# Tutorial AMPL

Eduardo Camponogara

Department of Automation and Systems Engineering
Federal University of Santa Catarina

October 2016

# Summary

Linear Programming

Duality

# LP – Example 3

**AMPL Model**:

- Consider a factory that produces $N$ products ($n$) in serial pipeline.
- For each product $n$, a profit $p_n$ is made on each unit sold, there is a weekly demand $d_n$ and production capacity $r_n$ in hours.
- We wish to determine the weekly production $x_n$, for each product $n$, so as to maximize the total profit in sales, given that the factory has a total number of $T$ hours per week.
- Model the problem in linear programming.

# LP – Example 3

**AMPL Model**:

- Consider a factory that produces $N$ products ($n$) in serial pipeline.
- For each product $n$, a profit $p_n$ is made on each unit sold, there is a weekly demand $d_n$ and production capacity $r_n$ in hours.
- We wish to determine the weekly production $x_n$, for each product $n$, so as to maximize the total profit in sales, given that the factory has a total number of $T$ hours per week.
- Model the problem in linear programming.

# LP – Example 3

**AMPL Model**:

- Consider a factory that produces $N$ products ($n$) in serial pipeline.
- For each product $n$, a profit $p_n$ is made on each unit sold, there is a weekly demand $d_n$ and production capacity $r_n$ in hours.
- We wish to determine the weekly production $x_n$, for each product $n$, so as to maximize the total profit in sales, given that the factory has a total number of $T$ hours per week.
- Model the problem in linear programming.

# LP – Example 3

**Linear programming model**:

$$
\begin{aligned}
\max \quad & \sum_{n=1}^{N} p_n \cdot x_n \\
\text{s.t. :} \ & \sum_{n=1}^{N} \frac{1}{r_n} \cdot x_n \le T \\
& 0 \le x_n \le d_n, \ n = 1 \ldots N
\end{aligned}
$$

# LP – Example 3

- Complete the AMPL model **examplo3.mod** according with your mathematical programming model.

```
# Part 1 – Variable Declaration (var, set, param, etc)
param N;
param T;
set Products := {1..N};
set InfoType := {p,r,d};
param data {Products,InfoType};
```

- Create a file **example3.run**.
- Use the file **example3.dat**. To this end, just issue the command "**data example3.dat;**" after the command that includes the .mod fiel in **example3.run**.

# LP – Example 3

**example3.dat**:

```
param N := 4; #Number of products
param T := 40; #Number of working hours per week
# Data Matrix
param data_base:
p  r   d   :=
1  1   40  1000
2  1.5 30  900
3  1   50  500
4  1.5 20  800
;
```

# LP – Example 3

**example3.mod**:

```
# Part 1: Variable Declaration (var, set, param, etc)
param N;
param T;
set Products := {1..N};
set InfoType := {p,r,d};
param data_base{Produto,InfoType};
var x{Products} >= 0;
# Part 2: Objective Function
maximize Profit: sum{n in Products} data_base[n,'p']*x[n];
# Part 3: Constraints
subject to available_working_hours: sum{n in Products}
(1/data_base[n,'r'])*x[n] <= T;
subject to demand{n in Products}: x[n] <= data_base[n,'d'];
```

# LP – Example 3

**example3.mod**:

```
# Part 1: Variable Declaration (var, set, param, etc)
param N;
param T;
set Products := {1..N};
set InfoType := {p,r,d};
param data_base{Produto,InfoType};
var x{Products} >= 0;
# Part 2: Objective Function
maximize Profit: sum{n in Products} data_base[n,'p']*x[n];
# Part 3: Constraints
subject to available_working_hours: sum{n in Products}
(1/data_base[n,'r'])*x[n] <= T;
subject to demand{n in Products}: x[n] <= data_base[n,'d'];
```

# LP – Example 3

**example3.mod**:

```
# Part 1: Variable Declaration (var, set, param, etc)
param N;
param T;
set Products := {1..N};
set InfoType := {p,r,d};
param data_base{Produto,InfoType};
var x{Products} >= 0;
# Part 2: Objective Function
maximize Profit: sum{n in Products} data_base[n,'p']*x[n];
# Part 3: Constraints
subject to available_working_hours: sum{n in Products}
(1/data_base[n,'r'])*x[n] <= T;
subject to demand{n in Products}: x[n] <= data_base[n,'d'];
```

# LP – Example 3

**example3.run**:

```
# Reset Memory
reset ;
# Load Memory
model example3.mod;
# Load Data
data exemple3.dat;
# Change Configurations (optional)
option solver cplex;
# Solve Problem
solve;
# Show Results
display x;
```

