 0$

- If $Q$ is semi-definite, the problem is convex. Otherwise, local convergence.
Outline

1. Introduction to LP/QP/MIP
   - Linear Programming
   - (Mixed) Integer Programming
   - Quadratic Programming

2. LP Software
   - Introduction
     - CPLEX
     - XPRESS-MP
     - MOSEK
     - LPSOL
     - CLP

3. Analysis and Comparison
   - Benchmark Analysis
   - Comparison
Developers’ View

- Programs
  - Algorithms
  - Modeling systems

- Cost
  - Free codes
  - Commercial products - providing the demo version

- Others
  - products type
  - interface
  - platforms and/or operating systems
  - development environment
  - modeling system
  - parallel
  - ...

---

He, Li, Nie, Shawwa

Linear Programming
Outline

1. Introduction to LP/QP/MIP
   - Linear Programming
   - (Mixed) Integer Programming
   - Quadratic Programming

2. LP Software
   - Introduction
   - CPLEX
   - XPRESS-MP
   - MOSEK
   - LPSOL
   - CLP

3. Analysis and Comparison
   - Benchmark Analysis
   - Comparison
Overview of CPLEX

- Created by Robert E. Bixby using C language in 1987
- Been Acquired by ILOG, inc. in 1997
- Widespread
  - 95% of papers that mention a solver mention CPLEX at any conference where papers are presented
  - Standard solver in supply-chain applications; Industry leaders including SAP, Oracle, i2, JD Edwards, and Manugistics
  - Most of the major airlines including United, American, Delta, Continental, Northwest, and Southwest
- 2004 INFORMS Impact Award
- Latest version: 10.1 (Parallel, Indictor Constraints, etc.)
Problems Covered

- Linear Programming
- Mixed Integer Linear Programming
- Quadratic Programming
- Mixed Integer Quadratic Programming
- Quadratic Constrained Programming
- Mixed Integer Quadratic Constrained Programming

Millions of constraints and variables
Algorithms

LP:
- Primal simplex
- Dual simplex
- Network simplex
- Primal/dual log barrier

QP:
- Primal simplex
- Dual simplex
- Primal dual log barrier

QCP:
- Primal dual log barrier
ILOG Optimization Suite

- Simplex Optimizers (Primal, Dual, Network) for LP and QP
- Barrier Optimizer for LP, QP and QCP
- Mixed Integer Optimizer for MIP, MIQP and MIQCP
- CPLEX Interactive Optimizer
- CPLEX Component Libraries
  - CPLEX Callable Library (C and VB6 APIs)
  - ILOG Concert Technology (C++, Java, .NET APIs)

Embedding CPLEX

- CPLEX Callable Library uses matrices to represent a problem
- ILOG Concert Technology uses objects and methods to represent a problem
- ILOG OPL Interfaces allow OPL models to be embedded inside an application

Embedding CPLEX

- CPLEX Callable Library uses matrices to represent a problem
- ILOG Concert Technology uses objects and methods to represent a problem
- ILOG OPL Interfaces allow OPL models to be embedded inside an application

Embedding CPLEX

- CPLEX Callable Library uses matrices to represent a problem
- ILOG Concert Technology uses objects and methods to represent a problem
- ILOG OPL Interfaces allow OPL models to be embedded inside an application

Modeling System, Platforms and Parallel

- Modeling System: AMPL, MPL, OPL Studio, and GAMS
- Platforms: PC/Windows, PC/Linux, Sun/Solaris, HP, IBM Power
- Parallel: Yes
Outline

1. Introduction to LP/QP/MIP
   - Linear Programming
   - (Mixed) Integer Programming
   - Quadratic Programming

2. LP Software
   - Introduction
   - CPLEX
   - XPRESS-MP
   - MOSEK
   - LPSOL
   - CLP

3. Analysis and Comparison
   - Benchmark Analysis
   - Comparison
Overview of Xpress-MP

- Developed by Dash Optimization
- Used in industry:
  - Finance and investment
  - Italcementi Group - Cement plant supply chain optimisation
  - Honeywell - Flowsheet Decomposition Heuristic for Scheduling
  - Power Optimization Limited - Generator Scheduling in the Electricity Industry
Problems covered

