# Introduction into Optimization
# GLPK model of examples

Máté Hegyháti [1]

Department of Computer Science and Systems Technology
Faculty of Information Technology
University of Pannonia

Last modified: November 30, 2012

[1] hegyhati@dcs.uni-pannon.hu
http://dcs.uni-pannon.hu/hegyhati/oktatas

# Contents

# Optimization softwares

The previously mentioned (and other) optimization algorithms has been implemented in many general purpose optimization softwares.

## Optimization softwares

The previously mentioned (and other) optimization algorithms has been implemented in many general purpose optimization softwares.

- **COIN-OR** open source linear optimization project
- **OpenOpt** free numerical optimization framework with automatic differentiation features
- **GLPK** open source MILP/LP solver
- **CLPEX** Proprietary linear and quadratic optimization software
- **Gurobi** Proprietary linear and quadratic optimization software
- ...

## Optimization softwares

The previously mentioned (and other) optimization algorithms has been implemented in many general purpose optimization softwares.

- **COIN-OR** open source linear optimization project
- **OpenOpt** free numerical optimization framework with automatic differentiation features
- **GLPK** open source MILP/LP solver
- **CLPEX** Proprietary linear and quadratic optimization software
- **Gurobi** Proprietary linear and quadratic optimization software
- ...

These softwares differ in

- input format
- capabilities
- algorithms
- implementation

# GUSEK,GLPK, and GMPL

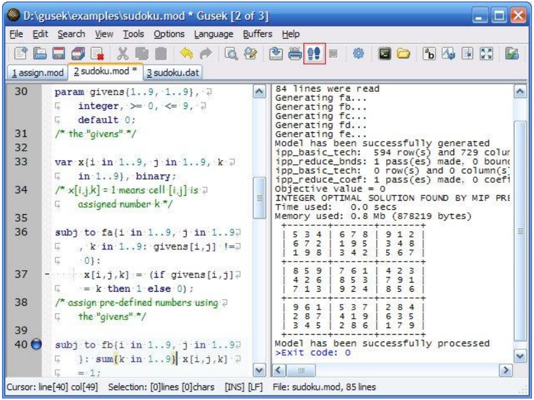
In this course we use the **GUSEK** software

# GUSEK,GLPK, and GMPL

In this course we use the **GUSEK** software, that is a graphical interface for the **GLPK** (GNU Linar Programming Kit) solver

# GUSEK,GLPK, and GMPL

In this course we use the **GUSEK** software, that is a graphical interface for the **GLPK** (GNU Linar Programming Kit) solver, that can solve problems described in **GMPL** (GNU MathProg Language).




Download GUSEK: http://sourceforge.net/projects/gusek/

# Files, commands

filename.mod the problem description in GMPL format

filename.out the solution in GLPK format

filename.d data file for the model (optional)

# Files, commands

filename.mod the problem description in GMPL format

filename.out the solution in GLPK format

filename.d data file for the model (optional)

Check **Tools** $\rightarrow$ **Generate Output File on Go**.

# Files, commands

filename.mod the problem description in GMPL format

filename.out the solution in GLPK format

filename.d data file for the model (optional)

Check **Tools** → **Generate Output File on Go**.

Test the model with **Ctrl** + **F7**, solve by **F5**.

# Files, commands

filename.mod the problem description in GMPL format

filename.out the solution in GLPK format

filename.d data file for the model (optional)

Check **Tools** $\rightarrow$ **Generate Output File on Go**.

Test the model with **Ctrl** + **F7**, solve by **F5**.

If the model is

bad the output of GLPK on the right pane gives some hints about the error

good the solution file is opened in a new tab

# Declaring variables

The general format is to declare a variable:

**var** Vname *Options* **;**

# Declaring variables

The general format is to declare a variable:

**var** Vname *Options* **;**

where

**var** keyword for variable declaration (mandatory)

Vname name of the variable (mandatory, and unique)

*Options* some examples (optional, can be combined):

- $>= 0$ for nonnegative variables
- binary for binary variables
- integer for integer variables

## Declaring variables

The general format is to declare a variable:

**var** Vname *Options* **;**

where

- **var** keyword for variable declaration (mandatory)
- Vname name of the variable (mandatory, and unique)
- *Options* some examples (optional, can be combined):
  - $>= 0$ for nonnegative variables
  - binary for binary variables
  - integer for integer variables

Don't forget the semicolon at the end.