# LP – Example 3

**example3.run**:

```
# Reset Memory
reset ;
# Load Memory
model example3.mod;
# Load Data
data exemple3.dat;
# Change Configurations (optional)
option solver cplex;
# Solve Problem
solve;
# Show Results
display x;
```

# LP – Example 3

**example3.run**:

```
# Reset Memory
reset ;
# Load Memory
model example3.mod;
# Load Data
data exemple3.dat;
# Change Configurations (optional)
option solver cplex;
# Solve Problem
solve;
# Show Results
display x;
```

# Dual – Example 4

**Obtain the dual model in linear programming for Example 3**

$$\max \quad \sum_{n=1}^{N} p_n \cdot x_n$$

$$\text{s.t.}: \sum_{n=1}^{N} \frac{1}{r_n} \cdot x_n \leq T$$

$$0 \leq x_n \leq d_n, \; n = 1 \ldots N$$

# Dual – Example 4

**Dual of Example 3**

$$\min \quad T \cdot w + \sum_{n=1}^{N} d_n \cdot y_n$$

$$\text{s.t.}: \frac{1}{r_n} \cdot w + y_n \geq p_n, \; n = 1 \ldots N$$

$$w, \; y_n \geq 0$$

**Create files example4.mod, example4.dat e example4.run**

*Hint: Copy the content of the files example3.dat to example4.dat.*

# Challenge 1

**Minimum Cost Network Flow Problem**

- a directed graph $G = (V, A)$ with set of vertices (nodes) and a set of arcs;
- unit cost $c_{ij}$ for transportation in arc $(i, j)$;
- lower bound $l_{ij}$ and upper bound $u_{ij}$ for flow in arc $(i, j)$; and
- Flow $b_i$ that must be injected or consumed at node $i$:
  - if $b_i > 0$, then $i$ is a supplier node;
  - if $b_i < 0$, then $i$ is a consumer node; and
  - if $b_i = 0$, then $i$ is a transshipment node.

# Challenge 1

**Minimum Cost Network Flow Problem**

- a directed graph $G = (V, A)$ with set of vertices (nodes) and a set of arcs;
- unit cost $c_{ij}$ for transportation in arc $(i, j)$;
- lower bound $l_{ij}$ and upper bound $u_{ij}$ for flow in arc $(i, j)$; and
- Flow $b_i$ that must be injected or consumed at node $i$:
  - if $b_i > 0$, then $i$ is a supplier node;
  - if $b_i < 0$, then $i$ is a consumer node; and
  - if $b_i = 0$, then $i$ is a transshipment node.

# Challenge 1

**Minimum Cost Network Flow Problem**

- a directed graph $G = (V, A)$ with set of vertices (nodes) and a set of arcs;
- unit cost $c_{ij}$ for transportation in arc $(i, j)$;
- lower bound $l_{ij}$ and upper bound $u_{ij}$ for flow in arc $(i, j)$; and
- Flow $b_i$ that must be injected or consumed at node $i$:
  - if $b_i > 0$, then $i$ is a supplier node;
  - if $b_i < 0$, then $i$ is a consumer node; and
  - if $b_i = 0$, then $i$ is a transshipment node.

# Challenge 1

Model in mathematical programming:

$$\text{Minimize} \quad \sum_{(i,j)\in A} c_{ij}x_{ij}$$

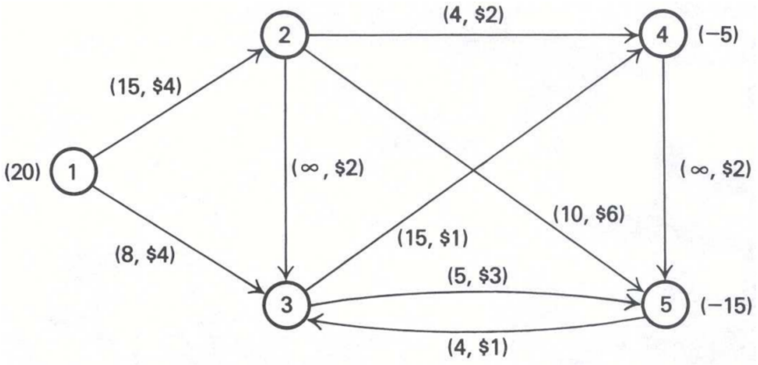
Subject to:

$$\sum_{\{j:(i,j)\in A\}} x_{ij} - \sum_{\{j:(j,i)\in A\}} x_{ji} = b_{ij}, \forall i \in V$$

$$l_{ij} \leq x_{ij} \leq u_{ij}, \forall (i,j) \in A$$

# Challenge 1

Describe the following problem in AMPL. Create .dat, .mod and .run files.



# Tutorial AMPL

- Thank you for attending this lecture!!!