Introduction to AMPL

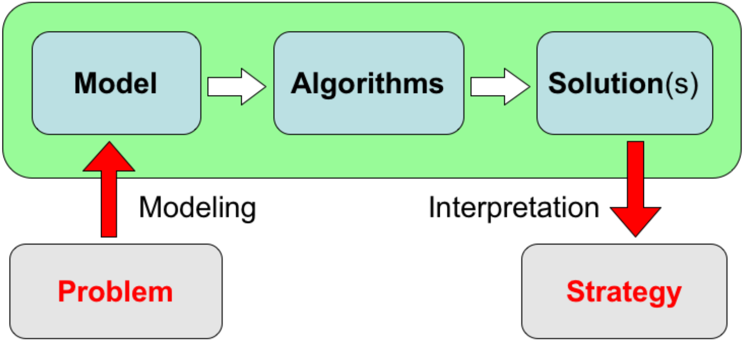
ORLAB - Operations Research Laboratory

Stefano Gualandi

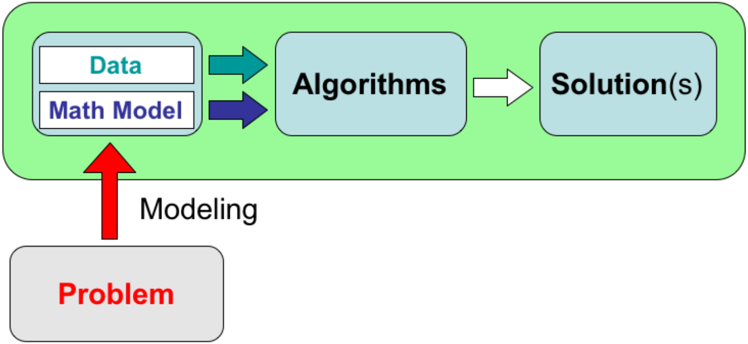
Politecnico di Milano, Italy

October 27, 2009

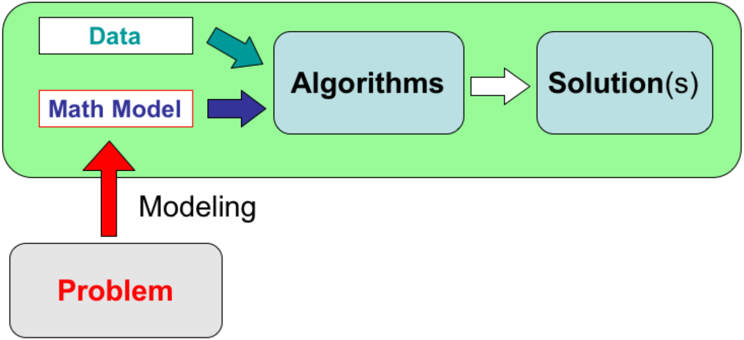
From Modeling to Strategies



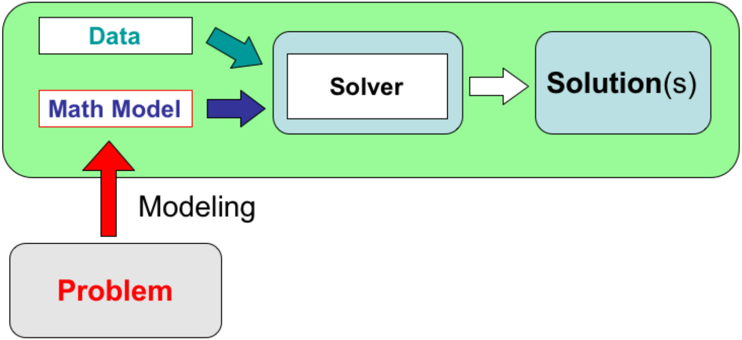
From Modeling to Strategies



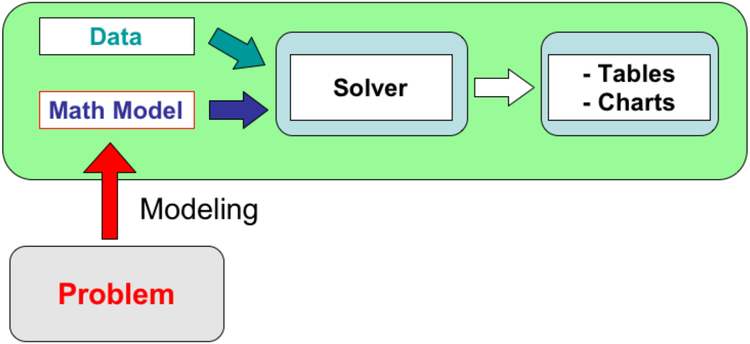
From Modeling to Strategies



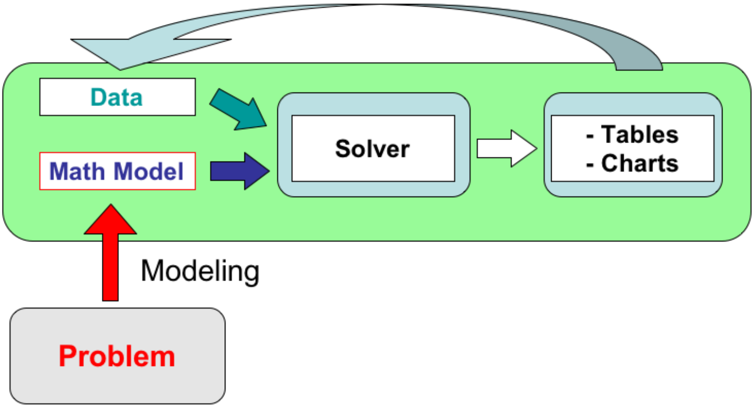
From Modeling to Strategies



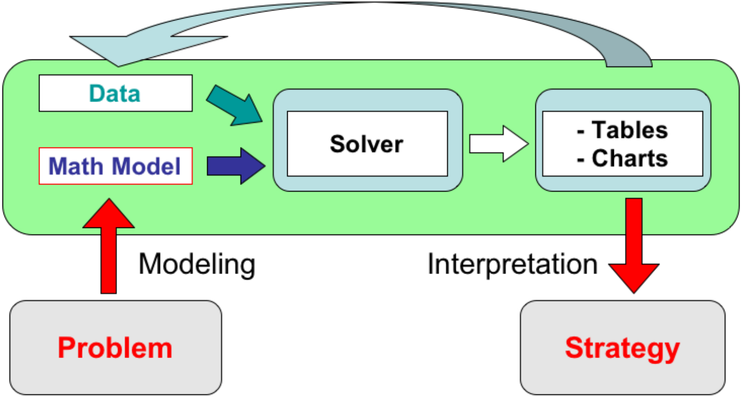
From Modeling to Strategies



From Modeling to Strategies



From Modeling to Strategies



Why lab sessions are useful?

Lab sessions close the gap between theory and practics. The goals are:

- To train *intuition* and *skills* necessary to formulate models
- To learn *algebraic modeling languages*
- To understand functionalities, strength, and weakness of *solvers*
- To acquire skills in *interpreting solutions*

Warning:

Modeling is more an art than a science

Model formulation

- Although modeling is a kind of *art*, good practices exist:

  Do not reinvent the wheel

- Both academia and industry have developed models for a wide classes of problems:
  - Set covering
  - Set partitioning
  - Matching
