Integer Programming

Dr. M. Angélica Salazar A. / Dr. Vincent Boyer

Graduate Program in Systems Engineering (PISIS)
Facultad de Ingeniería Mecánica y Eléctrica
Universidad Autónoma de Nuevo León

Content

Agenda

1. Examples of Integer and Mixed Integer Programming
2. Modeling
3. Branch & Bound Method

Definition

Definition

In an Integer Programming (IP) model, **some** or **all** of the decision variables are required to be integer.

Type of IPs

- **Total (or Pure) Integer Model:** All the decision variables are integer.
- **0-1 Integer Model (or Binary Model:** The decision variables can only take the values 0 or 1.
- **Mixed Integer Model:** Some decision variables are integer and others can be non-integer (fractional).

The carpenter's problem

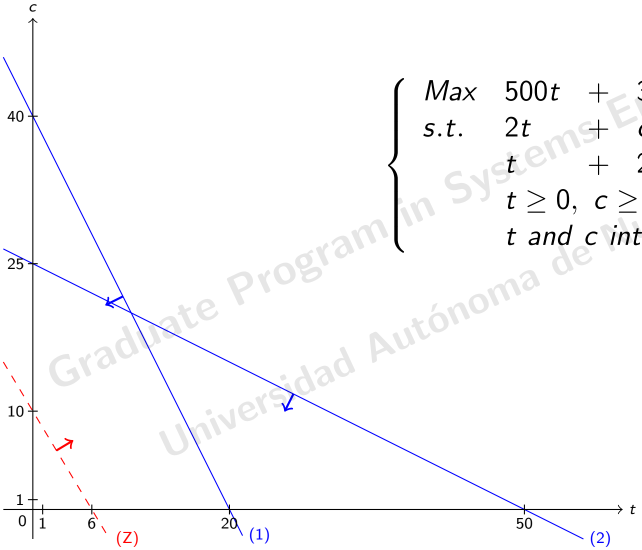
Decision variables

- Let t be the number of tables to be produced.
- Let c be the number of chairs to be produced.

The IP model

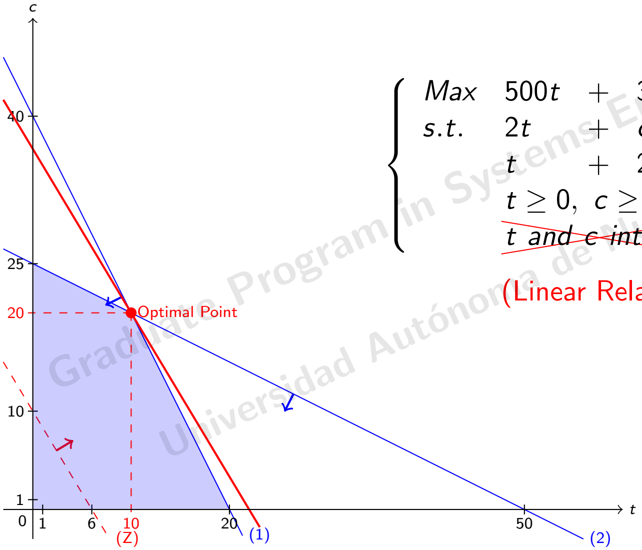
$$\begin{cases} Max & 500t & + & 300c & = & z \\ s.t. & 2t & + & c & \leq & 40 \\ & t & + & 2c & \leq & 50 \\ & t \geq 0, \;\; c \geq 0 \\ & t \text{ and } c \text{ integer} \end{cases}$$

The carpenter's problem



$$\begin{cases} Max & 500t & + & 300c & = & z & \\ s.t. & 2t & + & c & \leq & 40 & (1) \\ & t & + & 2c & \leq & 50 & (2) \\ & t \geq 0, \; c \geq 0 & & & & & \\ & t \text{ and } c \text{ integer} & & & & & \end{cases}$$

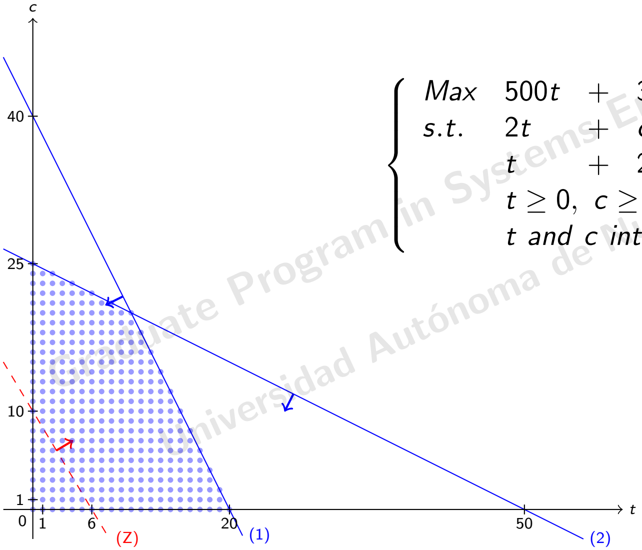
The carpenter's problem



$$\begin{cases} Max & 500t & + & 300c & = & z & \\ s.t. & 2t & + & c & \leq & 40 & (1) \\ & t & + & 2c & \leq & 50 & (2) \\ & t \geq 0, & c \geq 0 & & & & \\ & \cancel{t \text{ and } c \text{ integer}} & & & & & \end{cases}$$

