Mixed-Integer Programming

Prof. Cesar de Prada
ISA –UVA
prada@autom.uva.es

Outline

- Hybrid problems
- Types of mixed integer problems
- Branch and Bound algorithm
- Examples
- Software

Hybrid problems

- Many decision problems, besides variables that can be represented by real numbers, involve other decisions of a discrete nature that can be represented naturally by integer or binary variables.
- On other occasions, the formulation of the problem involves not only quantitative models but rules or conditions that are better described by logical expressions.
- The optimization problems that deal with these types of hybrid systems that involve real and integer variables are called mixed integer programming (MIP).
- If all the decision variables are integers, then the problem is classified as one of integer optimization

Example: Gang of burglars

Several burglars at work are in a store where there are N distinct objects. Each object j has a weight p_j and a value v_j. They have a van that can carry a maximum load P. Which objects should be selected by the burglars in order to maximize the benefit of the robbery?

The decision to be made on each object is to select it or not. A binary variables y_j can be used for this purpose

$$\max_{\mathbf{y}} \sum_{j=1}^{N} y_j v_j \quad \text{under} \quad \sum_{j=1}^{N} y_j p_j \le P$$

$$y_j = \begin{cases} 0 & \text{object j has not been selected} \\ 1 & \text{object j has been selected} \end{cases}$$

ILP problem
integer linear
programming

Modelling logic with binary variables

Select one alternative and only one

$$\sum_{i=1}^{N} y_i = 1$$

Select no more that one alternative

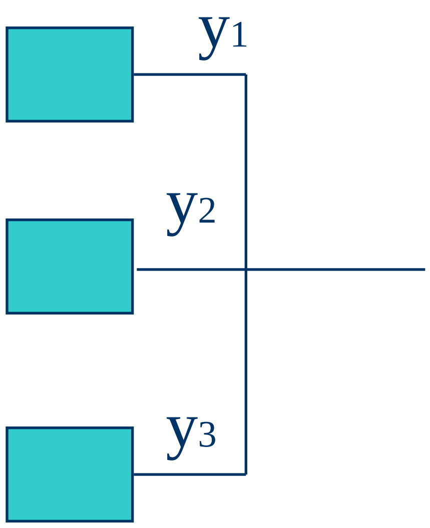
$$\sum_{i=1}^{N} y_i \leq 1$$

Select at least one alternative

$$\sum_{i=1}^{N} y_i \geq 1$$

Select alternative j if alternative i has been selected

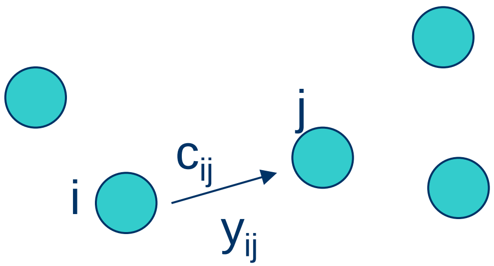
$$y_i \leq y_j$$



Example: Salesman problem

A salesman must travel from his town to N others going back home without staying twice in any of them. He knows the distance between any two towns. Which is the best route in order to travel through a minimum distance?

The decision to be made is to travel from town i to town j or not. We can associate a binary variable y_{ij} to this decision for each couple of towns and denote as c_{ij} the distance between them



$$\min_{\mathbf{y}} \sum_{i=1}^{N}\sum_{j=1}^{N} c_{ij} y_{ij} \qquad y_{ij} = \begin{cases} 0 & \text{the salesman does not travel from town i to j} \\ 1 & \text{the salesman travels from town i to j} \end{cases} \qquad y_{ii} = 0$$

$$\sum_{i=1}^{N} y_{ij} = 1 \qquad j = 1,\dots,N \qquad \text{he must arrive once and only once to town j}$$

$$\sum_{j=1}^{N} y_{ij} = 1 \qquad i = 1,\dots,N \qquad \text{he must depart once and only once from town i}$$

Assigning tasks

In a workshop n people able to develop n tasks with different performances are working. The time required by each person to develop a given task is known. How the different tasks should be assigned to each person in order to minimize the time required to perform the n tasks?