# Constraints

The general format for a constraint:

**s.t.** Cname **:** Constraint **;**

# Constraints

The general format for a constraint:

**s.t.** Cname **:** Constraint **;**

where

**s.t.** "subject to", keyword for constraints (mandatory)

Cname name of the constraint (mandatory, and unique)

Contraint constraint expression, use

$>=, =, <=$ for $\geq, =, \leq$

$*$ for multiplication

# Constraints

The general format for a constraint:

**s.t.** Cname **:** Constraint **;**

where

**s.t.** "subject to", keyword for constraints (mandatory)

Cname name of the constraint (mandatory, and unique)

Contraint constraint expression, use

$>=, =, <=$ for $\geq, =, \leq$

$*$ for multiplication

Pay attention to the use of colon and semicolon, it is a frequent source of syntactic errors.

# Objective function

The general format for the objective function:

**Goal** Oname : Objective ;

# Objective function

The general format for the objective function:

**Goal** Oname : Objective ;

where

**Goal** mandatory, either

- **maximize** or
- **maximize**

Oname name of the objective (mandatory, and unique)

Objective expression of the objective.

# Objective function

The general format for the objective function:

**Goal** Oname **:** Objective **;**

where

**Goal** mandatory, either

- **maximize** or
- **maximize**

Oname name of the objective (mandatory, and unique)

Objective expression of the objective.

Pay attention to the use of colon and semicolon, it is a frequent source of syntactic errors.

# Names, Comments

## Names

Should not contain spaces, dots, special characters.

# Names, Comments

## Names

Should not contain spaces, dots, special characters.

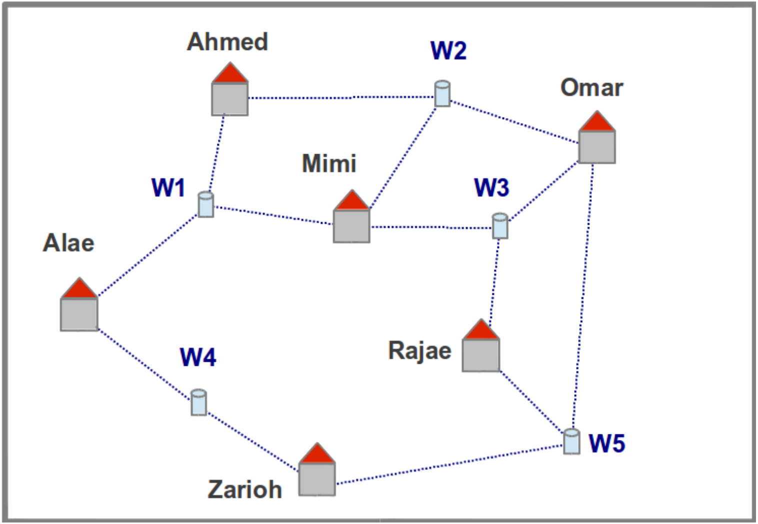
## Comments

Comments can be left in the source the facilitate understanding

Single line comments start with #

Multiple line comments start with /* and end with */

# Reminder: the problem



# Reminder: the model

$$
\begin{array}{ccccccccccc}
y_1 & + & y_2 & & & & & & & \geq & 1 \\
y_1 & & & & & + & y_4 & & & \geq & 1 \\
y_1 & + & y_2 & + & y_3 & & & & & \geq & 1 \\
 & & y_2 & + & y_3 & & & + & y_5 & \geq & 1 \\
 & & & & y_3 & & & + & y_5 & \geq & 1 \\
 & & & & & & y_4 & + & y_5 & \geq & 1
\end{array}
$$

$$y_1, y_2, y_3, y_4, y_5 \in \{0, 1\}$$

$$y_1 + y_2 + y_3 + y_4 + y_5 \rightarrow \min$$

# Variable declrations

$$y_1, y_2, y_3, y_4, y_5 \in \{0, 1\}$$

```
var y1, binary;
var y2, binary;
var y3, binary;
var y4, binary;
var y5, binary;
```