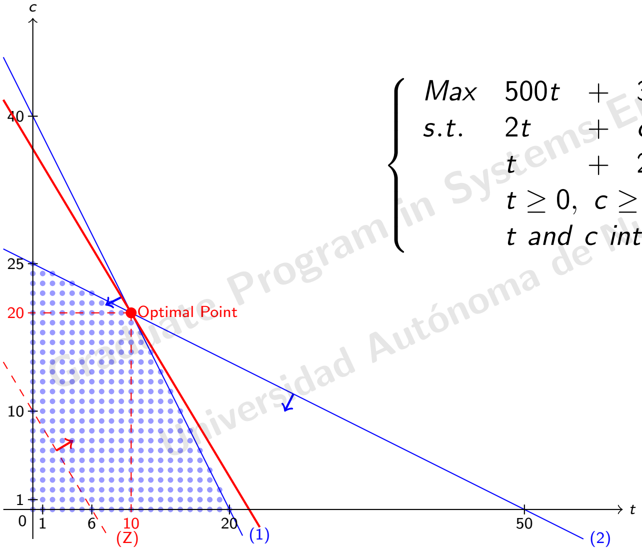
(Linear Relaxation)

The carpenter's problem



$$\begin{cases} Max & 500t & + & 300c & = & z & \\ s.t. & 2t & + & c & \leq & 40 & (1) \\ & t & + & 2c & \leq & 50 & (2) \\ & t \geq 0, & c \geq 0 & & & & \\ & t \text{ and } c \text{ integer} & & & & & \end{cases}$$

The carpenter's problem



$$\begin{cases} Max & 500t & + & 300c & = & z & \\ s.t. & 2t & + & c & \leq & 40 & (1) \\ & t & + & 2c & \leq & 50 & (2) \\ & t \geq 0, & c \geq 0 & & & & \\ & t \text{ and } c \text{ integer} & & & & & \end{cases}$$

Dorian's problem

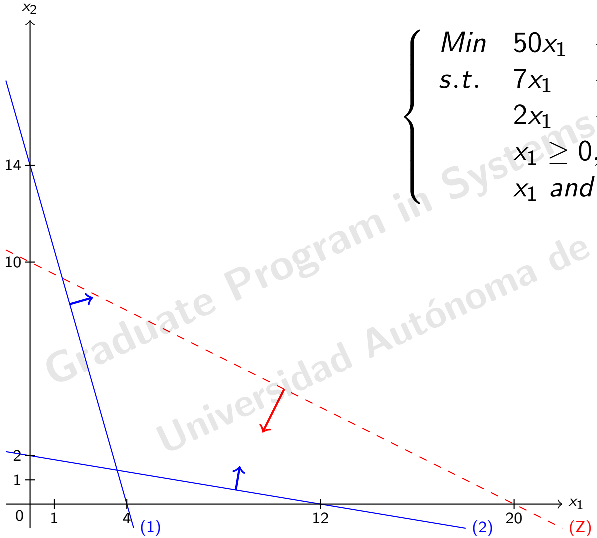
Decision variables

- x_1: number of 1-minute comedy ads to purchase.
- x_2: number of 1-minute football ads to purchase.

The IP model

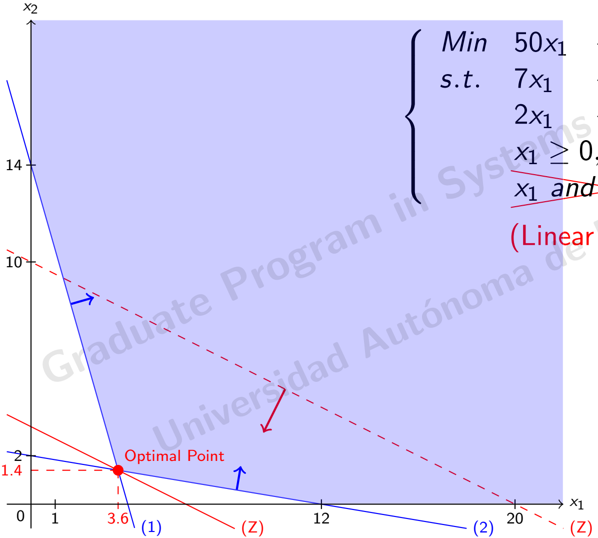
$$\begin{cases} Min & 50x_1 & + & 100x_2 & & \\ s.t. & 7x_1 & + & 2x_2 & \geq & 28 \\ & 2x_1 & + & 12x_2 & \geq & 24 \\ & x_1 \geq 0, \ x_2 \geq 0 & & & & \\ & x_1 \ \textit{and} \ x_2 \ \textit{integer} & & & & \end{cases}$$

Dorian's problem



$$\begin{cases} Min & 50x_1 & + & 100x_2 & & & \\ s.t. & 7x_1 & + & 2x_2 & \geq & 28 & (1) \\ & 2x_1 & + & 12x_2 & \geq & 24 & (2) \\ & x_1 \geq 0, \; x_2 \geq 0 & & & & & \\ & x_1 \text{ and } x_2 \text{ integer} & & & & & \end{cases}$$

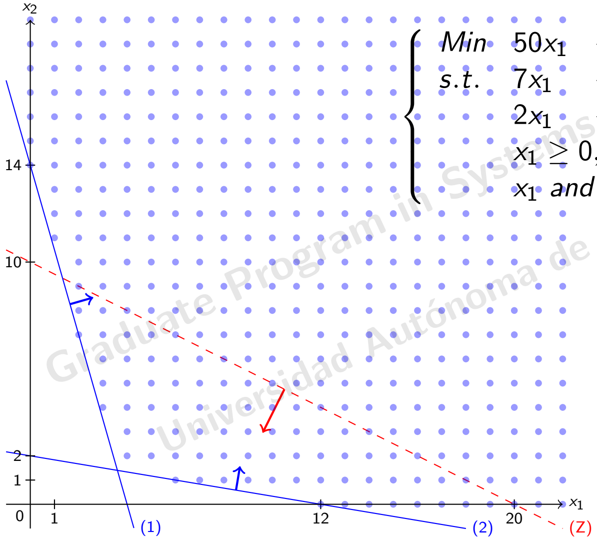
Dorian's problem



$$\begin{cases} Min & 50x_1 & + & 100x_2 & & & \\ s.t. & 7x_1 & + & 2x_2 & \geq & 28 & (1) \\ & 2x_1 & + & 12x_2 & \geq & 24 & (2) \\ & x_1 \geq 0, \ x_2 \geq 0 & & & & & \\ & \cancel{x_1 \text{ and } x_2 \text{ integer}} & & & & & \end{cases}$$