Variables

i people

j tasks

t_{ij} time required by person i to finish task j

y_{ij} binary variable, is 1 if the person i is assigned task j

Assigning tasks

$$\min_{\mathbf{x}} \sum_{i=1}^{n} \sum_{j=1}^{n} t_{ij} y_{ij}$$

Total time required to complete the n tasks

sujeto a

$$\sum_{i=1}^{n} y_{ij} = 1 \qquad j = 1, \ldots, n$$

Each person must have a task assigned and only one

$$\sum_{j=1}^{n} y_{ij} = 1 \qquad i = 1, \ldots, n$$

Each task have to be assigned to one person and only to one

y_{ij} binary

Types of mixed-integer problems

$$\begin{cases} \min\limits_{\mathbf{y}} \mathbf{c}'\mathbf{y} \\ \mathbf{A}\mathbf{y} = \mathbf{b} \\ \mathbf{y} \in Z \end{cases}$$

ILP Integer Linear Programming

$$\begin{cases} \min\limits_{\mathbf{x},\mathbf{y}} \mathbf{c}'\mathbf{x} + \mathbf{d}'\mathbf{y} \\ \mathbf{A}\mathbf{x} = \mathbf{b} \\ \mathbf{E}\mathbf{y} = \mathbf{e} \\ \mathbf{0} \leq \mathbf{x} \in R^n, \mathbf{y} \in Z \end{cases}$$

MILP Mixed-Integer Linear Programming

$$\begin{cases} \min\limits_{\mathbf{x},\mathbf{y}} J(\mathbf{x},\mathbf{y}) \\ \mathbf{h}(\mathbf{x},\mathbf{y}) = \mathbf{0} \\ \mathbf{g}(\mathbf{x},\mathbf{y}) \leq \mathbf{0} \\ \mathbf{x} \in R^n, \mathbf{y} \in Z \end{cases}$$

MINLP Mixed-Integer Non-Linear Programming

Slack variables can be used to transform problems with equalities into inequalities and vice versa, or min problems into max ones

Solution methods

- One possible approach consists of relaxing the integer variables into real ones, solving the corresponding NLP problem and then approximating the solution to the closest integer, usually leads to wrong solutions, except perhaps when a high number of values are admissible for each integer variable.
- Another method is to enumerate all possible combinations of integer variables, solving each of the associated NLP problems that results when the integer variables are given a fixed value and then choose the combination that provides a better cost function. Nevertheless, this is not a practical approach as the number of combinations grows exponentially with the number of integer variables.
- The most popular solution method is based on an intelligent selection of the integer combinations known as Branch and Bound (B&B)
- There are many other approaches, most of them using a succession of two phases; the so called Primary and Master ones. These phases provide upper and lower bounds that narrow the gap progressively. Examples: Outer Approximation (OA), Generalised Benders Decomposition (GBD)

Branch and Bound (B&B)

This method is based on an intelligent search of the optimum combining the choice of integer combinations with relaxations and the generation of lower and upper bounds of the cost function that leads to the solution.

It uses three main ideas:

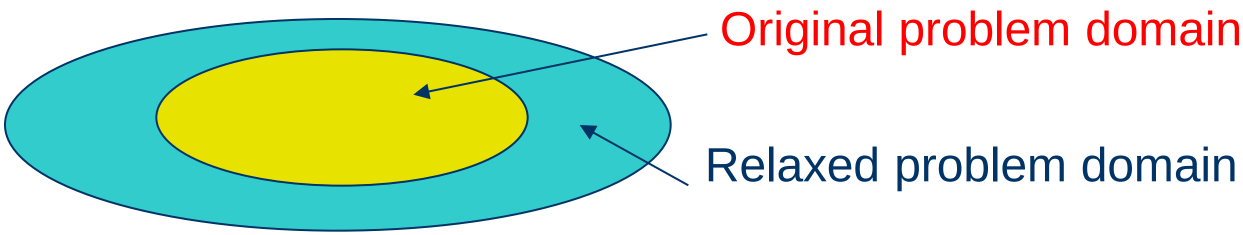
✓ **Relaxation**, that covert integer into real variables and allows to compute bounds on the cost function.

✓ **Branching**, that generate alternatives of combinations of integer variables in the decision tree.

✓ **Fathoming**, examining the bounds allows to eliminate groups of integer combinations improving the search in this way.