## Constrainsts

$$
\begin{array}{ccccccccccc}
y_1 & + & y_2 & & & & & & & \geq & 1 \\
y_1 & & & & & + & y_4 & & & \geq & 1 \\
y_1 & + & y_2 & + & y_3 & & & & & \geq & 1 \\
 & & y_2 & + & y_3 & & & + & y_5 & \geq & 1 \\
 & & & & y_3 & & & + & y_5 & \geq & 1 \\
 & & & & & & y_4 & + & y_5 & \geq & 1
\end{array}
$$

```
s.t. Ahmed:  y1 + y2 >= 1;
s.t. Alae:   y1 + y4 >= 1;
s.t. Mimi:   y1 + y2 + y3 >= 1;
s.t. Omar:   y2 + y3 + y5 >= 1;
s.t. Rajae:  y3 + y5 >= 1;
s.t. Zarioh: y4 + y5 >= 1;
```

# Objective

$$y_1 + y_2 + y_3 + y_4 + y_5 \rightarrow \min$$

```
minimize number_of_wells_to_intall:
  y1 + y2 + y3 + y4 + y5;
```

# Overall modell

```
var y1, binary;
var y2, binary;
var y3, binary;
var y4, binary;
var y5, binary;

s.t. Ahmed:  y1 + y2 >= 1;
s.t. Alae:   y1 + y4 >= 1;
s.t. Mimi:   y1 + y2 + y3 >= 1;
s.t. Omar:   y2 + y3 + y5 >= 1;
s.t. Rajae:  y3 + y5 >= 1;
s.t. Zarioh: y4 + y5 >= 1;

minimize number_of_wells_to_intall:
  y1 + y2 + y3 + y4 + y5;
```

# Output file

```
Problem:    model_simple
Rows:       7
Columns:    5 (5 integer, 5 binary)
Non-zeros:  19
Status:     INTEGER OPTIMAL
Objective:  number_of_wells_to_intall = 2 (MINimum)

   No.   Row name        Activity     Lower bound   Upper bound
------ ------------    ------------- ------------- -------------
     1 Ahmed                       1             1
     2 Alae                        1             1
     3 Mimi                        1             1
     4 Omar                        1             1
     5 Rajae                       1             1
     6 Zarioh                      1             1
     7 number_of_wells_to_intall
                                   2

   No. Column name       Activity     Lower bound   Upper bound
------ ------------    ------------- ------------- -------------
     1 y1           *              1             0             1
     2 y2           *              0             0             1
     3 y3           *              0             0             1
     4 y4           *              0             0             1
     5 y5           *              1             0             1

Integer feasibility conditions:
```

# Output file - Important parts

```
Status:       INTEGER OPTIMAL
Objective:    number_of_wells_to_intall = 2 (MINimum)
```

The optimal solution is found and its value is 2.

# Output file - Important parts

```
Status:      INTEGER OPTIMAL
Objective:   number_of_wells_to_intall = 2 (MINimum)
```

The optimal solution is found and its value is 2.

```
   No. Column name        Activity      Lower bound    Upper bound
------ ------------    ------------- ------------- -------------
     1 y1           *              1             0             1
     2 y2           *              0             0             1
     3 y3           *              0             0             1
     4 y4           *              0             0             1
     5 y5           *              1             0             1
```

The values of the variables in the optimal solution (Activity column).

# Reminder: the problem

| | Shampoo | Cream | Conditioner | Lotion | |
|---|---|---|---|---|---|
| Price (dh/l) | 30 | 25 | 35 | 40 | |
| Ingredients (kg/l) | | | | | Stock(kg) |
| Herb 1 | 0.1 | 0.01 | | 0.03 | 7 |
| Herb 2 | 0.1 | 0.05 | 0.1 | 0.01 | 10 |
| Herb 3 | | 0.2 | 0.1 | 0.02 | 13 |
| Herb 4 | | | 0.1 | 0.05 | 6 |
| Footprints (kg/l) | | | | | Limit(kg) |
| Water | 5 | 3 | 3 | 2 | 400 |
| $CO_2$ | 2.0 | 3.5 | 2.2 | 4.2 | 350 |

