AMPL aid for the Mexico project

1 Getting started

Once you have the model files, you are set to go. The available files are

`mex.mod` (the model file)
`mex.dat` (the data file)

We encourage you to look at the files to understand how the model is structured.
You may now start AMPL by giving the command
`ampl`
AMPL should start up and you get a prompt looking like
`ampl:`

2 AMPL

To load the model, write (all commands are terminated by ";")
`ampl: model mex.mod;`
To load the data, write
`ampl: data mex.dat;`
To obtain the optimal solution, write
`ampl: solve;`
You should get the result

```
MINOS 5.5: optimal solution found.
40 iterations, objective 21607.20587
```

The number is rather low, as the model is expressed in Mton and Mpesos in order to reduce the size of the constants.

You may now take a closer look at the solution. To see the value of a variable use the command `display`. As an example, to see the amount of raw-materials sent from the mines to the mills, write
`ampl: display raw_trans;`
The name of other variables may be found by studying the model. You should get the result

```
raw_trans [*,*]
:             Ahmsa    Fundidora     Hylsa     Hylsap     Sicartsa     :=
Cerro_Mer     2.24      0           0          0          0
Coahuila      5.24836   3.648       0          0          1.824
El_Encino     2.60484   5.84516     0          0          0
Laperla       3.47      0           0          0          0
Lastruchas    0         0           0          0          2.85
Penacol       0         0.220774    2.50418    1.65276    0.0263736
;
```

You may obtain the reduced cost for these variables by writing
```
ampl:  display raw_trans.rc;
```
In the same fashion, you may get the dual variables corresponding to the constraint `Raw_Cap` by writing
```
ampl:  display Raw_Cap.dual;
```
You may get the slack in the constraints by writing
```
ampl:  display Raw_Cap.slack;
```
If you want to aggregate values you may use summation in the displayed expressions. For example, to get the total amount of steel exported, you may write
```
ampl:  display sum{i in PRODUCER, c in EXPORTS, p in PRODUCT} prod_trans[i,c,p];
```
If you want to get specific elements you may index the variables and constraints, for example,
```
ampl:  display processing['Oven_Red','Ahmsa'];
```
returns the amount produced in the blast-furnace process in Ahmsa.

If you change the model and/or the data and wish to reload them, you must write either
```
ampl:  reset;
```
reseting everything, or
```
ampl:  reset data;
```
reseting everything from the data-file. If you do not do this, AMPL will complain as AMPL will believe that you are redefining variables and parameters.

Constants may be changed using the command `let`. As an example, the command
```
ampl:  let fixed_cost_raw:=40;
```
will increase the fixed cost of transporting raw-materials.

3 The most probable mistakes

Q: I wrote a command, but nothing happened. When I write the next command I get weird errors such as
```
syntax error
context:  >>>..............
```
A: You probably forgot a ";" after your last command. If nothing happens, look at the prompt. If it reads
```
ampl?
```
then AMPL is expecting the rest of the last command (perhaps you just forgot a ";" after the command).

Q: I get errors of the type
```
invalid subscript my_variable[j,i]
```
although it has indices i and j.
A: Check the order of your indices.