- LP, QP
- MIP, MIQP
Algorithms

- Primal and Dual simplex methods
- Interior points method for LP and QP
- Branch and Bound techniques to solve MIP and MIQP
- Techniques to reduce problem size
Platforms and OS

- Windows
- Linux, Unix
- Mac (PowerPC)
- HP, AIX
Modeling Systems

- Mosel
- AMPL
- GAMS
- Frontline
- MPL
Interfaces

- command line
- C/ C++
- Java
- Fortran
- VB6, .NET
- ODBC I/O driver for external spreadsheets and databases
Other Features

- Size: no fixed limit
- Input form: Industry standard formats - LP, MPS
- Supports parallel computing
- Available add-ons to solve other problem types - stochastic problems, NLP
- Total user control of optimization process
- Fast re-solve time
- Development tools - GUI development, problem formulation, solving and analysis
Cost

- $6,000 per license (basic)
- $22,500 per license (bundle)
- $1,000 per additional licence
- Academic: 90% off, free with academic partnership program
- Other Mosel and Optimizer add-ons available separately

Prices as of January 2007 - Dash Optimization Price List
1 Introduction to LP/QP/MIP
   - Linear Programming
   - (Mixed) Integer Programming
   - Quadratic Programming

2 LP Software
   - Introduction
   - CPLEX
   - XPRESS-MP
   - MOSEK
   - LPSOL
   - CLP

3 Analysis and Comparison
   - Benchmark Analysis
   - Comparison
Overviews of MOSEK

- Erling D. Andersen and Knud D. Andersen
- Implement language: C
- Latest Version: 4.0
Overviews of MOSEK

- Erling D. Andersen and Knud D. Andersen
- Implement language: C
- Latest Version: 4.0
Overviews of MOSEK

- Erling D. Andersen and Knud D. Andersen
- Implement language: C
- Latest Version: 4.0
Problems covered

- Linear optimization
- Convex quadratic optimization.
- Quadratically constrained convex optimization.
- Conic quadratic optimization (also known as second order optimization).
- Convex optimization.
- Geometric optimization (posynomial case).
- Handle integer valued variables in linearly and quadratically constrained optimization problems.
Algorithms

- Interior-point optimizer for all continuous problems.
- Primal or dual simplex optimizer for linear problems.
- Conic interior-point optimizer for conic quadratic problems.
- Mixed-integer optimizer based on a branch and cut technology.
Platforms and OS

- Linux 32 and 64 bit
- MAC OSX
- Solaris 32 and 64 bit
- Windows 32 and 64 bit
Modeling Systems

- AMPL
- GAMS
- AIMMS
Interfaces

- C/C++ API
- Command line interface
- Java API
- Microsoft. NET API
- MATLAB
## Costs:

<table>
<thead>
<tr>
<th>Code</th>
<th>Version</th>
<th>Problem size</th>
<th>Problem types</th>
<th>Cones</th>
<th>Platform A</th>
<th>Platform B</th>
</tr>
</thead>
<tbody>
<tr>
<td>PTD</td>
<td>Demo</td>
<td>150 x 300</td>
<td>Nonlinear</td>
<td>Yes</td>
<td>free</td>
<td>free</td>
</tr>
<tr>
<td>PTS</td>
<td>Standard</td>
<td>Unrestricted</td>
<td>Quadratic</td>
<td>Linear</td>
<td>1750US$</td>
<td>2100US$</td>
</tr>
</tbody>
</table>

Table 1: Prices for optimization tools

<table>
<thead>
<tr>
<th>Code</th>
<th>Version</th>
<th>Problem size</th>
<th>Problem types</th>
<th>Cones</th>
<th>Platform A</th>
<th>Platform B</th>
</tr>
</thead>
<tbody>
<tr>
<td>PTON</td>
<td>Nonlinear</td>
<td>Unrestricted</td>
<td>Nonlinear</td>
<td>No</td>
<td>1500US$</td>
<td>1800US$</td>
</tr>
<tr>
<td>PTOC</td>
<td>Conic</td>
<td>Unrestricted</td>
<td>Linear</td>
<td>Yes</td>
<td>1500US$</td>
<td>1800US$</td>
</tr>
<tr>
<td>PTOM</td>
<td>Mixed integer</td>
<td>Unrestricted</td>
<td>Quadratic</td>
<td>See notes</td>
<td>4000US$</td>
<td>4800US$</td>
</tr>
</tbody>
</table>