Relaxation

A **relaxation** of an integer variable in a MILP or MINLP problem consists of allowing it to take any real value between its maximum and minimum range. For instance, a binary variable could take values within the interval $0 \le y_j \le 1$. So, in the relaxed problem, all variables, **x** and **y**, are real ones and the corresponding problem is LP or NLP.



Consequently, as the search space is widening, the solution of the relaxed problem is a lower bound (upper bound if the problem is a maximization one) of the original MILP or MINLP. The relaxation is made with the purpose of obtaining such a bound.

Branch and Bound (B&B) algorithm

Example ILP (Himmelblau)

Max $J = 86\, y_1 + 4\, y_2 + 40\, y_3$

under

$774\, y_1 + 76\, y_2 + 42\, y_3 \le 875$

$67\, y_1 + 27\, y_2 + 53\, y_3 \le 875$

$y1, y2, y3 \in 0,1$

1 Relaxation

1

$0 \le y_1 \le 1$

$0 \le y_2 \le 1$

$0 \le y_3 \le 1$

$y^*=(1, 0.776, 1)$

$J_r^*=129.1$

LP

$y_2= 0$ — 2 Branching — $y_2= 1$

The relaxed problem is a LP one and its solution provides an upper bound Jr* of J*: $J^* \le 129.1$

Then, the two possible integer options for y_2, (the only remaining real number in the solution) are examined

Branch and Bound (B&B) algorithm

Max $J = 86\, y_1 + 4\, y_2 + 40\, y_3$

under

$774\, y_1 + 76\, y_2 + 42\, y_3 \le 875$

$67\, y_1 + 27\, y_2 + 53\, y_3 \le 875$

$y1, y2, y3 \in 0,1$

Node nº

LP

1

$0 \le y_1 \le 1$

$0 \le y_2 \le 1$

$0 \le y_3 \le 1$

$y^*=(1, 0.776, 1)$

$J_r^*=129.1$

1 Relaxation

2 Branching

$y_2 = 0$

$y_2 = 1$

3 Relaxation

LP

2

$0 \le y_1 \le 1$

$y_2 = 0$

$0 \le y_3 \le 1$

$y^*=(1, 0, 1)$

$J_r^*=126.0$

4 Fathoming

Current best feasible solution (incumbent):

Lower bound of J*

No more branching is possible at node 2. The B&B finish if the gap between the upper and lower bounds is less then a certain desired accuracy

$$\frac{\left|\text{Cota}_{\sup} - \text{cota}_{\inf}\right|}{1 + \left|\text{cota}_{\inf}\right|} \le tol$$

B&B

1 Relaxation

LP

Node 1: $0 \le y_1 \le 1$, $0 \le y_2 \le 1$, $0 \le y_3 \le 1$, $y^*=(1, 0.776, 1)$, $J_r^*=129.1$

129.1
J*
-∞

Upper bound for all solutions bellow

2 Branching

$y_2= 0$ → **Node 2**: $0 \le y_1 \le 1$, $y_2 = 0$, $0 \le y_3 \le 1$, $y^*=(1, 0, 1)$, $J_r^*=126.0$

129.1
J*
126.0

$y_2= 1$ → **Node 3**: $0 \le y_1 \le 1$, $y_2 = 1$, $0 \le y_3 \le 1$, $y^*=(0.978,1, 1)$, $J_r^*=128.11$