(Linear Relaxation)

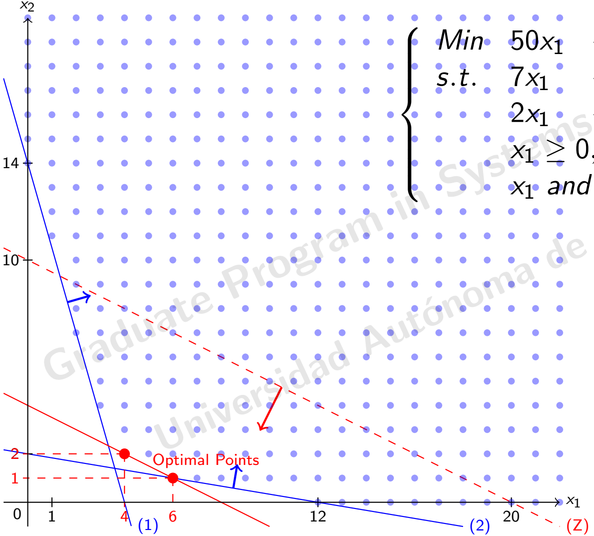
Dorian's problem



$$\left\{\begin{array}{lllll} Min & 50x_1 & + & 100x_2 & & \\ s.t. & 7x_1 & + & 2x_2 & \geq & 28 & (1) \\ & 2x_1 & + & 12x_2 & \geq & 24 & (2) \\ & x_1 \geq 0, \ x_2 \geq 0 & & & & \\ & x_1 \text{ and } x_2 \text{ integer} & & & & \end{array}\right.$$

Dorian's problem

$$
\begin{cases}
Min & 50x_1 & + & 100x_2 & & & \\
s.t. & 7x_1 & + & 2x_2 & \geq & 28 & (1) \\
 & 2x_1 & + & 12x_2 & \geq & 24 & (2) \\
 & x_1 \geq 0, \ x_2 \geq 0 & & & & & \\
 & x_1 \ and \ x_2 \ integer & & & & &
\end{cases}
$$



Agenda

1. Examples of Integer and Mixed Integer Programming
2. Modeling
3. Branch & Bound Method

Assignment problem

A company has 4 machines available for assignment to 4 tasks. Any machine can be assigned to any task, and each task requires to be processed by one machine. The time required to set up each machine for the processing of each task is given in the table below.

	Time (hrs)			
	Task 1	Task 2	Task 3	Task 4
Machine 1	13	4	7	6
Machine 2	1	11	5	4
Machine 3	6	7	2	8
Machine 4	1	3	5	9

The company wants to minimize the total setup time needed for the processing of all four tasks.

Mathematical model

Decision variables:

x_{ij}= 1 if machine i is assigned to task j, 0 otherwise.

Objective Function:

Min $z = 13x_{11} + 4x_{12} + 7x_{13} + 6x_{14} + 1x_{21} + 11x_{22} + 5x_{23} + 4x_{24} + 6x_{31} + 7x_{32} + 2x_{33} + 8x_{34} + 1x_{41} + 3x_{42} + 5x_{43} + 9x_{44}$

Constraints:

$$\left.\begin{aligned} x_{11} + x_{12} + x_{13} + x_{14} &= 1 \\ x_{21} + x_{22} + x_{23} + x_{24} &= 1 \\ x_{31} + x_{32} + x_{33} + x_{34} &= 1 \\ x_{41} + x_{42} + x_{43} + x_{44} &= 1 \end{aligned}\right\} \text{(Assignment of machines to tasks)}$$

$$\left.\begin{aligned} x_{11} + x_{21} + x_{31} + x_{41} &= 1 \\ x_{12} + x_{22} + x_{32} + x_{42} &= 1 \\ x_{13} + x_{23} + x_{33} + x_{43} &= 1 \\ x_{14} + x_{24} + x_{34} + x_{44} &= 1 \\ x_{ij} \in \{0,1\} & \end{aligned}\right\} \text{(Assignment of tasks to machines)}$$

How is the set of feasible solutions?

For the first machine we have 4 options of tasks (n).
Once we assign one task to the first machine, we have 3 options of tasks for the second one ($n-1$). If we assigned tasks to the first and second machines, we have 2 options of tasks for the third machine ($n-2$).
Hence, after the assignment of tasks to the first, second, and third machines, the fourth machine has only one option of task.

Number of feasible solutions

n	$n!$
3	6
5	120
10	3628800
100	9.33×10^{157}
1000	4.02×10^{2567}

The number of feasible solutions increases exponentially with n.

Knapsack Problem

Consider a set of items each of which has a predefined weight and a monetary value (profit). These items have to be packed into a bag whose maximum weight is limited in such a way that it can not carry all the available items. The problem consists in choosing a subset of items which:

- 1. fit into the bag with respect to the weight limit and
- 2. yield a maximum total profit

Formal Description: The Knapsack Problem

Suppose there is one knapsack with capacity Q and N items. Item j has weight w_j and value c_j for $j = 1, ..., N$. The problem consists in choosing a subset of items such that one obtained the maximal value under the constraint that the maximal weight is less than or equal to Q.

How can we formulate this problem?

Modeling the Knapsack Problem

Decision variables

$x_j = 1$ if item j is chosen, 0 otherwise.