Table 2: Prices for optional optimization tools extensions (PTS is always required)

- **A**: Linux and Windows 32 bit
- **B**: Other Platforms, like Solaris, Mac OS, Linux and Windows 64 bit.
- Additional maintenance fee charged
- Discount for academic use: 70% to 90%
Other Features

- Input format: MPS, LP, MBT (MOSEK binary task format)
- Parallel Computing
Outline

1. Introduction to LP/QP/MIP
   - Linear Programming
   - (Mixed) Integer Programming
   - Quadratic Programming

2. LP Software
   - Introduction
   - CPLEX
   - XPRESS-MP
   - MOSEK
   - LPSOL
   - CLP

3. Analysis and Comparison
   - Benchmark Analysis
   - Comparison
Overviews of LPSOL

- Michel Berkelaar and Jeroen Dirks
- Implement language: ANSI C
- Latest Version: Version 5.5
Overviews of LPSOL

- Michel Berkelaar and Jeroen Dirks
- Implement language: ANSI C
- Latest Version: Version 5.5
Overviews of LPSOL

- Michel Berkelaar and Jeroen Dirks
- Implement language: ANSI C
- Latest Version: Version 5.5
Problems covered

- (Large-scale) Linear Programming
- Mixed Integer Programming
Algorithms

- Revised Simplex Method for Linear Programming
- A Branch and Bound Method for Mixed Integer Programming
Platforms and OS

- Linux/Unix
- Windows
- MAC OS
Modeling Systems

- AMPL
- GAMS
- AIMMS
Introduction

CPLEX
XPRESS-MP
MOSEK
LPSOL
CLP

Interfaces

- C/C++ API
- Visual Basic
- Microsoft .NET
- Delphi
- Java
- MATLAB
- Excel
Other Features

- Sizes: Basically, no limit on model size
- Input form: LP format and MPS
- Parallel Computing: No
- Prices: Free
Introduction to LP/QP/MIP

1. Introduction to LP/QP/MIP
   - Linear Programming
   - (Mixed) Integer Programming
   - Quadratic Programming

2. LP Software
   - Introduction
   - CPLEX
   - XPRESS-MP
   - MOSEK
   - LPSOL
   - CLP

3. Analysis and Comparison
   - Benchmark Analysis
   - Comparison
Introduction to CLP

CLP means Coin OR Linear Programming. It is a high quality simplex code under the term of the Common Public License.

- Author: John Forrest; Published Time: 2002; The first version: version 0.90; Current version: Clp-1.3.3 (http://www.coin-or.org/)
- CLP is written in C++ and it is primarily intended to be used as a callable library.
Introduction to CLP

- CLP means Coin OR Linear Programming. It is a high quality simplex code under the term of the Common Public License.
- Author: John Forrest; Published Time: 2002; The first version: version 0.90; Current version: Clp-1.3.3 (http://www.coin-or.org/)
- CLP is written in C++ and it is primarily intended to be used as a callable library.
Introduction to CLP

- CLP means Coin OR Linear Programming. It is a high quality simplex code under the term of the Common Public License.
- Author: John Forrest; Published Time: 2002; The first version: version 0.90; Current version: Clp-1.3.3 (http://www.coin-or.org/)
- CLP is written in C++ and it is primarily intended to be used as a callable library.
What can CLP solve?

- **Linear Optimization**
  - Regular
  - Large Scale
    - It has been proved that LP with 1.5 million constraints still keeps reliability.
  - Sparse Coefficient Matrixes
  - Boolean Matrixes
  - Network Matrixes

- Some Nonlinear Optimization: Such as
  - Quadratic Programming
  - Piecewise Linear Convex function as the objective function
What can CLP solve?

- Linear Optimization
  - Regular
  - Large Scale
    - It has been proved that LP with 1.5 million constraints still keeps reliability.
  - Sparse Coefficient Matrixes
  - Boolean Matrixes
  - Network Matrixes

- Some Nonlinear Optimization: Such as
  - Quadratic Programming
  - Piecewise Linear Convex function as the objective function
What Are Algorithms Being Used?