128.11
J*
126.0

Relaxations

Fathoming

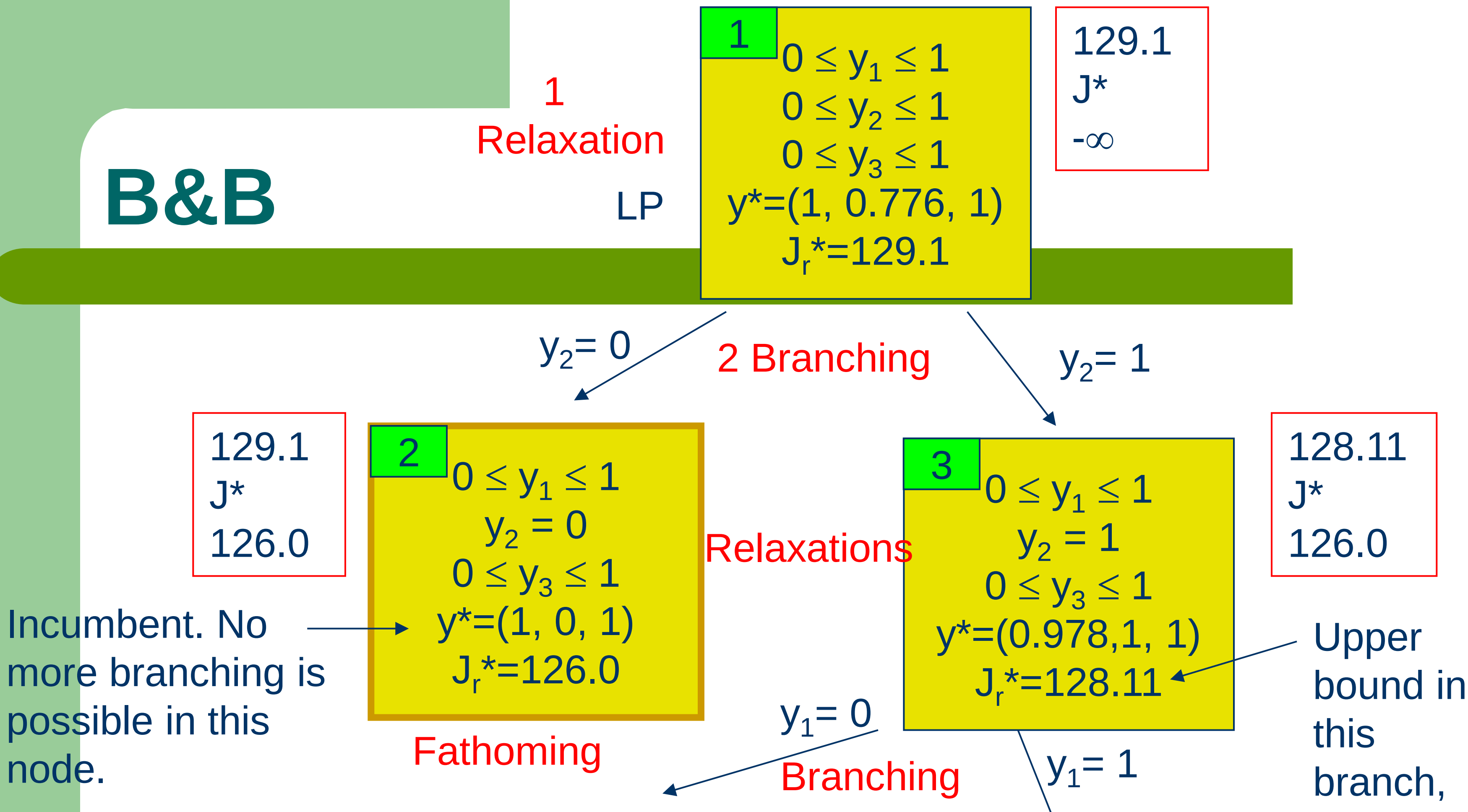
Branching: $y_1= 0$, $y_1= 1$

Candidate solution. Incumbent. No more branching is possible in this node. The J of the candidate is a lower bound for all branches.

Upper bound in this branch, and, as 128.1 < 129.1, it is also the new upper bound of the problem

If the gap in node 3 is higher than the desired accuracy, new branching should be made. Otherwise, the B&B finish and the incumbent is the optimum

B&B



Each feasible integer solution provides a lower bound of the problem

Each branching provides new upper bounds in the branch

The values of the bounds can be used to fathom branches without the need of computing its values

B&B

1 Relaxation

LP

Node 1
$0 \le y_1 \le 1$
$0 \le y_2 \le 1$
$0 \le y_3 \le 1$
$y^*=(1, 0.776, 1)$
$J_r^*=129.1$

129.1
J*
-∞

2 Branching

$y_2 = 0$ / $y_2 = 1$

Node 2
$0 \le y_1 \le 1$
$y_2 = 0$
$0 \le y_3 \le 1$
$y^*=(1, 0, 1)$
$J_r^*=126.0$