Objective function

$$\text{Max } z = \sum_{j=1}^{N} c_j x_j \text{ (profit)}$$

Constraints

$$\begin{aligned} \sum_{j=1}^{N} w_j x_j &\leq Q && \text{(the capacity should not be exceeded)} \\ x_j &\in \{0,1\} \quad j = 1, \ldots, N \end{aligned}$$

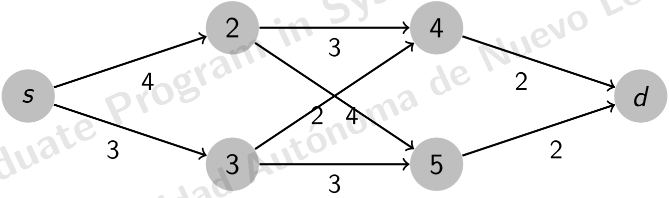
Knapsack instance

Product	Weight	Profit
1	3.2 tons	$727
2	4.0 tons	$763
3	4.4 tons	$60
4	2.0 tons	$606
5	0.1 tons	$45
6	2.9 tons	$370
7	0.3 tons	$414
8	1.3 tons	$880
9	0.6 tons	$133
10	3.9 tons	$820

Assume that the capacity is 11.3 tons and write the mathematical model for this instance of the problem.

Shortest Path Problem

Given a set of vertices V, a source vertex s, a destination vertex d, where $s, d \in V$, and a set of weighted edges E, over the set V, find the shortest-path between s and d that has the minimum weight. Notice that the edges can be directed or undirected. If they are directed, we will call them arcs.



How can we formulate this problem?

Modeling the Shortest Path Problem

Decision variables:

x_{ij}= 1 if arc (i, j) is chosen, 0 otherwise.

Objective function:

$$\text{Min } z = \sum_{(i,j)\in A} d_{ij}x_{ij} \text{ (total distance)}$$

Constraints:

$$\sum_{(s,j)\in A} x_{sj} = 1 \qquad (\textit{source})$$

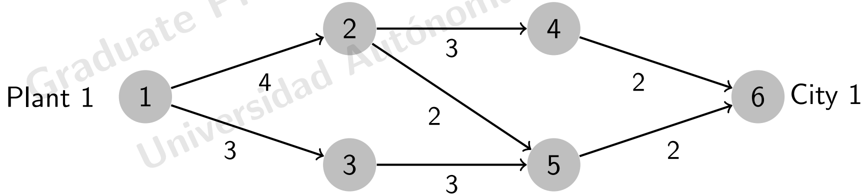
$$\sum_{(i,d)\in A} x_{id} = 1 \qquad (\textit{destination})$$

$$\sum_{(k,i)\in A} x_{ki} - \sum_{(i,j)\in A} x_{ij} = 0 \qquad i \neq s, d$$

$$x_{ij} \in \{0, 1\} \qquad (i, j) \in A$$

Shortest Path Problem

Suppose that when power is sent from plant 1 to city 1, it must pass through relay substations. For any pair of nodes between which power can be transported, the distance is known in advance (see the graph). Thus, substations 2 and 4 are 3 miles apart, and power cannot be sent between substations 4 and 5. Powerco wants the power sent from plant 1 to city 1 to travel the minimum possible distance, so it must find the shortest path in the graph that joins node 1 to node 6.



Agenda

Branch & Bound Method

Step 1. Build the first pending node from the linear relaxation

Step 2. Select a pending node and solve the associated sub-problem

Step 3. Branch on a fractional variable if it exists

Step 4. Go to Step 2 if there are still some pending nodes

Branch & Bound Method

Step 1. Build the first pending node from the linear relaxation

Integer Model $\Rightarrow$ **Linear Relaxation**

$$\begin{cases} Max & 10x_1 & + & 15x_2 & = & z \\ s.t. & 8x_1 & + & 4x_2 & \leq & 40 \\ & 15x_1 & + & 30x_2 & \leq & 200 \\ & \multicolumn{5}{l}{x_1 \geq 0,\ \ x_2 \geq 0} \\ & \multicolumn{5}{l}{x_1 \ \ and \ \ x_2 \ \ integer} \end{cases} \Rightarrow \begin{cases} Max & 10x_1 & + & 15x_2 & = & z \\ s.t. & 8x_1 & + & 4x_2 & \leq & 40 \\ & 15x_1 & + & 30x_2 & \leq & 200 \\ & \multicolumn{5}{l}{x_1 \geq 0,\ \ x_2 \geq 0} \end{cases}$$

Root (0)	
x_1	$= \dots$
x_2	$= \dots$
z	$= \dots$

Branch & Bound Method

Step 2. Select a pending node and solve the associated sub-problem

Root (0)	
x_1	$= \ldots$
x_2	$= \ldots$
z	$= \ldots$

Branch & Bound Method

Step 2. Select a pending node and solve the associated sub-problem

Root (0)	
x_1	$= \dots$
x_2	$= \dots$
z	$= \dots$

Branch & Bound Method

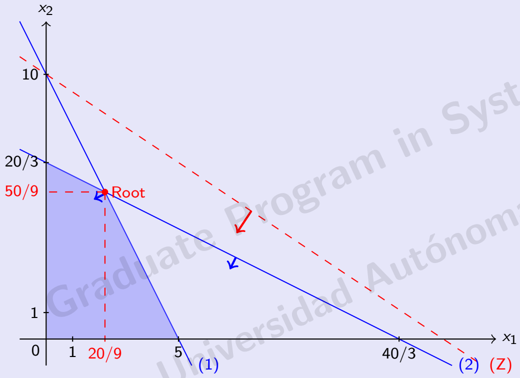
Step 2. Select a pending node and solve the associated sub-problem

Root (0)	
x_1	$= \dots$
x_2	$= \dots$
z	$= \dots$

$$
\begin{cases}
Max & 10x_1 + 15x_2 = z & \\
s.t. & 8x_1 + 4x_2 \leq 40 & (1) \\
 & 15x_1 + 30x_2 \leq 200 & (2) \\
 & x_1 \geq 0,\ x_2 \geq 0 &
\end{cases}
$$