  - Routing
  - Scheduling
  - ...
- Different application domain problems do have the same mathematical model

  Hint: study by heart all the examples of the course notes!

Warning: *Lab sessions complement exercise sessions*

Algebraic modeling languages

- Algebraic modeling languages have a well-defined syntax and semantic close to mathematics
- They are declarative, that is, they say what to solve (the model), but not how (which algorithm or solver to use)
- They force users to distinguish between the logic structure of the problem (i.e., the model) and the numerical data (i.e., the instances of the problem)
- During the lab sessions we use AMPL.

WARNING: read carefully the error messages!

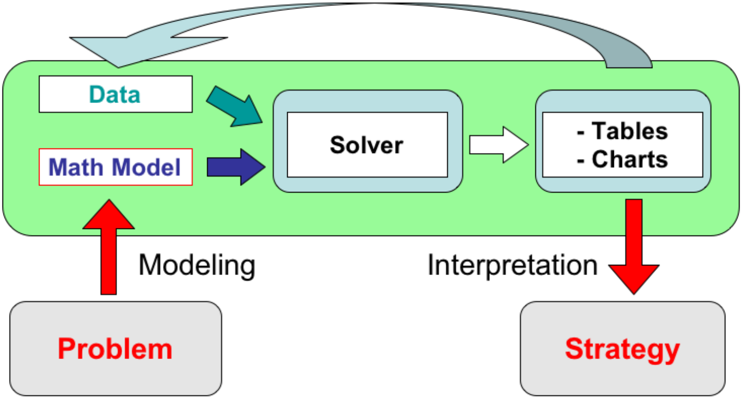
Solvers

- A *solver* is a software that takes as input a problem description (i.e., model+data) and output a solution (if one exists)
- Depending on the problem structure different solvers should be used:
  - Linear Programming solver
  - Integer Programming solver
  - Semi-definite Programming solver
  - Non-Linear solver
- Solvers have many options which can improve the resolution time if properly used
- During the lab sessions you will solve only Linear and Integer programs using the CPLEX solver.

