ECE 3484 - Combinatorial Optimization - Using AMPL and CPLEX

The following is an introduction to using AMPL and CPLEX at Northeastern.

CPLEX is an industrial-strength integer and linear programming (ILP) solver. AMPL is a language that is used to express ILP models in an intuitive way. To solve a linear system, you first express the system using AMPL which is then translated into a form that CPLEX recognizes. CPLEX processes that input and reports the result back to AMPL. This entire processes is usually completed while you are within an AMPL command environment. This environment uses the CPLEX solver as a subroutine.

You should access AMPL and CPLEX from the COE network. To do this, telnet to gateway.coe.neu.edu. You will automatically be connected to a random workstation within the COE domain.

Edit the file .cshrc and add the following command:

`source /ECEnet/Apps1/ILOG/etc/ilog_setup.csh`. Then logout and log back in again.

Examples of ILP formulations are in `/ECEnet/Apps1/ILOG/ampl20010215/examples/models`. Copy the files `diet.mod` and `diet.dat` from this directory into your home directory, then run the package using the command `ampl`.

Within AMPL, you can solve an ILP formulation and display the results using the following commands. Note that each AMPL command ends with a semicolon.

```
model diet.mod;
data diet.dat;
solve;
display Buy;
```

The `model` and `data` commands cause AMPL to load the formulation. The `solve` commands causes AMPL to actually send the formulation to the CPLEX ILP solver. When the solver stops running, you can print the optimal values of the `Buy` variables using the `display` command. If there were syntax errors in your model description they would have been reported to you.

These commands could also have been placed in a single command file called run.file. The commands are then run from within AMPL using the command `include run.file;` You can also run these commands from the Unix prompt using the command `ampl run.file`.

There is a limit on the number of CPLEX jobs that can be running simultaneously anywhere on the COE network. If the limit has been reached, you will see an error message when you attempt to use the `solve` command. If this happens, wait a bit and tryy again. There is no limit on the number of AMPL jobs that can be simultaneously run.

The example formulations are a good place to find examples of how to implement common linear programming constructs. Files ending in .mod contain the main declarations and constraints of the LP model. Files ending in .dat contain the data that is unique to the specific problem to be solved. Every command in the .mod and .dat files can also be issued in the command mode. However it is more convenient to gather these commands together into command files and execute them using the model, data, and include commands.

The AMPL manual, on reserve in Snell library, contains detailed descriptions of how to use all of AMPL's features. Chapters 1-4 contain examples, chapters 5-9 give a thorough description of

AMPL, chapter 10 describes the AMPL command environment, and chapter 15 describes integer programming. The number of AMPL manuals is limited so please be considerate to your fellow students when using them. Try to share them as much as possible. There are also several online documents on the website that give a complete description of how to use AMPL and CPLEX.

Both AMPL and CPLEX have options that control how they function and display information. AMPL options are described in the AMPL manual and are set using the `option` command and displayed using the `options` command. CPLEX options are described in the CPLEX manual and are set by setting the AMPL option `cplex_options` to the desired CPLEX options.

By setting the shell environment variable `OPTIONS_IN` to the name of a file containing a list of AMPL option settings, you can easily initialize your AMPL environment. From Unix, use the command `setenv OPTIONS_IN amplinit`, and put the options commands in a file called `amplinit`.

To display output to a file, use the `display >filename` command. Commands that output to an existing file append to the end of it. This is a useful way to document the results of a computation.