- Prime simplex method: the column pivot choice
- Dual simplex method: Dantzig and Steepest edge row pivot choice
- CLP barrier method: solving convex QPs as well as LPs
What Are Algorithms Being Used?

- Prime simplex method: the column pivot choice
- Dual simplex method: Dantzig and Steepest edge row pivot choice
- CLP barrier method: solving convex QPs as well as LPs
What Are Algorithms Being Used?

- Prime simplex method: the column pivot choice
- Dual simplex method: Dantzig and Steepest edge row pivot choice
- CLP barrier method: solving convex QPs as well as LPs
CLP often reads or loads MPS file. MPS (Mathematical Programming System) is a file format for LO and MIP.
What Platforms Are Available?

- Linux using g++ version 3.1.1 (or later)
- Windows using Microsoft Visual C++ 6 (or later)
- Windows using cygwin
- AIX using xIC (not supported in the current Makefile)
Features of CLP

- Apparently as reliable as OSL
- Slightly slower that OSL on calculation speed
- Barrier code is not as mature as the simplex code
- But, CPL is easier than OSL, free or low cost, and the high ratio of the function to the price.
Benchmark Analysis

Comparison

<table>
<thead>
<tr>
<th>problem</th>
<th>rows</th>
<th>columns</th>
<th>nonzeros</th>
<th>CLP</th>
<th>CPLEX-B</th>
<th>CPLEX-D/P</th>
<th>MOSEK</th>
</tr>
</thead>
<tbody>
<tr>
<td>cont1</td>
<td>160793</td>
<td>48398</td>
<td>399991</td>
<td>9138</td>
<td>952</td>
<td>2174/1530</td>
<td>5855</td>
</tr>
<tr>
<td>cont11</td>
<td>160793</td>
<td>80396</td>
<td>439989</td>
<td>f</td>
<td>1983</td>
<td>/</td>
<td>45727</td>
</tr>
<tr>
<td>cont4</td>
<td>160793</td>
<td>48398</td>
<td>398399</td>
<td>3812</td>
<td>3685</td>
<td>2929/736</td>
<td>686</td>
</tr>
<tr>
<td>cont11_1</td>
<td>1918399</td>
<td>641598</td>
<td>5752001</td>
<td>3171</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cont11_1</td>
<td>1468509</td>
<td>981396</td>
<td>4488301</td>
<td>f</td>
<td></td>
<td></td>
<td>2081</td>
</tr>
<tr>
<td>dan03mip_1p</td>
<td>3283</td>
<td>13873</td>
<td>79656</td>
<td>123</td>
<td>16</td>
<td>36/23</td>
<td>14</td>
</tr>
<tr>
<td>dbic1</td>
<td>43200</td>
<td>183235</td>
<td>1038761</td>
<td>428</td>
<td>112</td>
<td>116/23</td>
<td>129</td>
</tr>
<tr>
<td>df1001</td>
<td>6872</td>
<td>12230</td>
<td>41873</td>
<td>29</td>
<td>14</td>
<td>20/46</td>
<td>12</td>
</tr>
<tr>
<td>fome12</td>
<td>24265</td>
<td>48928</td>
<td>167492</td>
<td>213</td>
<td>169</td>
<td>141/407</td>
<td>57</td>
</tr>
<tr>
<td>fome13</td>
<td>48569</td>
<td>97848</td>
<td>334984</td>
<td>330</td>
<td>87</td>
<td>669/1170</td>
<td>88</td>
</tr>
<tr>
<td>gen4</td>
<td>1538</td>
<td>4297</td>
<td>118174</td>
<td>79</td>
<td>31</td>
<td>2/71</td>
<td>24</td>
</tr>
<tr>
<td>ken-18</td>
<td>165128</td>
<td>154699</td>
<td>512719</td>
<td>17</td>
<td>11</td>
<td>12/68</td>
<td>21</td>
</tr>
<tr>