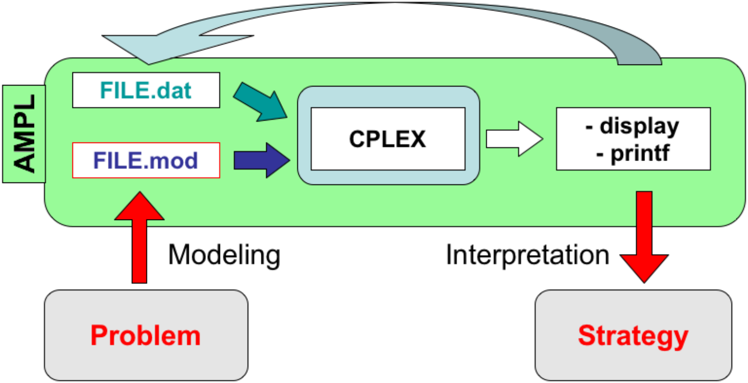
Interpreting solution

- Solvers give correct solutions with respect to your model... but is the given solution valid for the given problem?
- If they are not valid, 99.999% of the times you wrote the wrong model or the wrong data (%50 of the times you commit syntax errors!)
- Common pitfalls:
  - unbounded variables
  - missing constraints
  - missunderstood constraints
  - wrong quantification in the summation of constraints
  - wrong objective function
- You should know what solutions look like
- Getting a solution is not the end of the story!

What you do during lab sessions



What you do during lab sessions



A short introduction to AMPL

- In AMPL the model and data are written in two separate text (ASCII) files:
  - *FILENAME*.**mod**: it contains the model of the problem, i.e., variables, constraints, and objective function;
  - *FILENAME*.**dat**: it contains the data specifing an instance of the problem, i.e., the coefficients of the constraints and objective function
- Useful AMPL keywords (syntax sugar):
  - `set`: defines set of elements;
  - `param`: defines constant parameters like constants, vectors, or matrices;
  - `var`: defines continuous variables. They can be defined also as `integer` and `binary`;
  - `sum`: defines a linear sum;
  - `subject to`: defines a family of constraints;
  - `minimize` or `maximize`: defines the objective function;

An example: math model

Math model:

$$
\begin{aligned}
\min \quad & \sum_{i \in I} c_i x_i \\
\text{s.t.} \quad & Ax \leq b \\
& x \geq 0 \quad x \in R^n
\end{aligned}
$$

where

- A matrix $m \times n$
- $I = \{1, \ldots, n\}$ set indexes of columns of A
- $J = \{1, \ldots, m\}$ set indexes of rows of A

An example: FILE.mod

```
# YOU CAN WRITE COMMENTS
set I;
set J;

param A{J,I};
param b{J};
param c{I};

var x{I}, >= 0, integer;

minimize Cost:
    sum {i in I} c[i]*x[i];

subject to MatrixConstraint {j in J}:
    sum {i in I} A[j,i]*x[i] <= b[j];
```

An example: FILE.dat

```
set I = a b c;
set J = 1 2;

param A :
      a    b    c    :=
 1   -1    2   3.1
 2   3.3  -7   -1    ;

param b :=
    1     -3
    2     9.2;

param c :=
    a     2
    b     4
    c     3;
```

Using AMPL

1. select the solver you want to use (CPLEX):
   `% option solver cplex;`
2. write/load the model (.mod) and data (.dat) files
3. solve the problem
4. if you get (syntax) errors, check your files and try again
5. if you get a solution, look it:
   `% display x;`
6. in any case, inspect how AMPL interpret your .mod file:
   `% expand Cost;`
   `% expand MatrixConstraints;`

Exercises:

- If the variable x were continuous, the optimum would be greater or smaller? Why?
- And what if the x variable were `binary`? The model/data are still correct? Why?

Mobile phones: problem statement

We have an assemblage kit containing pieces of different colors and dimensions, each corresponding to a specific component of a mobile phone. We have: 10 display modules; 18 memory modules; 12 transmission modules; 6 keyboards; 9 mouse; 10 microcameras. Appropriately combining these elements, we can assemble mobile phones of two different models. Each phone of type A brings 3 points; each phone of type B brings 8 points.
The game consists in assembling the pieces that we have, trying to make the highest possible score.

Mobile phones: modeling

- Sets:
  - M = set of types of mobile phones
  - C = set of components
- Parameters:
  - p_i = score for model i
  - $a_{i,j}$ = number of components j necessary for a type i
  - d_j = availability of component j
- Variables:
  - x_i = number of produced mobile phone of type i

Mobile phones: modeling

Objective function:

$$\max \sum_{i \in M} p_i x_i$$

Constraints:

$$\sum_{i \in M} a_{ij} x_i \leq d_j \qquad \forall j \in C$$

$$x_i \geq 0 \quad \text{integer} \qquad \forall i \in M$$

Mobile phones: file ex-2.mod

```
set Tyeps;
set Components;

param Score{Types};
param Available{Components};
param Necessary{Types, Components};

var x{Types} >=0, integer;

maximize Profit:
    sum {i in Types} Score[i]*x[i];

subject to ComponentConstraint {j in Components}:
    sum{i in Types} Necessary[i,j]*x[i] <= Available[j];
```

