AMPL, CPLEX, C Callable library and Concert Technology

Sujeevraja Sanjeevi

INFORMS Seminar, Texas A&M University

November 7, 2012

1. Introduction
2. CPLEX
   - First CPLEX Problem
   - Reading .lp and .mps files
3. AMPL
   - Introduction
   - First AMPL model
   - The MPS file
4. Callable Library and Concert Technology

CPLEX

What is CPLEX?

- Optimization solver
- used to solve LPs, MIPs, QPs
- One of the most commonly used solvers in the world
- Comprehensive libraries for interaction with
  - C (Callable library, callback functions)
  - C++ (Concert Technology)
  - Java
- Interaction with MATLAB and Python
- Other solvers: Xpress, Gurobi
- Code resources: COIN-OR

Interactive Optimizer

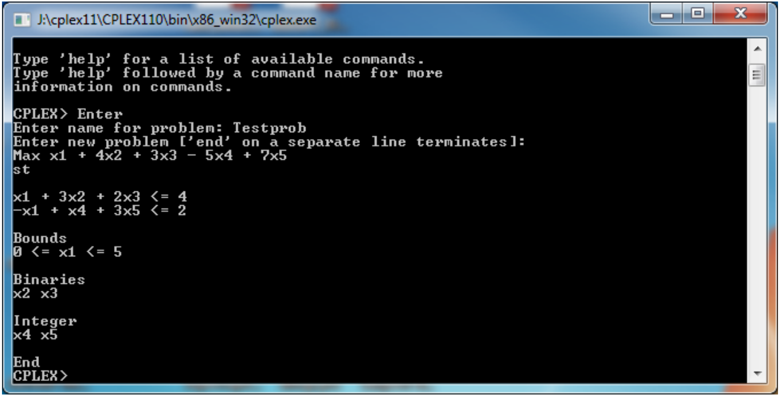
CPLEX location -
J:\CPLEX11\CPLEX110\ bin\x86_win32\cplex.exe

Model:

$$\begin{aligned} \text{Max } & x_1 + 4x_2 + 3x_3 - 5x_4 + 7x_5 \\ \text{s.t. } & x_1 + 3x_2 + 2x_3 \leq 4 \\ & -x_1 + x_4 + 3x_5 \leq 2 \\ & x_1 \text{ continuous}, x_2, x_3 \text{ binary}, x_4, x_5 \text{ integer} \\ & \text{All variables non-negative} \end{aligned}$$

Interactive Optimizer - Entering Problem

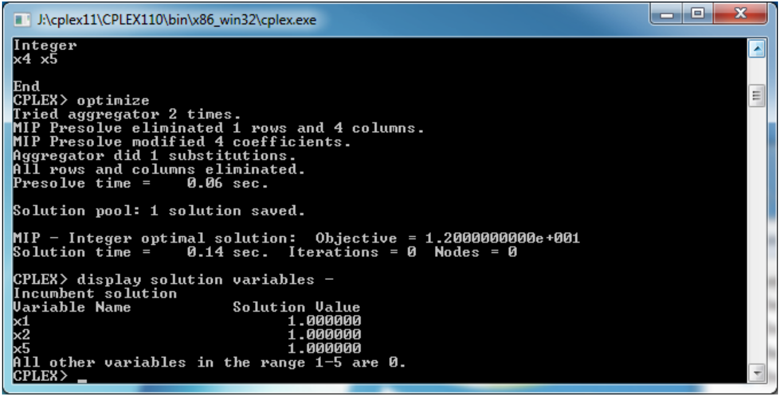
Entering the problem directly in CPLEX:



```
Type 'help' for a list of available commands.
Type 'help' followed by a command name for more
information on commands.

CPLEX> Enter
Enter name for problem: Testprob
Enter new problem ['end' on a separate line terminates]:
Max x1 + 4x2 + 3x3 - 5x4 + 7x5
st

x1 + 3x2 + 2x3 <= 4
-x1 + x4 + 3x5 <= 2

Bounds
0 <= x1 <= 5

Binaries
x2 x3

Integer
x4 x5

End
CPLEX>
```

Interactive Optimizer - Solution

Optimizing the problem and obtaining solution values:



```
Integer
x4 x5

End
CPLEX> optimize
Tried aggregator 2 times.
MIP Presolve eliminated 1 rows and 4 columns.
MIP Presolve modified 4 coefficients.
Aggregator did 1 substitutions.
All rows and columns eliminated.
Presolve time =    0.06 sec.

Solution pool: 1 solution saved.

MIP - Integer optimal solution:  Objective = 1.2000000000e+001
Solution time =    0.14 sec.  Iterations = 0  Nodes = 0

CPLEX> display solution variables -
Incumbent solution
Variable Name           Solution Value
x1                            1.000000
x2                            1.000000
x5                            1.000000
All other variables in the range 1-5 are 0.
CPLEX> _
```