Branch & Bound Method

Step 2. Select a pending node and solve the associated sub-problem

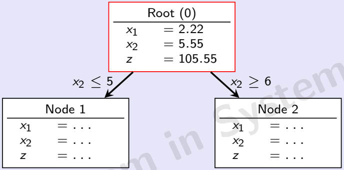


Root (0)	
x_1	$= 2.22$
x_2	$= 5.55$
z	$= 105.55$

$$\begin{cases} Max & 10x_1 + 15x_2 = z \\ s.t. & 8x_1 + 4x_2 \leq 40 \quad (1) \\ & 15x_1 + 30x_2 \leq 200 \quad (2) \\ & x_1 \geq 0, \; x_2 \geq 0 \end{cases}$$

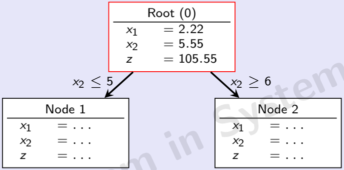
Branch & Bound Method

Step 3. Branch on a fractional variable if it exists



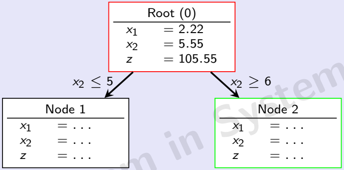
Branch & Bound Method

Step 4. Go to Step 2 if there are still some pending nodes



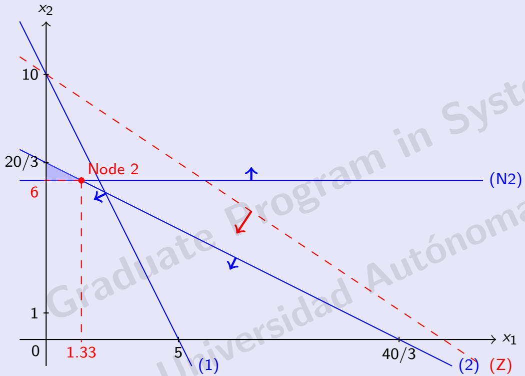
Branch & Bound Method

Step 2. Select a pending node and solve the associated sub-problem



Branch & Bound Method

Step 2. Select a pending node and solve the associated sub-problem

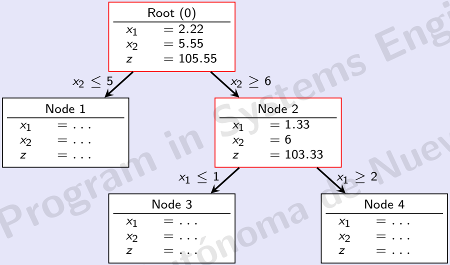


Node 2	
x_1	$= 1.33$
x_2	$= 6$
z	$= 103.33$

$$\begin{cases} Max & 10x_1 & + & 15x_2 & = & z & \\ s.t. & 8x_1 & + & 4x_2 & \leq & 40 & (1) \\ & 15x_1 & + & 30x_2 & \leq & 200 & (2) \\ & & & x_2 & \geq & 6 & (N2) \\ & x_1 \geq 0, & x_2 \geq 0 & & & & \end{cases}$$

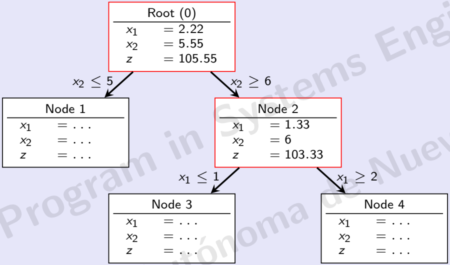
Branch & Bound Method

Step 3. Branch on a fractional variable if it exists



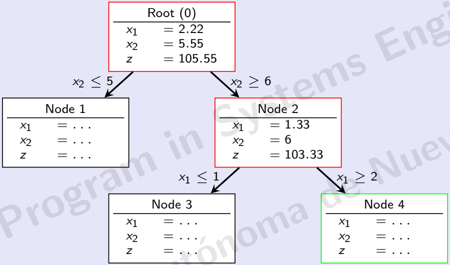
Branch & Bound Method

Step 4. Go to Step 2 if there are still some pending nodes



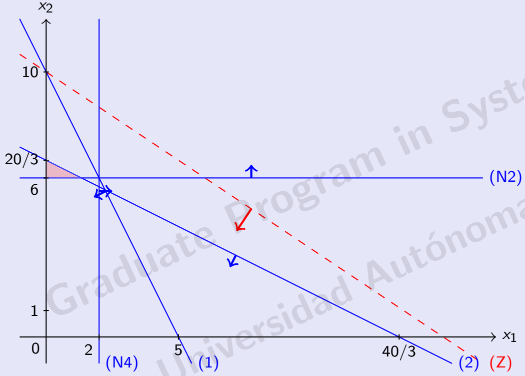
Branch & Bound Method

Step 2. Select a pending node and solve the associated sub-problem



Branch & Bound Method

Step 2. Select a pending node and solve the associated sub-problem

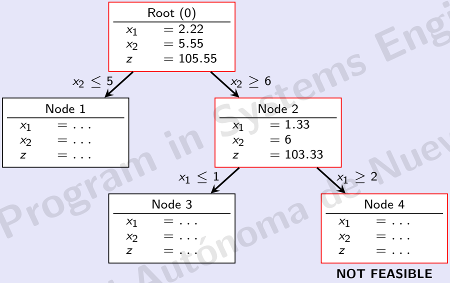


Node 4	
x_1	$= \dots$
x_2	$= \dots$
z	$= \dots$

$$\begin{cases} \begin{array}{llllllll} Max & 10x_1 & + & 15x_2 & = & z & \\ s.t. & 8x_1 & + & 4x_2 & \leq & 40 & (1) \\ & 15x_1 & + & 30x_2 & \leq & 200 & (2) \\ & & & x_2 & \geq & 6 & (N2) \\ & x_1 & & & \geq & 2 & (N4) \\ & x_1 \geq 0, & x_2 \geq 0 & & & & \end{array} \end{cases}$$