129.1
J*
126.0

Incumbent

Relaxations

Node 3
$0 \le y_1 \le 1$
$y_2 = 1$
$0 \le y_3 \le 1$
$y^*=(0.978,1, 1)$
$J_r^*=128.11$

128.11
J*
126.0

Branching

$y_1 = 0$ / $y_1 = 1$

Node 4
$y_1 =0$
$y_2 = 1$
$0 \le y_3 \le 1$
$y^*=(0,1, 1)$
$J_r^*=44.0$

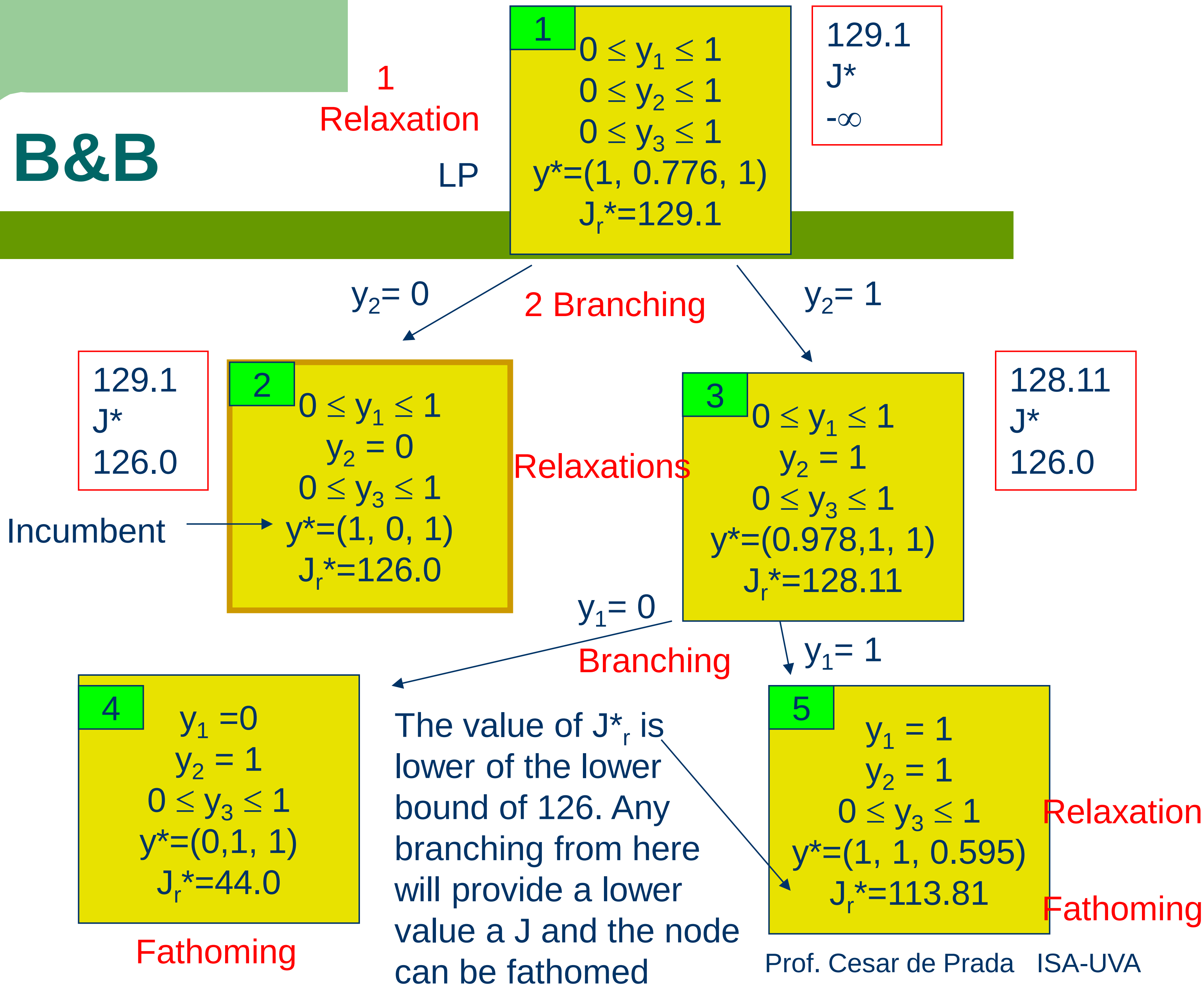
Relaxation

Fathoming

New integer feasible solution, but as the associated cost is lower than the lower bound, it can be discarded and the node fathomed