<td>130</td>
<td>2782</td>
<td>15388</td>
<td>64790</td>
<td>29</td>
<td>f</td>
<td>28/80</td>
<td>3</td>
</tr>
<tr>
<td>lp22</td>
<td>2959</td>
<td>13434</td>
<td>78994</td>
<td>19</td>
<td>6</td>
<td>40/72</td>
<td>7</td>
</tr>
<tr>
<td>mod2</td>
<td>35665</td>
<td>31728</td>
<td>220116</td>
<td>161</td>
<td>10</td>
<td>82/192</td>
<td>29</td>
</tr>
<tr>
<td>neos</td>
<td>479120</td>
<td>36786</td>
<td>1084461</td>
<td>2129</td>
<td>149</td>
<td>27/154</td>
<td>107</td>
</tr>
<tr>
<td>neos1</td>
<td>131582</td>
<td>1892</td>
<td>468804</td>
<td>107</td>
<td>30</td>
<td>808/16</td>
<td>35</td>
</tr>
<tr>
<td>neos2</td>
<td>132569</td>
<td>1560</td>
<td>552596</td>
<td>461</td>
<td>21</td>
<td>492/29</td>
<td>30</td>
</tr>
<tr>
<td>neos3</td>
<td>512209</td>
<td>6624</td>
<td>1542816</td>
<td>31033</td>
<td>281</td>
<td>3208/5915</td>
<td>562</td>
</tr>
<tr>
<td>nsc2</td>
<td>23004</td>
<td>14981</td>
<td>686396</td>
<td>4</td>
<td>69</td>
<td>2/2</td>
<td>50</td>
</tr>
<tr>
<td>nug15</td>
<td>6331</td>
<td>22275</td>
<td>118700</td>
<td>122</td>
<td>77</td>
<td>297/1278</td>
<td>111</td>
</tr>
<tr>
<td>nug20</td>
<td>15240</td>
<td>72650</td>
<td>304800</td>
<td>26947</td>
<td>1276</td>
<td>1490</td>
<td></td>
</tr>
<tr>
<td>nug08-3rd</td>
<td>19728</td>
<td>28448</td>
<td>139008</td>
<td>2047</td>
<td>1279</td>
<td>2367/</td>
<td>1925</td>
</tr>
<tr>
<td>pds-40</td>
<td>68645</td>
<td>212859</td>
<td>605678</td>
<td>44</td>
<td>142</td>
<td>37/166</td>
<td>183</td>
</tr>
<tr>
<td>pds-100</td>
<td>156244</td>
<td>505486</td>
<td>3985530</td>
<td>238</td>
<td>740</td>
<td>168/1586</td>
<td>1951</td>
</tr>
<tr>
<td>gap12</td>
<td>3193</td>
<td>8856</td>
<td>44244</td>
<td>42</td>
<td>10</td>
<td>203/112</td>
<td>15</td>
</tr>
<tr>
<td>gap15</td>
<td>6331</td>
<td>22275</td>
<td>118700</td>
<td>151</td>
<td>74</td>
<td>2871/1230</td>
<td>107</td>
</tr>
<tr>
<td>rail4284</td>
<td>4284</td>
<td>1092610</td>
<td>1237358</td>
<td>260</td>
<td>239</td>
<td>5235/6476</td>
<td>386</td>
</tr>
<tr>
<td>rlprim</td>
<td>58867</td>
<td>8052</td>
<td>265075</td>
<td>6</td>
<td>4</td>
<td>1/6</td>
<td>5</td>
</tr>
<tr>
<td>self</td>
<td>960</td>
<td>7364</td>
<td>1148845</td>
<td>117</td>
<td>64</td>
<td>113/100</td>
<td>203</td>
</tr>
<tr>
<td>sgp596</td>
<td>246078</td>
<td>308634</td>
<td>902275</td>
<td>24</td>
<td>11</td>
<td>3/2</td>
<td>18</td>
</tr>
<tr>
<td>spal_004</td>
<td>10203</td>
<td>321696</td>
<td>46167908</td>
<td>4669</td>
<td>4669</td>
<td></td>
<td>8020</td>
</tr>
<tr>
<td>math596</td>
<td>5257</td>
<td>462176</td>
<td>539702</td>
<td>263</td>
<td>208</td>
<td>123/6541</td>
<td>13</td>
</tr>
</tbody>
</table>

Run on a Linux-PC(3.2GHz P4, 4GB RDRAM, Linx-2.6)

Version: CPLEX-10.0; MOSEK-4.0.0.32; CLP-1.03.03;

Times are user times in seconds including input and crossover to a feasible basis for all codes.

