

**Operations Research: theory and  
applications to networking**  
*Lesson 2 – The MOSEL language  
and the Xpress<sup>MP</sup> solver*

**E. Leonardi**

**CCNE+ICT**

# Outline

**Overview of the MOSEL Language**

**The Xpress<sup>MP</sup> Solver**

# Definitions

## **Xpress<sup>MP</sup>:**

Large-scale optimisation software developed by FICO™(<http://www.fico.com/>)

## **Xpress-IVE:**

IVE stands for Interactive Visual Environment

Provides a friendly user interface for model coding, code debugging and result display. (Windows environment)

Student versions can be downloaded from

<http://optimization.fico.com/student-version-of-fico-xpress.html>

## **Mosel:**

a programming language that translates your mathematical programs/algorithms to a format readable by Xpress Optimiser.

File extension: \*.mos

# General MOSEL program

Model Name

Comment

```
model "NAME" !NAME your model.  
  
1  uses "mmxprs" !Use Xpress-Optimiser.  
  
2  declarations !Arrays and variables declaration.  
    !insert declarations here.  
end-declarations !End of declaration  
  
3  !Initialise input parameters.  
  
4  !Insert objective function here.  
  
5  !Insert constraints here.  
  
6  !Insert optimisation statement here, i.e. Maximise OR Minimise.  
  
7  !Insert output statements here.  
  
end-model !End of model
```

# The declaration part

Sets  
Declaration

Variables  
Declaration

Parameters  
Declaration

```
!sample declarations section
declarations
  Set1 = 1..10           !set1 is from 1 to 10

  Set2 = 1..5           !set2 is from 1 to 5

  Var1: mpvar
  !Var1 is a single variable

  Var2: array(Set1) of mpvar
  !Var2 is an array of 10 variables

  Var3: array(Set1,Set2) of mpvar
  !Var3 is a matrix of 10x5 variables

  InputParam1: real
  !InputParam1 is a single real paramter

  InputParam2: array(Set1) of integer
  !InputParam2 is an array of 10 integer paramters

  InputParam3: array(Set1,Set2) of real
  !InputParam3 is a matrix of 10x5 real paramters
end-declarations
```

# Parameters Initialization

```
! Initialise input parameters
```

```
InputParam1 := 10
```

```
! initialisation of a single parameter
```

```
InputParam2:: [10,20,30,40,50,60,70,80,90,100]
```

```
! array initialisation
```

```
InputParam3 (1,3) := 10.5
```

```
InputParam3 (5,9) := 20.5
```

```
! initialisation of elements
```

# Objective Function

ObjectiveName := *ObjectiveFunction*

- where “ObjectiveName” is the name of the objective function (no need to be declared)
- “*ObjectiveFunction*” is the function expressing the objective. It can be expressed in a format very similar to a mathematical function

Example:

```
TotalCost := sum(i in Set1) InputParam2 (i) * Var2 (i)
```

# Writing Constraints

$$\sum_{i=1}^N x_{ij} \leq 1, \forall j$$

`forall(j in SET1) sum(i in SET2) x(i,j) <= 1`

$\sum$  becomes `sum`

$\forall$  becomes `forall`



# The power of “forall”

If “forall” **does not** exist, you will have to type in every single constraint (*many* times) expressed in the specific version of the constraint, i.e.

```
sum(i in SET2) x(i,1) <= 1    !j=1
sum(i in SET2) x(i,2) <= 1    !j=2
sum(i in SET2) x(i,3) <= 1    !j=3
sum(i in SET2) x(i,4) <= 1    !j=4
sum(i in SET2) x(i,5) <= 1    !j=5
```

Fortunately, “forall” loop can simplify this task. We only need the following line of code

```
forall(j in SET1) sum(i in SET2) x(i,j) <= 1
```

## Writing Constraints - II

$$x_{ij} \leq 1, \forall i < j$$

```
forall(i in SET1, j in SET2 | i < j) x(i,j) <= 1
```

$$x_{ij} \leq 1, \forall i = j$$

```
forall(i in SET1, j in SET2 | i = j) x(i,j) <= 1
```

$$x_{ij} \leq 1, \forall i \neq j$$

```
forall(i in SET1, j in SET2 | i <> j) x(i,j) <= 1
```

## Writing Constraints - III

$$\sum_{i=1}^N x_{ij} \leq 1, \forall j$$

$$\sum_{k=1}^L z_{kj} \leq 1, \forall j$$

```
forall(j in SET1) do
    sum(i in SET2) x(i,j) <= 1
    sum(k in SET3) z(k,j) <= 1
end-do
```

This form of “forall” loop must be used when several statements are included within the loop.