No more branching is allowed in this node

B&B


1
Relaxation
LP
1
0 ≤ y1 ≤ 1
0 ≤ y2 ≤ 1
0 ≤ y3 ≤ 1
y*=(1, 0.776, 1)
Jr*=129.1
129.1
J*
-∞
y2= 0
2 Branching
y2= 1
129.1
J*
126.0
2
0 ≤ y1 ≤ 1
y2 = 0
0 ≤ y3 ≤ 1
y*=(1, 0, 1)
Jr*=126.0
Incumbent
Relaxations
3
0 ≤ y1 ≤ 1
y2 = 1
0 ≤ y3 ≤ 1
y*=(0.978,1, 1)
Jr*=128.11
128.11
J*
126.0
y1= 0
Branching
y1= 1
4
y1 =0
y2 = 1
0 ≤ y3 ≤ 1
y*=(0,1, 1)
Jr*=44.0
Fathoming
The value of J*r is lower of the lower bound of 126. Any branching from here will provide a lower value a J and the node can be fathomed
5
y1 = 1
y2 = 1
0 ≤ y3 ≤ 1
y*=(1, 1, 0.595)
Jr*=113.81
Relaxation
Fathoming

B&B

1 Relaxation

LP

Node 1

$0 \le y_1 \le 1$
$0 \le y_2 \le 1$
$0 \le y_3 \le 1$
$y^*=(1, 0.776, 1)$
$J_r^*=129.1$

129.1
J*
-∞

$y_2= 0$ 2 Branching $y_2= 1$

Node 2

129.1
J*
126.0

$0 \le y_1 \le 1$
$y_2 = 0$
$0 \le y_3 \le 1$
$y^*=(1, 0, 1)$
$J_r^*=126.0$

Incumbent → $y^*=(1, 0, 1)$

Node 3

$0 \le y_1 \le 1$
$y_2 = 1$
$0 \le y_3 \le 1$
$y^*=(0.978,1, 1)$
$J_r^*=128.11$

128.11
J*
126.0

$y_1= 0$ Branching $y_1= 1$

Node 4

$y_1 =0$
$y_2 = 1$
$0 \le y_3 \le 1$
$y^*=(0,1, 1)$
$J_r^*=44.0$

fathoming

Node 5

$y_1 = 1$
$y_2 = 1$
$0 \le y_3 \le 1$
$y^*=(1, 1, 0.595)$
$J_r^*=113.81$

fathoming

As no more branching is possible, the incumbent of node 2 is the optimal solution

Integer and binary variables

Any integer variable z taking values between 0 and n, can be substituted by a set of binary variables, that is variables that only take 0 or 1 values:

$$z = y_1 + 2\, y_2 + 3\, y_3 + \ldots. + n\, y_n$$

$$1 \geq y_1 + y_2 + y_3 + \ldots. + y_n$$

$$y = \{0 , 1\}$$

Also $z = 2\, y_1 + 2^2\, y_2 + \ldots.. + 2^k\, y_k$ does the same with less integer variables

This can represent integers up to $2^{k+1}-1$

Then, mixed integer optimization problem can always be formulated in terms of binary variables

Example: Paint factory

A paint manufacturing facility has three production units with capacities given in the table bellow. The costs associated to the start up of the unit and to producing one Kg of paint are also given there. One production unit can be started either in the morning or in the afternoon, but, once started, must remain working at least for half a day (one period: morning or afternoon)

Unit	Start up cost €	Cost per Kg of paint produced €	Capacity, Kg/period
1	2800	5	1900
2	2000	3	1700
3	1900	8	2900

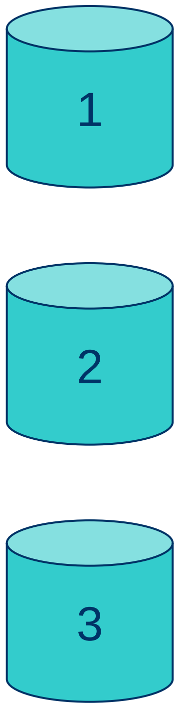
Paint factory

If one unit was started in the morning and continues operating in the afternoon, obviously, only generates starting up costs in the morning. All units are switched off at night, and the planning of the day operation is made daily in the morning according to the existing demand.