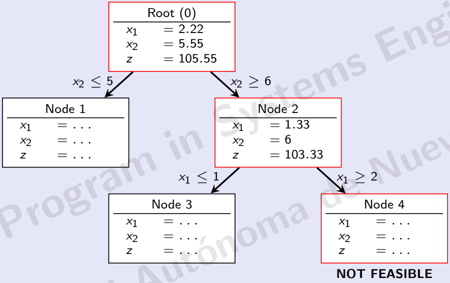
Branch & Bound Method

Step 3. Branch on a fractional variable if it exists



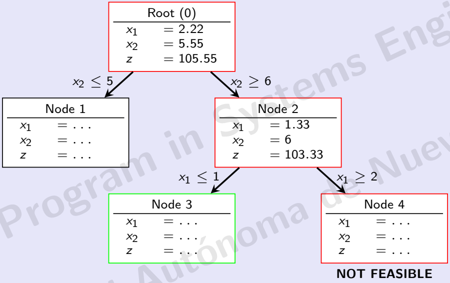
Branch & Bound Method

Step 4. Go to Step 2 if there are still some pending nodes



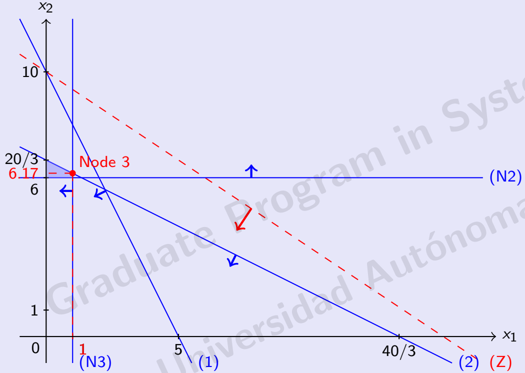
Branch & Bound Method

Step 2. Select a pending node and solve the associated sub-problem



Branch & Bound Method

Step 2. Select a pending node and solve the associated sub-problem

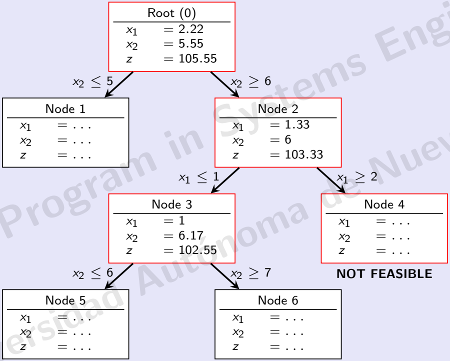


Node 3	
x_1	$= 1$
x_2	$= 6.17$
z	$= 102.55$

$$
\left\{
\begin{array}{llllllll}
Max & 10x_1 & + & 15x_2 & = & z & \\
s.t. & 8x_1 & + & 4x_2 & \leq & 40 & (1) \\
 & 15x_1 & + & 30x_2 & \leq & 200 & (2) \\
 & & & x_2 & \geq & 6 & (N2) \\
 & x_1 & & & \leq & 1 & (N3) \\
 & \multicolumn{5}{l}{x_1 \geq 0, \;\; x_2 \geq 0} & \\
\end{array}
\right.
$$

