A Quick introdution to CPLEX

Jesper Larsen   Jens Clausen

February 8, 2005

CPLEX can be run in interactive mode or used as a library callable from e.g. C, C++ or Java programs. In this course we will only use CPLEX in interactive mode.

To start a CPLEX session type `cplex` and press the enter-key. You will then get an introductory text something like:

```
ILOG CPLEX 9.000, licensed to "university-lyngby", options: e m b q

Welcome to CPLEX Interactive Optimizer 9.0.0
  with Simplex, Mixed Integer & Barrier Optimizers
Copyright (c) ILOG 1997-2003
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>
```

To end the session type `quit` and press the enter-key. Please do not quit CPLEX by closing the xterm window before having exited CPLEX. At IMM we have 20 licenses, so at most 20 persons can run CPLEX independently of each other. If you quit by closing the xterm window before leaving CPLEX in a proper manner it will take the license manager some time to recover the CPLEX license.

CPLEX solves linear and integer programming problems. These must be entered either through the built-in editor of CPLEX or by entering the problem in ”your favorite editor”, saving it to a file, and then reading the problem into CPLEX by typing

`read <filename.lp>`.

The file has to have extension `.lp` and the contents has to be in the lp-format. A small example is shown below.

```
\Problem
name: soejle.lp

Minimize
 obj: x1 + x2 + x3 + x4
Subject To
 c1: x1 + 2 x3 + 4 x4 >= 6
 c2: x2 + x3 >= 3
End
```

The first line is a remark and it stretches the entire line. `Subject to` can be replaced with `st`. The text written in the start of each line containing the objective function or constraints is optional, so `obj:` can be omitted.

After having entered a problem, it can be solved by giving the command `optimize` at the `CPLEX>` prompt and press enter. To see the result, the command `display solution variables -` and press enter is used, ”-” indicating that the values of all variables are to be displayed.

CPLEX writes a log-file, which records the events of the session. An example of the log-file corresponding to the solution of the example above is shown below. The events of the session has been:

```
cplex <enter>
read soejle.lp <enter>
optimize <enter>
display solution variables - <enter>
quit <enter>
```

and the resulting log-file looks like:

```
Log started (V6.5.1) Tue Feb 15 10:24:58 2000


Problem 'soejle.lp' read.
```

```
Read time =    0.00 sec.
Tried aggregator 1 time.
LP Presolve eliminated 0 rows and 1 columns.
Reduced LP has 2 rows, 3 columns, and 4 nonzeros.
Presolve time =    0.00 sec.

Iteration log . . .
Iteration:     1    Infeasibility =             3.000000
Switched to devex.
Iteration:     3    Objective     =             3.000000

Primal - Optimal:  Objective =    3.0000000000e+00
Solution time =    0.00 sec.  Iterations = 3 (2)


Variable Name           Solution Value
x3                            3.000000
All other variables in the range 1-4 are zero.
```

Now let us take our initial problem and assume that we want `x1` and `x2` to be integer variables between 0 and 10. That the variables are positive are implicitely assumed by CPLEX, but we need to state the upper bound and the integrality condition. In this case our program will look like:

```
\Problem
name: soejle.lp

Minimize
 obj: x1 + x2 + x3 + x4
Subject To
 c1: x1 + 2 x3 + 4 x4 >= 6
 c2: x2 + x3 >= 3
Bounds
 x1<=10
 x2<=10
Integer
 x1
 x2
End
```

`Bounds` is used to declare bounds on variables, and the section afterwards, `Integer` states that `x1` and `x2` must be integer solutions. The bounds section must be placed before the section declaring the integer variables. It does not seem intuitive nevertheless if you do not state a bounds part **CPLEX will assume the integer variables to be binary**. If you want the integer variable to have no upper bound you can `x2<=INF` in the bounds section.

The command `help` shows the possible commands in the current situation. Also, CPLEX provides help if the current command is not sufficient to uniquely to determine an action. As an example, if one types `display` CPLEX will respond with listing the options and the question "Display what ?" CPLEX also offers possibilities to change parameters in a problem already entered - these possibilities may be investigated by entering `help` as the first command after having entered CPLEX.