# Initialization of Variables

By default variables are real numbers non-negative.

To initialize BINARY variables:

```
forall(i in SET1, j in SET2) x(i,j) is_binary
```

To initialize INTEGER variables:

```
forall(i in SET1, j in SET2) x(i,j) is_integer
```

# Optimization Statement

The easy task: just tell the program if the objective function has to be minimized or maximized.

In case of minimization:

`minimize(ObjectiveName)`

where ObjectiveName is the name of the objective function

Example:

```
minimize (TotalCost)
```

# Solution Output

To display the value of the objective function:

```
writeln("Objective value is ", getobjval, ".")
```

To display the value of a decision variable:

```
writeln("The value of x is ", getsol(x), ".")
```

To display arrays and matrices:

```
forall(i in PLANTS, j in CITIES ) do      || getsol(  
    writeln(i, " to ", j, ": ", getsol(x(i,j)))  
end-do
```

## Solution Output - II


More Complex Output with if-then-else cycle:

```
forall(j in BIN) do
  if getsol(y(j)) = 1 then
    writeln("Bin ", j, " is used. (Size: ", BINSIZE(j), ")")
    writeln(" Items: ")

    forall(i in ITEM)
      if getsol(x(i,j)) = 1 then
        writeln(" ", i, " (Size: ", ITEMSIZE(i), ")")
      end-if
    end-if
  else
    writeln("Bin ", j, " is NOT used. (Size: ", BINSIZE(j), ")")
    writeln(" Items: NONE.")
  end-if
end-do
```

Xpress-IVE Student License - Commercial Use Prohibited - [max\_flow.mos]

File Project Edit View Build Debug Deploy Modules Wizards Window Optimizer Help

Search: File Position: 

Model Explorer

Most recent entities

Solution 1 / 1

Entities A--> Z

- (C:\Users\lpolin\Documents\My Drop
- Parameters
- Constants
- Primitives
- Subroutines
- User-defined Types
- Problems
  - Main Problem
    - Decision Variables
      - arrays:
        - flow**
      - Constraints
        - scalars:
          - NoReturn
        - arrays:
          - Balance
          - Capacity

**Model Bar**

max\_flow.mos

```
*****  
model "Network reliability"  
  uses "mmxprs", "mmive"  
  
declarations  
  NODES: range                                ! Set of nodes  
  SOURCE = 1; SINK = 11                       ! Source and Sink  
  
  ARC: array(NODES,NODES) of integer          ! 1 if arc exists  
  
  flow: array(NODES,NODES) of mpvar           ! 1 if flow exists  
end-declarations  
  
initializations from 'maxflow.dat'  
  ARC  
end-initializations  
  
forall(n,m in NODES | exists(ARC(n,m)) and n<m)  
  forall(n,m in NODES | exists(ARC(n,m)) ) create  
  
! Objective: number of disjunctive paths  
Paths:= sum(n in NODES) flow(SOURCE,n)  
  
! Flow conservation  
forall(n in  
  Balance(n):= sum(m in NODES | flow(n,m)) - sum(m in  
  Capacity(n):= sum(m in NODES | flow(n,m)) <= 1  
do
```

**Coding Window**

Output/Input

Clear

Total number of paths: 10

- 1 - 2 - 11
- 1 - 3 - 11
- 1 - 4 - 11
- 1 - 5 - 11
- 1 - 6 - 11
- 1 - 7 - 11
- 1 - 8 - 11
- 1 - 9 - 11
- 1 - 10 - 11

**Run Bar**

Type here:

Output/Input Stats Matrix

BB tree User graph IIS

Solutions Objective MIP search

Information

C:\Users\lpolin\Documents\My Dropbox\Operational Research TLM 2011 (1)\Lab\examples\max\_flow.mos compiled successfully.  
Model version: 3.2.3  
Module(s) in use: mmxprs version 2.2.3, mmive version 1.22.4.  
Started running C:\Users\lpolin\Documents\My Dropbox\Operational Research TLM 2011 (1)\Lab\examples\max\_flow.mos  
Xpress-IVE: Model run complete

