Tutorial AMPL

Parte III

Eduardo Camponogara

Department of Automation and Systems Engineering
Federal University of Santa Catarina

October 2016

Example 5

Example 6

Example 7

Summary

Integer Programming – Example 5

AMPL Model: The decision variables x_1, x_2, x_3, x_4 are binary. We wish to maximize the weighted sum:

$$4x_1 + 3x_2 + 2x_3 + 1x_4$$

However, these variables are subject to a number of rules/conditions:

1. $x_1 = 1$ or $x_2 = 1$ (or);
2. $x_1^2 = x_2 + x_3$;
3. $x_2 = 1$ only if $x_4 = 1$ (implication);
4. only two variables may assume value 1 simultaneously.

Task: model the problem as an integer program.

Integer Programming – Example 5

AMPL Model: The decision variables x_1, x_2, x_3, x_4 are binary. We wish to maximize the weighted sum:

$$4x_1 + 3x_2 + 2x_3 + 1x_4$$

However, these variables are subject to a number of rules/conditions:

1. $x_1 = 1$ or $x_2 = 1$ (or);
2. $x_1^2 = x_2 + x_3$;
3. $x_2 = 1$ only if $x_4 = 1$ (implication);
4. only two variables may assume value 1 simultaneously.

Task: model the problem as an integer program.

Integer Programming – Example 5

AMPL Model: The decision variables x_1, x_2, x_3, x_4 are binary. We wish to maximize the weighted sum:

$$4x_1 + 3x_2 + 2x_3 + 1x_4$$

However, these variables are subject to a number of rules/conditions:

1. $x_1 = 1$ or $x_2 = 1$ (or);
2. $x_1^2 = x_2 + x_3$;
3. $x_2 = 1$ only if $x_4 = 1$ (implication);
4. only two variables may assume value 1 simultaneously.

Task: model the problem as an integer program.

Integer Programming – Example 5

Mathematical Programming Model:

$$
\begin{aligned}
\max \quad & 4x_1 + 3x_2 + 2x_3 + 1x_4 \\
\text{s.t. :} \quad & x_1 + x_2 \geq 1 \\
& x_1 = x_2 + x_3 \\
& x_2 \leq x_4 \\
& x_1 + x_2 + x_3 + x_4 \leq 2 \\
& x_1, x_2, x_3, x_4 \in \{0, 1\}
\end{aligned}
$$

Integer Programming – Example 5

example5.mod:

```
# Part 1: Variable Declaration (var, set, param, etc)
set K = 1..4 by 1;
var x{k in K} binary;
param c{k in K};
let {k in K} c[k] := card(K)-k+1;
# Part 2: Objective Function
maximize objective: sum{k in K} c[k]*x[k];
# Part 3: Constraints
subject to R1: x[1] + x[2] >= 1;
subject to R2: x[1] = x[2] + x[3];
subject to R3: x[2] <= x[4];
subject to R4: x[1] + x[2] + x[3] + x[4] <=2;
```

Integer Programming – Example 5

example5.mod:

```
# Part 1: Variable Declaration (var, set, param, etc)
set K = 1..4 by 1;
var x{k in K} binary;
param c{k in K};
let {k in K} c[k] := card(K)-k+1;
# Part 2: Objective Function
maximize objective: sum{k in K} c[k]*x[k];
# Part 3: Constraints
subject to R1: x[1] + x[2] >= 1;
subject to R2: x[1] = x[2] + x[3];
subject to R3: x[2] <= x[4];
subject to R4: x[1] + x[2] + x[3] + x[4] <=2;
```

Integer Programming – Example 5

example5.mod:

```
# Part 1: Variable Declaration (var, set, param, etc)
set K = 1..4 by 1;
var x{k in K} binary;
param c{k in K};
let {k in K} c[k] := card(K)-k+1;
# Part 2: Objective Function
maximize objective: sum{k in K} c[k]*x[k];
# Part 3: Constraints
subject to R1: x[1] + x[2] >= 1;
subject to R2: x[1] = x[2] + x[3];
subject to R3: x[2] <= x[4];
subject to R4: x[1] + x[2] + x[3] + x[4] <=2;
```

Summary

Integer Programming – Example 6

Consider the problem of Example 3:

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

Add the following constraint:

- If more than 300 items of product 2 are manufactured weekly, then at least 200 items of product 1 must be produced.
- Also, a client will pay a bonus of $B_4 = 7$ for each package of 10 items of product 4 delivered weekly.

Integer Programming – Example 6

Consider the problem of Example 3:

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

Add the following constraint:

- If more than 300 items of product 2 are manufactured weekly, then at least 200 items of product 1 must be produced.
- Also, a client will pay a bonus of $B_4 = 7$ for each package of 10 items of product 4 delivered weekly.

Integer Programming – Example 6

Mathematical programming model:

$$\max \quad B_4 \cdot w + \sum_{n=1}^{4} p_n \cdot x_n$$

$$\text{s.t.} : \sum_{n=1}^{4} \frac{1}{r_n} \cdot x_n \leq T$$

$$0 \leq x_n \leq d_n,\ n = 1 \ldots 4$$

$$300 \cdot z \leq x_2 \leq 300 + M \cdot z$$

$$200 \cdot z \leq x_1$$

$$10 \cdot w \leq x_4$$

$$x \in \mathbb{R}^4,\ z \in \{0, 1\} \text{ and } w \in \mathbb{Z}$$

Integer Programming – Example 6

Taking as a starting point the AMPL models for Example 3, develop the files:

- **example6.dat**,
- **example6.run**, and
- **example6.mod**

Implement the required extension in **example6.mod** to account for the new specifications.

Summary

Integer Programming – Example 7

Branch-and-Bound Algorithm:

- Consider Problem 6 described above. Notice that this problem has binary and integer variables.
- Using the AMPL code, relax the integrality constraints and apply the Branch-and-Bound Algorithm.
- Generate the B-&-B tree iteratively.

AMPL Tutorial

- Thank you for attending this lecture!!!