Mobile phones: file ex-2.dat

```
set Types:= a b;

set Components :=
     display memory trasmission
     keyboards mouse microcamera;

param Score :=
     a       3
     b       8;

param Available:=
 display        10
 memory         18
 trasmission    12
 keyboards       6
 mouse           9
 microcamera    10;

param Necessary:
     display   memory   trasmission   keyboards   mouse   microcamera:=
 a      1        2          1             2         1          1
 b      2        2          3             3         0          0;
```

Mobile phones: questions

1. load the model and data file
2. solve the problem: which is the optimum?
3. which and how many models are produced?
4. what if you had only 0 *mouse* component? Try to modify the .dat file and explain what happen.
5. what if you had 8 *keyboards* component? Try to modify the .dat file and explain what happen.
6. what if you had 9 *keyboards* component? Try to modify the .dat file and explain what happen.

Mobile phones: questions

1. load the model and data file
2. solve the problem: which is the optimum?
3. which and how many models are produced?
4. what if you had only 0 *mouse* component? Try to modify the .dat file and explain what happen.
5. what if you had 8 *keyboards* component? Try to modify the .dat file and explain what happen.
6. what if you had 9 *keyboards* component? Try to modify the .dat file and explain what happen.
7. can theory help you to decide which components you should have or buy?

Mobile phones: questions

1. load the model and data file
2. solve the problem: which is the optimum?
3. which and how many models are produced?
4. what if you had only 0 *mouse* component? Try to modify the .dat file and explain what happen.
5. what if you had 8 *keyboards* component? Try to modify the .dat file and explain what happen.
6. what if you had 9 *keyboards* component? Try to modify the .dat file and explain what happen.
7. can theory help you to decide which components you should have or buy?
8. type: `% display ComponentConstraint.slack`

A boatload problem

A private company working on packing materials has to organize a boatload for a client. Their boat has a weight capacity of 8 tons and a volume capacity of 150 m^3. It has on stock the following materials: polyester, straw, waved cardboard, plastic for packing and normal cardboard.
The following table shows the quantity required for each material by the client, the weight of this quantity, and the corresponding gain.

A boatload problem

material	$[m^3]$	[kg]	[euro]
polyester	45	1800	80
straw	60	1200	50
waved cardboard	30	4000	90
plastic for packing	50	1500	60
normal cardboard	55	5500	100

The client buys the material only if he gets the required amount of each material.
Write the model maximizing the gain.

Boatload: modeling

- Sets:
  - M = set of materials
- Data:
  - p_i = gain of materials i
  - w_i = weight of materials i
  - s_i = volume of materials i
  - S = volume capacity of the boat
  - W = weight capacity of the boat
- Variables:
  - $x_i = 1$ if the materials i is included in the boatload, 0 otherwise

Boatload: model

$$
\begin{aligned}
\max \quad & \sum_{i \in M} p_i x_i \\
& \sum_{i \in M} w_i x_i \leq W \\
& \sum_{i \in M} s_i x_i \leq S \\
& x_i \in \{0, 1\} \quad \forall i \in M
\end{aligned}
$$

Question: how many constraints this model have?

Homework

A private company is planning the production of three devices (A_1,A_2,A_3). The monthly working days are 22. The following table shows: the maximal demand (quintals), the selling price (dollars/quintal), the production price per produced quintal, and the producing quota (i.e., the maximum number of produced quintals per day, if all the resources are used to produce a single device).

Device	A_1	A_2	A_3
Maximum demand	5300	4500	5400
Selling price	\$124	\$109	\$115
Producing price	\$73.30	\$52.90	\$65.40
Producing quota	500	450	550

Homework

1. 1.1 Express the model using AMPL and find the production plan which maximizes the total gain.
   1.2 Study how the mathematical model, the model and data in AMPL, and the corresponding solution change if a set-up fix cost is introduced for every device:

Product	A_1	A_2	A_3
Set-up cost	\$170000	\$150000	\$100000

Homework

1. Study how the mathematical model, the model and data in AMPL, and the corresponding solution changes if beside the set-up cost, there is a constraint which states that if a device is produced, then it must be produced in at least a certain amount:

Device	A_1	A_2	A_3
Minimal amount	20	20	16

Links and email

- Lab sessions web site:
  http://home.dei.polimi.it/gualandi/FRO-D
- E-mail:
  gualandi_at_elet.polimi.it
- Italian guide to AMPL:
  http://.../gualandi/FRO-D/guida_ampl.pdf
- Set of examples of translating math model into AMPL
  http://.../gualandi/FRO-D/ampl_examples.pdf