**Information Bar**

Build Search Debug Watch

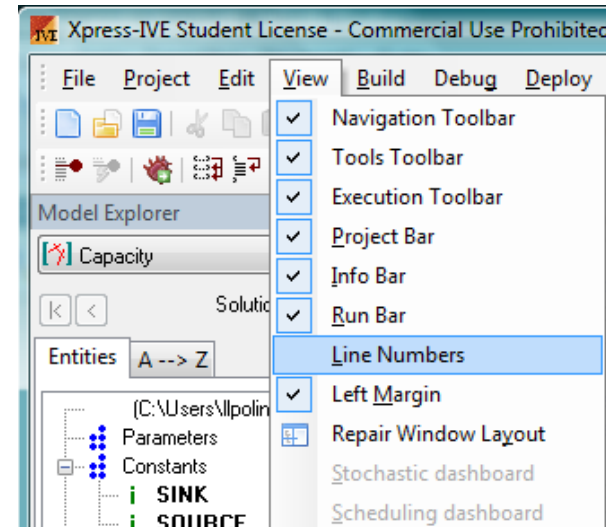
Copy to clipboard

Ready: Idle Free Memory: 755 MB Running in Student Mode Line: 39/107 Col: 34 OVR

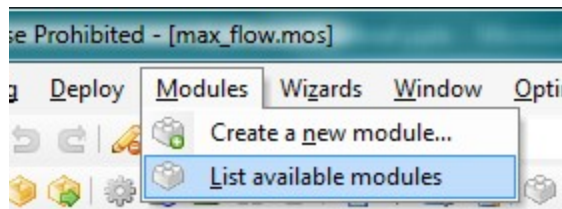


# Initial Setup

## 1- Enable line numbers



## 2- Check if the Optimizer Module is installed

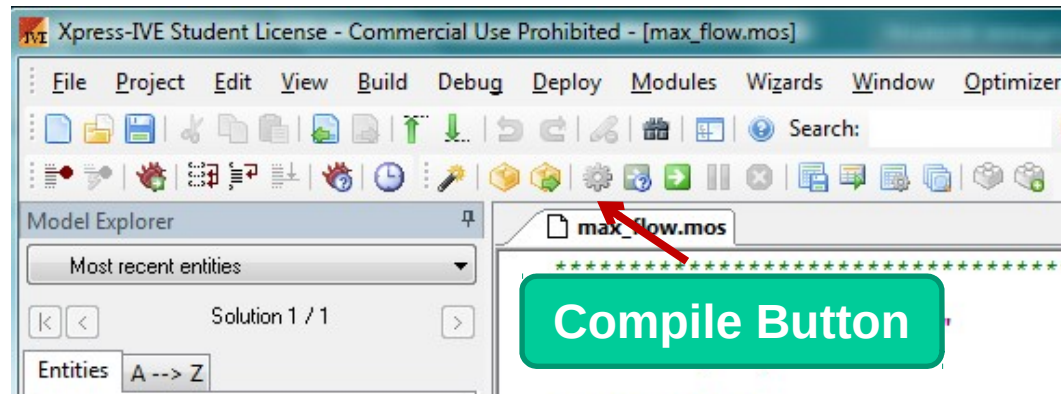


The screenshot shows the 'Modules available to Xpress-IVE' dialog box. It contains a table with two columns: 'Name' and 'Version'. The table lists various modules and their versions. The 'mmxprs' module is highlighted in blue.

Name	Version
mmetc	1.8.2
mmive	1.22.4
mmive	(IVE)1.22.4
mmjobs	2.0.3
mmnl	1.2.3
mmoci	invalid
mmodbc	2.2.2
mmquad	1.2.6
mmsystem	1.8.8
mmxad	1.2.6
mmxprs	2.2.3

# Compile a Mosel file

## 1- Click on the Compile Button



## 2- Check the Information Bar



## Model Explorer

Capacity (n/a)

Entities A --&gt; Z

## max\_flow.mos

```

ARC
end-initializations

forall(n,m in NODES | exists(ARC(n,m)) and n<m )
forall(n,m in NODES | exists(ARC(n,m)) ) create

! Objective: number of disjunctive paths
Paths:= sum(n in NODES) flow(SOURCE,n)

! Flow conservation and capacities
forall(n in NODES | n<>SOURCE and n<>SINK) do
  Balance(n):= sum(m in NODES) flow(m,n) = sum(m in
  Capacity(n):= sum(m in NODES) flow(n,m) <= 1
end-do;

! No return to SOURCE node
NoReturn:= sum(n in NODES) flow(n, SOURCE) = 0

forall(n,m in NODES | exists(ARC(n,m)) ) flow(n,

! Solve the problem
maximize(Paths)

! Solution printing
writeln("Total number of paths: ", getobjval)

forall(n in NODES | n<>SOURCE and n<>SINK)
  if(getsol(flow(SOURCE,n))>0) then