File Formats

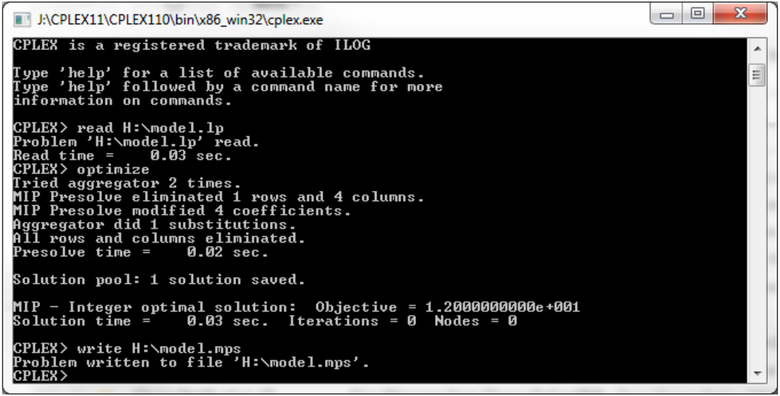
Formats used by most solvers:

- .lp - what we just used
- .mps - Mathematical Programming System format

Commands:

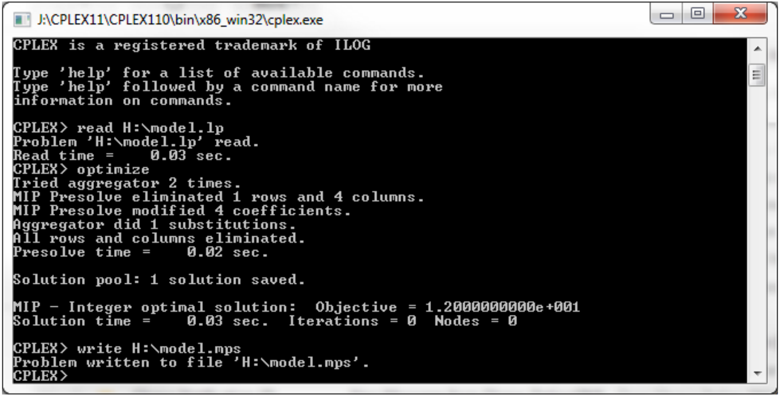
- read $< PATH >$ \filename.lp
- write $< PATH >$ \filename.mps

Reading .lp and .mps files



```
CPLEX is a registered trademark of ILOG

Type 'help' for a list of available commands.
Type 'help' followed by a command name for more
information on commands.

CPLEX> read H:\model.lp
Problem 'H:\model.lp' read.
Read time =    0.03 sec.
CPLEX> optimize
Tried aggregator 2 times.
MIP Presolve eliminated 1 rows and 4 columns.
MIP Presolve modified 4 coefficients.
Aggregator did 1 substitutions.
All rows and columns eliminated.
Presolve time =    0.02 sec.

Solution pool: 1 solution saved.

MIP - Integer optimal solution:  Objective = 1.2000000000e+001
Solution time =    0.03 sec.  Iterations = 0  Nodes = 0

CPLEX> write H:\model.mps
Problem written to file 'H:\model.mps'.
CPLEX>
```

Reading .lp and .mps files



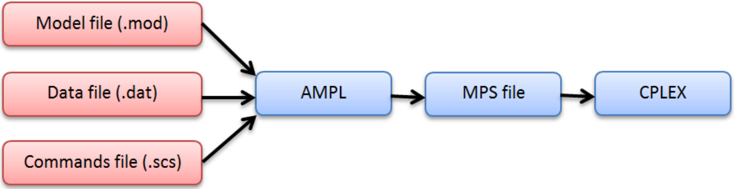
```
CPLEX is a registered trademark of ILOG

Type 'help' for a list of available commands.
Type 'help' followed by a command name for more
information on commands.

CPLEX> read H:\model.lp
Problem 'H:\model.lp' read.
Read time =    0.03 sec.
CPLEX> optimize
Tried aggregator 2 times.
MIP Presolve eliminated 1 rows and 4 columns.
MIP Presolve modified 4 coefficients.
Aggregator did 1 substitutions.
All rows and columns eliminated.
Presolve time =    0.02 sec.

Solution pool: 1 solution saved.

MIP - Integer optimal solution:  Objective = 1.2000000000e+001
Solution time =    0.03 sec.  Iterations = 0  Nodes = 0

CPLEX> write H:\model.mps
Problem written to file 'H:\model.mps'.
CPLEX>
```

What happens if you have 1000 variables and 500 constraints?