B/D/P:barrier/dual/primal simplex

source:http://plato.asu.edu/ftp/
Parallel CPLEX on MIP problems

CPLEX-10.1 was run in default mode on a single and on a dual processor 2.4GHz Opteron (64-bit, Linux), and a 2.2GHz dual-core Opteron (64-bit, Linux)

<table>
<thead>
<tr>
<th>class</th>
<th>problem</th>
<th>c</th>
<th>Opter-1</th>
<th>Opter-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>MILP</td>
<td>bienst2</td>
<td>y</td>
<td>2529</td>
<td>608</td>
</tr>
<tr>
<td></td>
<td>lrn</td>
<td>y</td>
<td>114</td>
<td>85</td>
</tr>
<tr>
<td></td>
<td>mas74</td>
<td>y</td>
<td>897</td>
<td>441</td>
</tr>
<tr>
<td></td>
<td>neos13</td>
<td>y</td>
<td>2073</td>
<td>1694</td>
</tr>
<tr>
<td></td>
<td>neos5</td>
<td>y</td>
<td>1169</td>
<td>&gt;40000</td>
</tr>
<tr>
<td></td>
<td>seymour1</td>
<td>y</td>
<td>669</td>
<td>449</td>
</tr>
</tbody>
</table>

| MIQP  | ibienst1 | y | 2742    | 1330    |
|       | inug08   | y | 7973    | 4761    |
|       | iqap10   | y | 1679    | 457     |
|       | isqp     | y | 4755    | 2824    |

| MIQPP | ibienst1 | 3132 | 1878 |
|       | misc07   | 6460 | 3255 |
|       | imod011  | 7348 | 9463 |
|       | inug06-3rd | 6588 | 6890 |
|       | inug08   | 4221 | 2336 |
|       | iran13x13| 8756 | 3876 |
|       | Clay0304M| y  | 1278   | 630    |

Times given are elapsed CPU times in seconds. "c" problem convex. AMPL or MPS input.

source: http://plato.asu.edu/ftp/ser_par.html
- CPLEX parallel benchmark shows significant improvement with dual processors
- Some problems may have dense coefficient matrix → bad for parallel computing
- Commercial packages were more robust and efficient than open-source software
- BUT there are cases were CPLEX algorithms failed to converge while MOSEK and CLP did.
- There are tests where CPLEX has reported in-feasible solution while MOSEK converges
- IPM is more robust than Primal/Dual for large scale problems
- CPLEX parallel benchmark shows significant improvement with dual processors
- Some problems may have dense coefficient matrix → bad for parallel computing
- Commercial packages were more robust and efficient than open-source software
- BUT there are cases were CPLEX algorithms failed to converge while MOSEK and CLP did.
- There are tests where CPLEX has reported in-feasible solution while MOSEK converges
- IPM is more robust than Primal/Dual for large scale problems
- CPLEX parallel benchmark shows significant improvement with dual processors
- Some problems may have dense coefficient matrix → bad for parallel computing
- Commercial packages were more robust and efficient than open-source software
  - BUT there are cases were CPLEX algorithms failed to converge while MOSEK and CLP did.
  - There are tests where CPLEX has reported in-feasible solution while MOSEK converges
- IPM is more robust than Primal/Dual for large scale problems
• CPLEX parallel benchmark shows significant improvement with dual processors
• Some problems may have dense coefficient matrix $\rightarrow$ bad for parallel computing
• Commercial packages were more robust and efficient than open-source software
• BUT there are cases were CPLEX algorithms failed to converge while MOSEK and CLP did.
• There are tests where CPLEX has reported in-feasible solution while MOSEK converges
• IPM is more robust than Primal/Dual for large scale problems
- CPLEX parallel benchmark shows significant improvement with dual processors
- Some problems may have dense coefficient matrix → bad for parallel computing
- Commercial packages were more robust and efficient than open-source software
- BUT there are cases were CPLEX algorithms failed to converge while MOSEK and CLP did.
- There are tests where CPLEX has reported in-feasible solution while MOSEK converges
- IPM is more robust than Primal/Dual for large scale problems
- CPLEX parallel benchmark shows significant improvement with dual processors
- Some problems may have dense coefficient matrix → bad for parallel computing
- Commercial packages were more robust and efficient than open-source software
- BUT there are cases were CPLEX algorithms failed to converge while MOSEK and CLP did.
- There are tests where CPLEX has reported in-feasible solution while MOSEK converges
- IPM is more robust than Primal/Dual for large scale problems
Outline