# Reminder: the model

$$\begin{array}{rcrcrcrcl}
0.1 \cdot x_{Sh} & + & 0.01 \cdot x_{Cr} & + & 0 \cdot x_{Co} & + & 0.03 \cdot x_{Lo} & \leq & 7 \\
0.1 \cdot x_{Sh} & + & 0.05 \cdot x_{Cr} & + & 0.1 \cdot x_{Co} & + & 0.01 \cdot x_{Lo} & \leq & 10 \\
0 \cdot x_{Sh} & + & 0.2 \cdot x_{Cr} & + & 0.1 \cdot x_{Co} & + & 0.02 \cdot x_{Lo} & \leq & 13 \\
0 \cdot x_{Sh} & + & 0 \cdot x_{Cr} & + & 0.1 \cdot x_{Co} & + & 0.05 \cdot x_{Lo} & \leq & 6 \\
5 \cdot x_{Sh} & + & 5 \cdot x_{Cr} & + & 5 \cdot x_{Co} & + & 2 \cdot x_{Lo} & \leq & 400 \\
2.0 \cdot x_{Sh} & + & 3.5 \cdot x_{Cr} & + & 2.2 \cdot x_{Co} & + & 4.2 \cdot x_{Lo} & \leq & 350
\end{array}$$

$$x_{Sh}, x_{Cr}, x_{Co}, x_{Lo} \in [0, \infty[$$

$$30 \cdot x_{Sh} + 25 \cdot x_{Cr} + 35 \cdot x_{Co} + 40 \cdot x_{Lo} \to \max$$

# Variable declrations

$$x_{Sh}, x_{Cr}, x_{Co}, x_{Lo} \in [0, \infty[$$

```
var xSh >=0;
var xCr >=0;
var xCo >=0;
var xLo >=0;
```

# Constrainsts

$$
\begin{array}{rcrcrcrcr}
0.1 \cdot x_{Sh} & + & 0.01 \cdot x_{Cr} & + & 0 \cdot x_{Co} & + & 0.03 \cdot x_{Lo} & \leq & 7 \\
0.1 \cdot x_{Sh} & + & 0.05 \cdot x_{Cr} & + & 0.1 \cdot x_{Co} & + & 0.01 \cdot x_{Lo} & \leq & 10 \\
0 \cdot x_{Sh} & + & 0.2 \cdot x_{Cr} & + & 0.1 \cdot x_{Co} & + & 0.02 \cdot x_{Lo} & \leq & 13 \\
0 \cdot x_{Sh} & + & 0 \cdot x_{Cr} & + & 0.1 \cdot x_{Co} & + & 0.05 \cdot x_{Lo} & \leq & 6 \\
5 \cdot x_{Sh} & + & 5 \cdot x_{Cr} & + & 5 \cdot x_{Co} & + & 2 \cdot x_{Lo} & \leq & 400 \\
2.0 \cdot x_{Sh} & + & 3.5 \cdot x_{Cr} & + & 2.2 \cdot x_{Co} & + & 4.2 \cdot x_{Lo} & \leq & 350
\end{array}
$$

```
s.t. Herb1:
   0.1 * xSh + 0.01 * xCr + 0.03 * xLo <= 7;

s.t. Herb2:
   0.1 * xSh + 0.05 * xCr + 0.1 * xCo + 0.01 * xLo <= 10;

s.t. Herb3:
   0.2 * xCr + 0.1 * xCo + 0.02 * xLo <= 13;

s.t. Herb4:
   0.1 * xCo + 0.05 * xLo <= 6;

s.t. Water:
   5 * xSh + 3 * xCr + 3 * xCo + 2 * xLo <= 400;

s.t. CO2:
   2 * xSh + 3.5 * xCr + 2.2 * xCo + 4.2 * xLo <= 350;
```

# Objective

$$30 \cdot x_{Sh} + 25 \cdot x_{Cr} + 35 \cdot x_{Co} + 40 \cdot x_{Lo} \rightarrow \max$$

```
maximize profit:
  30 * xSh + 25 * xCr + 35 * xCo + 40 * xLo;
```