AMPL

What is AMPL?

- Preprocessor
- Processes generic model files
- Outputs MPS files that can be used by CPLEX



Transportation Model

Notation:

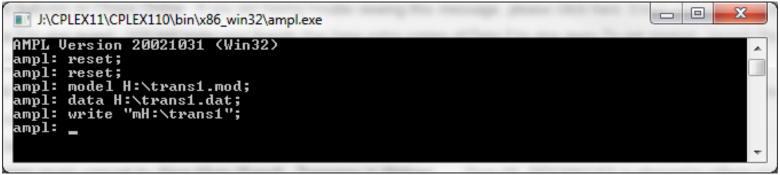
- I, J: sets of origins and destinations
- $s_i, i \in I$: supply at origin i
- $d_j, j \in J$: demand at destination j
- $c_{ij}, i \in I, j \in J$: cost of shipping a unit from origin i to destination j
- $x_{ij}, i \in I, j \in J$: number of units shipped from i to j

$$\begin{aligned}
\text{Min} \quad & \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} & \\
\text{s.t.} \quad & \sum_{j \in J} x_{ij} = s_i, i \in I & \text{No. of units shipped from } i \text{ is } s_i \\
& \sum_{i \in I} x_{ij} = d_j, j \in J & \text{No. of units shipped to } j \text{ is } d_j \\
& x_{ij} \geq 0, i \in I, j \in J &
\end{aligned}$$

Data from Anurag Verma's presentation used for this problem.

Obtaining the MPS file

If you want only the MPS file:



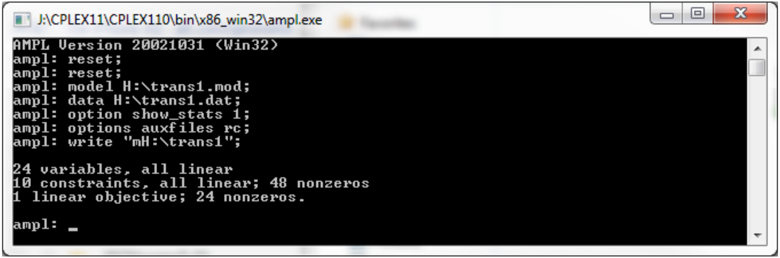
```
AMPL Version 20021031 (Win32)
ampl: reset;
ampl: reset;
ampl: model H:\trans1.mod;
ampl: data H:\trans1.dat;
ampl: write "mH:\trans1";
ampl: _
```

Notice the write command!

write “mH:\trans1” creates the file H:\trans1.mps.

Obtaining the MPS file

If you want something more...



```
AMPL Version 20021031 (Win32)
ampl: reset;
ampl: reset;
ampl: model H:\trans1.mod;
ampl: data H:\trans1.dat;
ampl: option show_stats 1;
ampl: options auxfiles rc;
ampl: write "mH:\trans1";

24 variables, all linear
10 constraints, all linear; 48 nonzeros
1 linear objective; 24 nonzeros.

ampl: _
```

- option show_stats 1; - displays problem statistics
- options auxfiles rc; - creates files trans1.row,trans1.col

The .scs file

An alternative:

- Create a file with extension .scs
- Add all the above commands to this file
- Open AMPL
- commands H:\trans1.scs; or
- include H:\trans1.scs;

Callable Library and Concert Technology

Callable Library:

- What? - set of routines used to interact with CPLEX from C code.
- Why?
  - Large-scale problems
  - Changing problem/data
  - Computational experiments
  - Using CPLEX in algorithms such as Column generation, Benders decomposition
- Routines written in C

Concert Technology

- Methods written in C++
- Object-oriented framework

References

References for you:

- www.google.com
- AMPL a modeling language for mathematical programming, *Fourer, Gay and Kernighan*

References I used:

- AMPL notes of Dr. Kianfar
- AMPL and CPLEX notes of Dr. Wilhelm
- AMPL a modeling language for mathematical programming, *Fourer, Gay and Kernighan*
- `http://www.decf.berkeley.edu/help/apps/ampl/cplex-doc/refcallablelibrary/index.html`