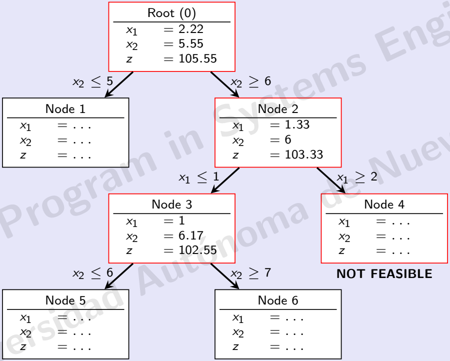
Branch & Bound Method

Step 3. Branch on a fractional variable if it exists



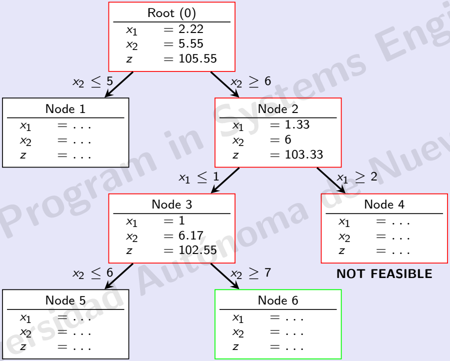
Branch & Bound Method

Step 4. Go to Step 2 if there are still some pending nodes



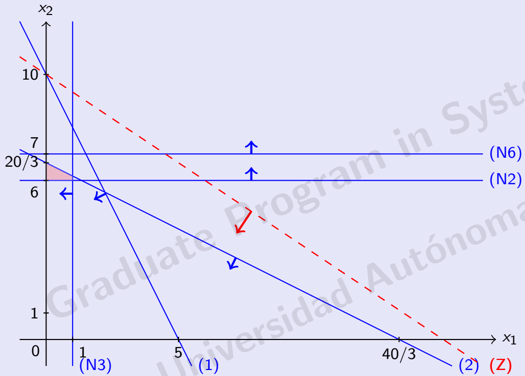
Branch & Bound Method

Step 2. Select a pending node and solve the associated sub-problem



Branch & Bound Method

Step 2. Select a pending node and solve the associated sub-problem

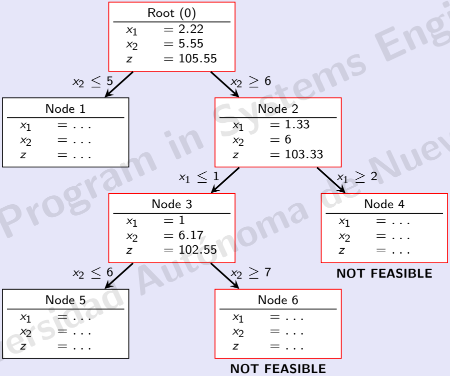


Node 6	
x_1	$= \dots$
x_2	$= \dots$
z	$= \dots$

$$
\begin{cases}
\begin{array}{lllllll}
Max & 10x_1 & + & 15x_2 & = & z & \\
s.t. & 8x_1 & + & 4x_2 & \leq & 40 & (1) \\
 & 15x_1 & + & 30x_2 & \leq & 200 & (2) \\
 & & & x_2 & \geq & 6 & (N2) \\
 & x_1 & & & \leq & 1 & (N3) \\
 & & & x_2 & \geq & 7 & (N6) \\
 & x_1 \geq 0, & & x_2 \geq 0 & & &
\end{array}
\end{cases}
$$

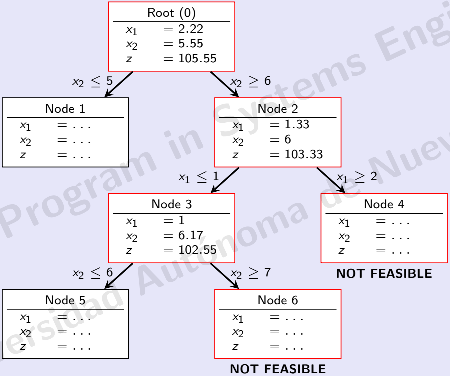
Branch & Bound Method

Step 3. Branch on a fractional variable if it exists



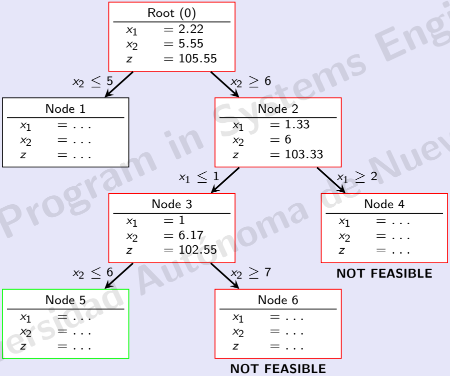
Branch & Bound Method

Step 4. Go to Step 2 if there are still some pending nodes



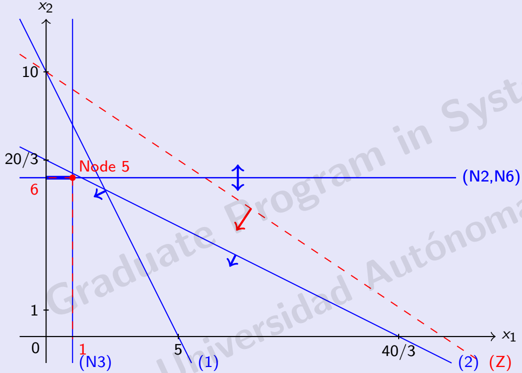
Branch & Bound Method

Step 2. Select a pending node and solve the associated sub-problem



Branch & Bound Method

Step 2. Select a pending node and solve the associated sub-problem

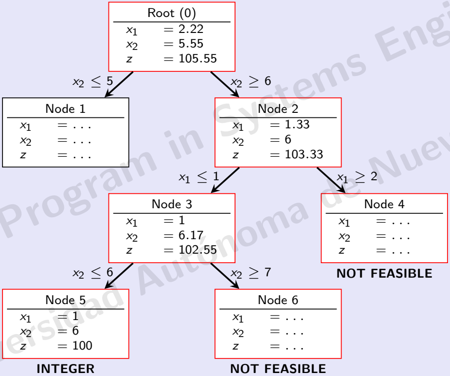


Node 5	
x_1	$= 1$
x_2	$= 6$
z	$= 100$

$$
\begin{cases}
\begin{array}{llclcll}
Max & 10x_1 & + & 15x_2 & = & z & \\
s.t. & 8x_1 & + & 4x_2 & \leq & 40 & (1)\\
 & 15x_1 & + & 30x_2 & \leq & 200 & (2)\\
 & & & x_2 & \geq & 6 & (N2)\\
 & x_1 & & & \leq & 1 & (N3)\\
 & & & x_2 & \leq & 6 & (N5)\\
 & x_1 \geq 0, & x_2 \geq 0 & & & &
\end{array}
\end{cases}
$$