```

## Output/Input

Clear

Type here:

Output/Input

Stats

Matrix

BB tree

User graph

IIS

Solutions

Objective

MIP search

## Information

\\Documents\\My Dropbox\\Operational Research TLM 2011 (1)\\Lab\\examples\\max\_flow.mos(Line: 53, Col: 8): error 100: syntax error before ';

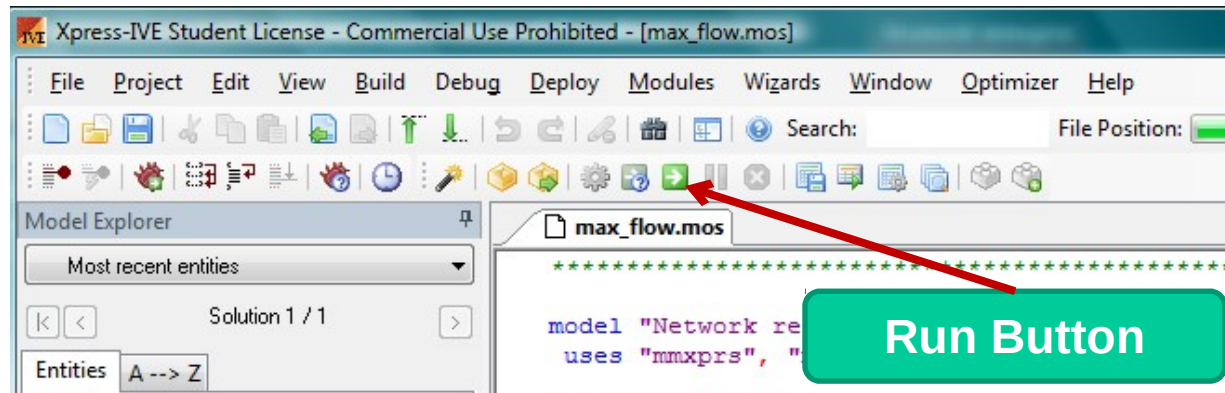
No matrix is available

# Debug

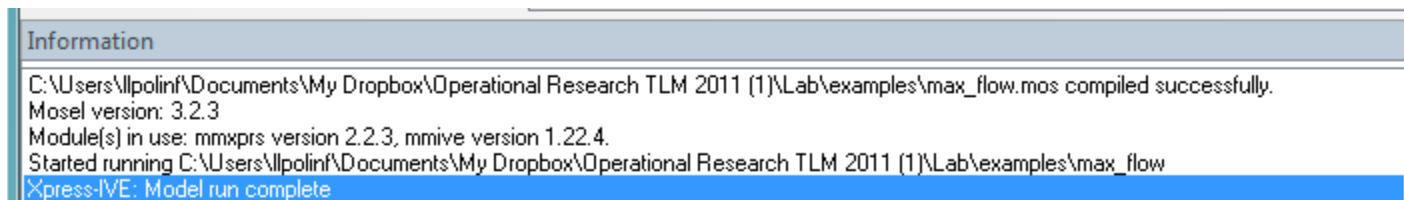
- 1- Always start from the first error on the top**
- 2 - Remember that MOSEL is case-sensitive**  
**Typical Error “Forall” instead of “forall”**

# Problem Execution

## 1- Click on the Run button



## 2- Information Bar reports problem status



## 3- Run Bar reports the outputs

## 4- Additional information (dual/slack) reported in the Model Bar

# Statistics

## “Stats” label of the Run Bar

Stats			
<b>Matrix:</b>		<b>Presolved:</b>	
Rows(constraints):	19	Rows(constraints):	18
Columns(variables):	120	Columns(variables):	90
Nonzero elements:	290	Nonzero elements:	243
Global entities:	120	Global entities:	90
Sets:	0	Sets:	0
Set members:	0	Set members:	0
Overall status: <b>Finished global search.</b>			
<b>LP relaxation:</b>		<b>Global search:</b>	
<b>Algorithm:</b>	<b>Simplex primal</b>	Current node:	1
Simplex iterations:	15	Depth:	1
Objective:	10	Active nodes:	0
Status:	Unfinished	Best bound:	10
Time:	0.0s	Best solution:	10
		Gap:	0%
		Status:	Solution is optimal.
		Time:	0.0s
<b>Time overheads:</b>			
Progress graphs:	0.0s		
Writing output:	0.0s		
Pausing:	0.0s		
Updating status:	0.0s		
Output/Input	Stats	Matrix	Solutions
User graph	IIS	Objective	MIP search
		BB tree	



Questions?