# Overall modell

```
var xSh >=0;
var xCr >=0;
var xCo >=0;
var xLo >=0;

s.t. Herb1:
    0.1 * xSh + 0.01 * xCr + 0.03 * xLo <= 7;

s.t. Herb2:
    0.1 * xSh + 0.05 * xCr + 0.1 * xCo + 0.01 * xLo <= 10;

s.t. Herb3:
    0.2 * xCr + 0.1 * xCo + 0.02 * xLo <= 13;

s.t. Herb4:
    0.1 * xCo + 0.05 * xLo <= 6;

s.t. Water:
    5 * xSh + 3 * xCr + 3 * xCo + 2 * xLo <= 400;

s.t. CO2:
    2 * xSh + 3.5 * xCr + 2.2 * xCo + 4.2 * xLo <= 350;

maximize profit:
  30 * xSh + 25 * xCr + 35 * xCo + 40 * xLo;
```

# Output file - Important parts

```
Status:        OPTIMAL
Objective:     profit = 4294.137931 (MAXimum)
```

The maximal profit is 4294.137931.

# Output file - Important parts

```
Status:       OPTIMAL
Objective:    profit = 4294.137931 (MAXimum)
```

The maximal profit is 4294.137931.

It can be achieved by producing 39.52 l of shampoo, 37.59 l of Conditioner, and 44.83 l of Lotion:

```
   No. Column name  St   Activity     Lower bound   Upper bound    Marginal
------ ------------ -- ------------- ------------- ------------- -------------
     1 xSh          B        39.5172             0
     2 xCr          NL             0             0                    -8.46552
     3 xCo          B        37.5862             0
     4 xLo          B        44.8276             0
```

# Reminder: the problem


| Distance (km) | Talsint | Khouribga | Ouarzazate | Nador | | Stock |
|---|---|---|---|---|---|---|
| Berkane | 500 | 600 | 1200 | 80 | | 100 |
| Driouch | 700 | 600 | 1000 | 60 | | 150 |
| Errachidia | 100 | 700 | 500 | 600 | | 300 |
| | | | | | | |
| Demand | 50 | 100 | 250 | 150 | | |

# Reminder: the model

$$x_{B\to T} + x_{B\to K} + x_{B\to O} + x_{B\to N} = 100$$
$$x_{D\to T} + x_{D\to K} + x_{D\to O} + x_{D\to N} = 150$$
$$x_{E\to T} + x_{E\to K} + x_{E\to O} + x_{E\to N} = 300$$
$$x_{B\to T} + x_{D\to T} + x_{E\to T} = 50$$
$$x_{B\to K} + x_{D\to K} + x_{E\to K} = 100$$
$$x_{B\to O} + x_{D\to O} + x_{E\to O} = 250$$
$$x_{B\to N} + x_{D\to N} + x_{E\to N} = 150$$

$$x_{B\to T}, x_{B\to K}, x_{B\to O}, x_{B\to N}, x_{D\to T}, x_{D\to K},$$
$$x_{D\to O}, x_{D\to N}, x_{E\to T}, x_{E\to K}, x_{E\to O}, x_{E\to N} \geq 0$$

minimize:

$$\begin{array}{ccccccccc} 500 \cdot x_{B\to T} & + & 600 \cdot x_{B\to K} & + & 1200 \cdot x_{B\to O} & + & 80 \cdot x_{B\to N} & + \\ 700 \cdot x_{D\to T} & + & 600 \cdot x_{D\to K} & + & 1000 \cdot x_{D\to O} & + & 60 \cdot x_{D\to N} & + \\ 100 \cdot x_{E\to T} & + & 700 \cdot x_{E\to K} & + & 500 \cdot x_{E\to O} & + & 600 \cdot x_{E\to N} & \end{array}$$

# Variable declrations

$$x_{B\to T}, x_{B\to K}, x_{B\to O}, x_{B\to N}, x_{D\to T}, x_{D\to K},$$
$$x_{D\to O}, x_{D\to N}, x_{E\to T}, x_{E\to K}, x_{E\to O}, x_{E\to N} \geq 0$$

```
var xBT>=0; var xBK>=0; var xBO>=0; var xBN>=0;
var xDT>=0; var xDK>=0; var xDO>=0; var xDN>=0;
var xET>=0; var xEK>=0; var xEO>=0; var xEN>=0;
```