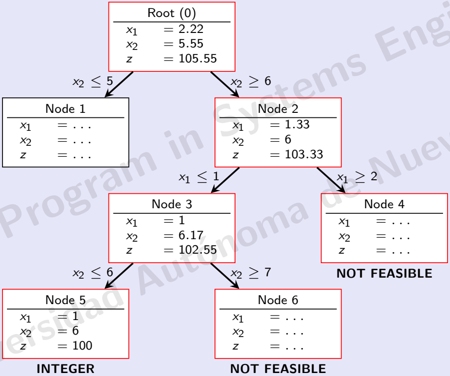
Branch & Bound Method

Step 3. Branch on a fractional variable if it exists



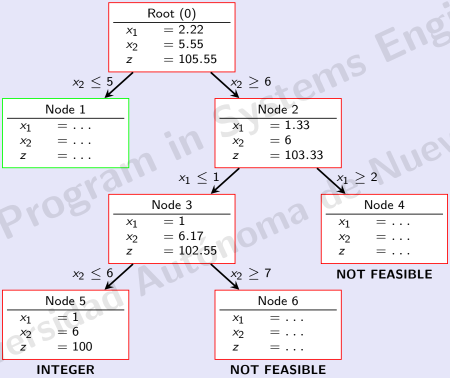
Branch & Bound Method

Step 4. Go to Step 2 if there are still some pending nodes



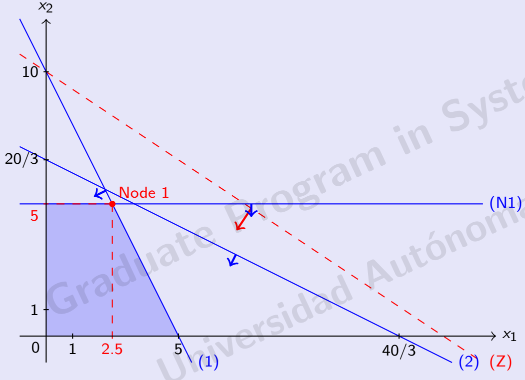
Branch & Bound Method

Step 2. Select a pending node and solve the associated sub-problem



Branch & Bound Method

Step 2. Select a pending node and solve the associated sub-problem

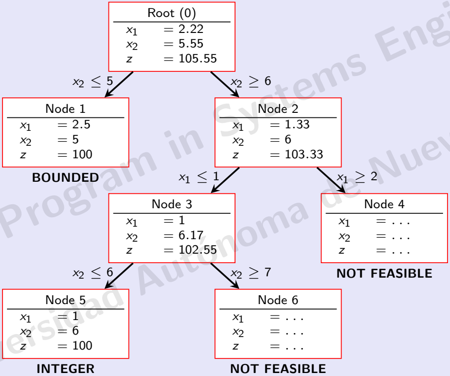


Node 1		
x_1	$=$	2.5
x_2	$=$	5
z	$=$	100

$$\begin{cases} Max & 10x_1 & + & 15x_2 & = & z & \\ s.t. & 8x_1 & + & 4x_2 & \leq & 40 & (1) \\ & 15x_1 & + & 30x_2 & \leq & 200 & (2) \\ & & & x_2 & \leq & 5 & (N1) \\ & x_1 \geq 0, & x_2 \geq 0 & & & & \end{cases}$$

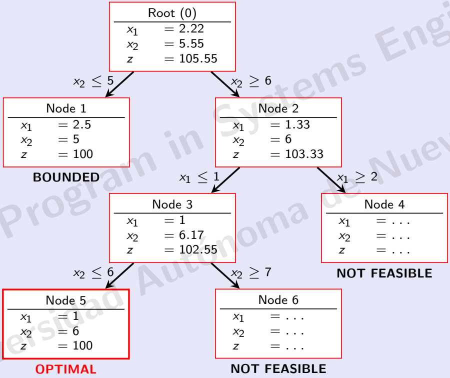
Branch & Bound Method

Step 3. Branch on a fractional variable if it exists



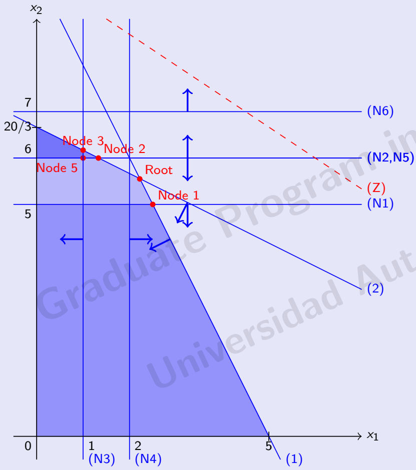
Branch & Bound Method

Step 4. Go to Step 2 if there are still some pending nodes

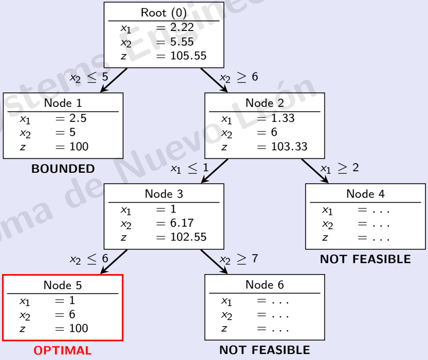


Branch & Bound Method

Sub-Problems' Solution

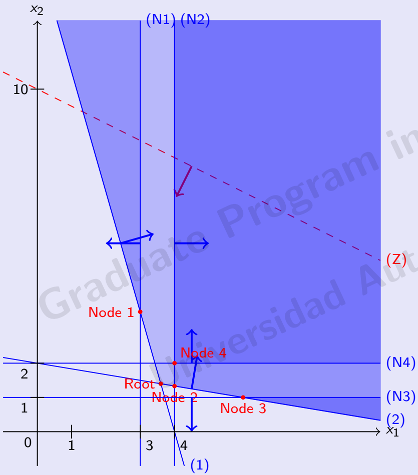


Exploration Tree

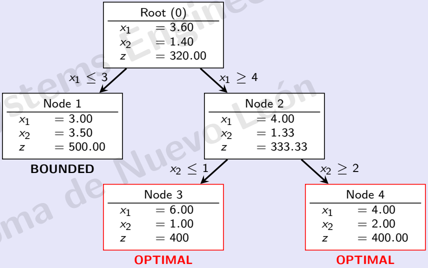


Dorian's problem

Sub-Problems' Solution



Exploration Tree



$$
\left\{
\begin{array}{llcllll}
Min & 50x_1 & + & 100x_2 & & & \\
s.t. & 7x_1 & + & 2x_2 & \geq & 28 & (1) \\
 & 2x_1 & + & 12x_2 & \geq & 24 & (2) \\
 & \multicolumn{6}{l}{x_1 \geq 0,\ x_2 \geq 0} \\
 & \multicolumn{6}{l}{x_1 \text{ and } x_2 \text{ integer}}
\end{array}
\right.
$$