1. Introduction to LP/QP/MIP
   - Linear Programming
   - (Mixed) Integer Programming
   - Quadratic Programming

2. LP Software
   - Introduction
   - CPLEX
   - XPRESS-MP
   - MOSEK
   - LPSOL
   - CLP

3. Analysis and Comparison
   - Benchmark Analysis
   - Comparison
<table>
<thead>
<tr>
<th>Name</th>
<th>Commercial</th>
<th>Academic</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPLEX</td>
<td>Call</td>
<td>$995</td>
<td>Free student</td>
</tr>
<tr>
<td>XPRESS-MP</td>
<td>$6,000</td>
<td>$600</td>
<td>Free student 30 day demo</td>
</tr>
<tr>
<td>MOSEK</td>
<td>$1,750</td>
<td>$175</td>
<td>Free student 15 day demo</td>
</tr>
</tbody>
</table>
## Platform Supported

<table>
<thead>
<tr>
<th>Product</th>
<th>CPLEX</th>
<th>XPRESS-MP</th>
<th>MOSEK solver engine</th>
<th>LPSOL</th>
<th>CLP</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC/Windows</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
</tr>
<tr>
<td>PC/Linux</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
</tr>
<tr>
<td>Sun/Solaris</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
</tr>
<tr>
<td>Mac/Mac OS</td>
<td>-</td>
<td>Power PC</td>
<td>-</td>
<td>y</td>
<td>y</td>
</tr>
<tr>
<td>Other</td>
<td>HP,IBM power</td>
<td>HP,AIX</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Shared Memory Parallel</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

(source: [http://lionhrtpub.com/orms/surveys/LP/LP-survey.html](http://lionhrtpub.com/orms/surveys/LP/LP-survey.html))
## Size of Problems solvable by the system

<table>
<thead>
<tr>
<th>Product</th>
<th>Internal Restrictions</th>
<th>Max No. of Constraints</th>
<th>Available Memory</th>
<th>Available Disk Space</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPLEX</td>
<td>-</td>
<td>-</td>
<td>y</td>
<td>-</td>
</tr>
<tr>
<td>XPRESS Solver Engine</td>
<td>-</td>
<td>-</td>
<td>y</td>
<td>-</td>
</tr>
<tr>
<td>MOSEK Solver Engine</td>
<td>-</td>
<td>-</td>
<td>y</td>
<td>-</td>
</tr>
<tr>
<td>LPSOL</td>
<td>-</td>
<td>-</td>
<td>y</td>
<td>-</td>
</tr>
<tr>
<td>CLP</td>
<td>-</td>
<td>-</td>
<td>y</td>
<td>-</td>
</tr>
</tbody>
</table>

*source: http://lionhrtpub.com/orms/surveys/LP/LP-survey.html*
## Algorithms

<table>
<thead>
<tr>
<th>Product</th>
<th>Primal Simplex</th>
<th>Dual Simplex</th>
<th>Interior Point</th>
<th>Branch-and-Cut</th>
<th>Branch-and-Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPLEX</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>-</td>
</tr>
<tr>
<td>XPRESS Solver Engine</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
</tr>
<tr>
<td>MOSEK Solver Engine</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>-</td>
</tr>
<tr>
<td>LPSOL</td>
<td>y</td>
<td>y</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CPL</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>-</td>
</tr>
</tbody>
</table>

*source: http://lionhrtpub.com/orms/surveys/LP/LP-survey.html*
Summary

- All of them can solve the LP programs, but commercial programs
  - provide more flexible interface;
  - support more modeling systems;
  - provide development tools;
  - provide parallel computation;

- Parallel computation provide a new way to improve the efficiency and solve large-scale problems, but is limited by the algorithm used and matrix sparsity.

- Which program is the fastest one?

- XPRESS: http://www.dashoptimization.com/
- MOSEK: http://www.mosek.com/
- CLP: http://www.coin-or.org/Clp/index.html
- http://www.informs.org/index.php?
Thanks!  
Questions?