## Constrainsts

$$x_{B\to T} + x_{B\to K} + x_{B\to O} + x_{B\to N} = 100$$

$$x_{D\to T} + x_{D\to K} + x_{D\to O} + x_{D\to N} = 150$$

$$x_{E\to T} + x_{E\to K} + x_{E\to O} + x_{E\to N} = 300$$

$$x_{B\to T} + x_{D\to T} + x_{E\to T} = 50$$

$$x_{B\to K} + x_{D\to K} + x_{E\to K} = 100$$

$$x_{B\to O} + x_{D\to O} + x_{E\to O} = 250$$

$$x_{B\to N} + x_{D\to N} + x_{E\to N} = 150$$

```
s.t. Supply_Berkane:
  xBT + xBK + xBO + xBN = 100;

s.t. Supply_Driouch:
  xDT + xDK + xDO + xDN = 150;

s.t. Supply_Errachidia:
  xET + xEK + xEO + xEN = 300;

s.t. Demand_Talsint:
  xBT + xDT + xET = 50;

s.t. Demand_Khouribga:
  xBK + xDK + xEK = 100;

s.t. Demand_Ouarzazate:
  xBO + xDO + xEO = 250;

s.t. Demand_Nador:
  xBN + xDN + xEN = 150;
```

# Objective

minimize:

$$
\begin{array}{ccccccccc}
500 \cdot x_{B\to T} & + & 600 \cdot x_{B\to K} & + & 1200 \cdot x_{B\to O} & + & 80 \cdot x_{B\to N} & + \\
700 \cdot x_{D\to T} & + & 600 \cdot x_{D\to K} & + & 1000 \cdot x_{D\to O} & + & 60 \cdot x_{D\to N} & + \\
100 \cdot x_{E\to T} & + & 700 \cdot x_{E\to K} & + & 500 \cdot x_{E\to O} & + & 600 \cdot x_{E\to N} &
\end{array}
$$

```
minimize transportation_cost:
  500*xBT + 600*xBK + 1200*xBO + 80*xBN + 700*
     xDT + 600*xDK + 1000*xDO + 60*xDN + 100*
     xET + 700*xEK + 500*xEO + 600*xEN;
```

# Overall modell

```
var xBT>=0; var xBK>=0; var xBO>=0; var xBN>=0;
var xDT>=0; var xDK>=0; var xDO>=0; var xDN>=0;
var xET>=0; var xEK>=0; var xEO>=0; var xEN>=0;


s.t. Supply_Berkane:
  xBT + xBK + xBO + xBN = 100;

s.t. Supply_Driouch:
  xDT + xDK + xDO + xDN = 150;

s.t. Supply_Errachidia:
  xET + xEK + xEO + xEN = 300;

s.t. Demand_Talsint:
  xBT + xDT + xET = 50;

s.t. Demand_Khouribga:
  xBK + xDK + xEK = 100;

s.t. Demand_Ouarzazate:
  xBO + xDO + xEO = 250;

s.t. Demand_Nador:
  xBN + xDN + xEN = 150;

minimize transportation_cost:
  500*xBT + 600*xBK + 1200*xBO + 80*xBN + 700*xDT + 600*xDK + 1000*xDO +
      60*xDN + 100*xET + 700*xEK + 500*xEO + 600*xEN;
```

# Output file - Important parts

```
Status:       OPTIMAL
Objective:    transportation_cost = 199000 (MINimum)
```

The minimal cost is 199000.

# Output file - Important parts

```
Status:       OPTIMAL
Objective:    transportation_cost = 199000 (MINimum)
```

The minimal cost is 199000.

It can be achieved by the following transportation plan:

| No. | Column name | St | Activity | Lower bound | Upper bound | Marginal |
|---|---|---|---|---|---|---|
| 1 | xBT | NL | 0 | 0 | | 500 |
| 2 | xBK | B | 100 | 0 | | |
| 3 | xBO | NL | 0 | 0 | | 800 |
| 4 | xBN | B | 0 | 0 | | |
| 5 | xDT | NL | 0 | 0 | | 720 |
| 6 | xDK | NL | 0 | 0 | | 20 |
| 7 | xDO | NL | 0 | 0 | | 620 |
| 8 | xDN | B | 150 | 0 | | |
| 9 | xET | B | 50 | 0 | | |
| 10 | xEK | B | 0 | 0 | | |
| 11 | xEO | B | 250 | 0 | | |
| 12 | xEN | NL | 0 | 0 | | 420 |

# Advanced GLPK models

If the data of the problem (number of wells, products, shops, ...) changes, the structure of the mathematical model remains the same, if the problem is the same.

# Advanced GLPK models

If the data of the problem (number of wells, products, shops, ...) changes, the structure of the mathematical model remains the same, if the problem is the same.

The model can be separated from the data, as it is not dependent on the data itself.

# Advanced GLPK models

If the data of the problem (number of wells, products, shops, . . . ) changes, the structure of the mathematical model remains the same, if the problem is the same.

The model can be separated from the data, as it is not dependent on the data itself.

If the moel and the data is in one file, the file should be built like this:

1. Model section (variables, constraints, objective function)
2. **data;** (Keyword)
3. Data section (sets, parameter values, etc.)
4. **end;** (Keyword)

# Sets and parameters

Instead of using the values of the exact problem, sets and parameters are introduced, that takes values in the data section.

# Sets and parameters

Instead of using the values of the exact problem, sets and parameters are introduced, that takes values in the data section.

Declarations extended:

**set** Sname ;
**param** Pname *{Domain}, default Nvalue* ;
**var** Vname *{Domain} Options* ;
**s.t.** Cname *{Domain}*: Constraint ;
**minimize/maximize** Oname: Objective ;

# Well installation example - model section

```
set Houses;
set Wells;

param covers{w in Wells, h in Houses}, default 0;

var install{w in Wells}, binary;

s.t. House_cover{h in Houses}:
  sum{w in Wells} install[w] * covers[w,h] >= 1;

minimize number_of_installed_wells:
  sum{w in Wells} install[w];

data;
```

# Well installation example - data section

```
data;

set Houses := Alae Ahmed Mimi Zarioh Omar Rajae;
set Wells:= W1 W2 W3 W4 W5;

param covers :
        Alae   Ahmed    Mimi   Zarioh   Omar   Rajae :=
   W1   1      1        1      .        .      .
   W2   .      1        1      .        1      .
   W3   .      .        1      .        1      1
   W4   1      .        .      1        .      .
   W5   .      .        .      1        1      1
   ;

end;
```

# Product mix - model section

```
set Products;
set Resources;

param ingredient{r in Resources,p in Products}, default 0;

param price{p in Products};

param stock{r in Resources};

var quantity{p in Products}>=0;


s.t. stock_constraint{r in Resources}:
  sum{p in Products} quantity[p] * ingredient[r,p] <= stock[r];

maximize profit:
  sum{p in Products} price[p]*quantity[p];

data;
```

# Product mix - data section

```
data;

set Products := Shampoo Cream Conditioner Lotion;
set Resources := Herb1 Herb2 Herb3 Herb4 Water CO2;

param ingredient:
          Shampoo    Cream    Conditioner    Lotion    :=
  Herb1   0.1        0.01     .              0.03
  Herb2   0.1        0.05     0.1            0.01
  Herb3   .          0.2      0.1            0.02
  Herb4   .          .        0.1            0.05
  Water   5          3        3              2
  CO2     2          3.5      2.2            4.2
;

param price:=
  Shampoo     30
  Cream       25
  Conditioner 35
  Lotion      40
;

param stock:=
  Herb1 7
  Herb2 10
  Herb3 13
  Herb4 6
  Water 400
  CO2   350
;
```

# Transportation problem - model section

```
set Farms;
set Shops;

param distance{f in Farms, s in Shops}, default 999999;

param stock{f in Farms};
param demand{s in Shops};

var transported{f in Farms, s in Shops}>=0;

s.t. Supply{f in Farms}:
  sum{s in Shops} transported[f,s] = stock[f];

s.t. Demand{s in Shops}:
  sum{f in Farms} transported[f,s] = demand[s];

minimize transportation_cost:
  sum{f in Farms, s in Shops} transported[f,s]*distance[f,s];

data;
```

# Transportation problem - data section

```
data;

set Farms:= Berkane Driouch Errachidia;
set Shops:= Talsint Khouribga Ouarzazate Nador;

param distance: Talsint Khouribga Ouarzazate Nador  :=
    Berkane      500     600        1200        80
    Driouch      700     600        1000        60
    Errachidia   100     700        500         600
;

param stock:=
  Berkane    100
  Driouch    150
  Errachidia 300
;

param demand:=
  Talsint    50
  Khouribga  100
  Ouarzazate 250
  Nador      150
;
```

# Lake Balaton at sunset