Assume that a certain day the factory must deliver 2500 kg of paint in the morning and 3500 kg in the afternoon. Which units should be used and when in order to reduce the cost as much as possible?

How much would change the cost if the demand in the afternoon were of 3600Kg?

Paint factory



Variables:

i unit number (1, 2, 3)

j working period: 1 morning 2 afternoon

y_{ij} binary variable: equal to 1 if the unit i works in the period j

c_i start up costs of unit i

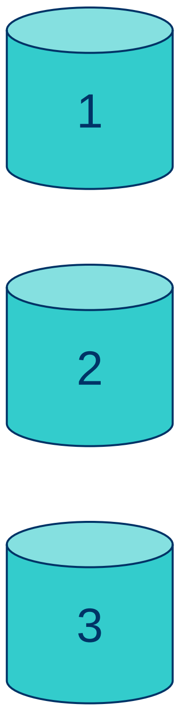
p_i production costs of a Kg of paint in unit i

w_i production of unit i in a period (= capacity)

D_j paint demand in the period j

z_i auxiliary binary variable , 1 if y_{i1} or y_{i2} are 1

Paint factory



$$\min_{y_{ij}, z_i} \sum_{i=1}^{3} c_i z_i + p_i w_i (y_{i1} + y_{i2})$$

Total cost per day

$$\sum_{i=1}^{3} w_i y_{ij} \geq D_j \qquad j = 1,2$$

$$z_i \geq y_{ij} \qquad i = 1,2,3 \quad j = 1,2$$

Excel

Variable z_i is 1 if unit i has been started up in the morning or in the afternoon

Other possible logic constraint: If we assume that in the morning no more than a unit can work simultaneously:

$$\sum_{i=1}^{3} y_{i1} \leq 1$$

GAMS

```
sets    i   units / u1, u2, u3 /
        j  periods     /  m, t /

parameters costea(i)  starting up cost of a unit
                             / u1=2800, u2=2000, u3=1900 /
           costeKg(i) cost per Kg per period / u1=5 , u2=3, u3=8 /
           capacidad(i) capacity /u1=1900, u2=1700, u3=2900/
           demanda(j) demand per period / m= 2500, t = 3500/;

variables    y(i,j)    unit I works in period j
             z(i)      unit I start up that day
             coste     total cost per day
binary variables y, z;
```

GAMS

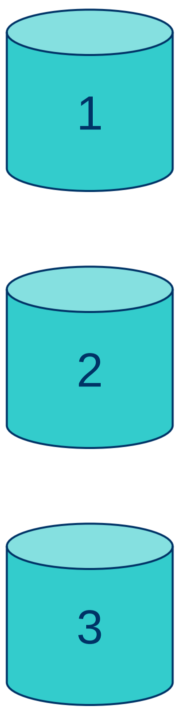
```
equations produccion(j) production per period
          restriccion(i,j)   constraints in z
          costetotal    total cost;

     produccion(j)..  sum(i, y(i,j)*capacidad(i)) =g= demanda(j);
     restriccion(i,j).. z(i) =g= y(i,j);
     costetotal..     coste =e= sum(i,
              costea(i)*z(i)+costeKg(i)*capacidad(i)*sum(j,y(i,j)));

model  pinturas production planning / all /;
solve   pinturas  minimizing coste using mip;
display  coste.l
```

Paint factory

Now production w_i of each unit is no longer equal to capacity C_i and we have to distinguish between morning and afternoon w_{ij}



$$\min_{y_{ij}, z_i} \sum_{i=1, j=1}^{3,2} c_i z_i + p_i w_{ij}$$

Total cost per day

$$C_i y_{ij} \geq w_{ij} \geq 0 \qquad i = 1,2,3 \quad j = 1,2$$

$$\sum_{i=1}^{3} w_{ij} y_{ij} \geq D_j \qquad j = 1,2$$

$$z_i \geq y_{ij} \qquad i = 1,2,3 \quad j = 1,2$$

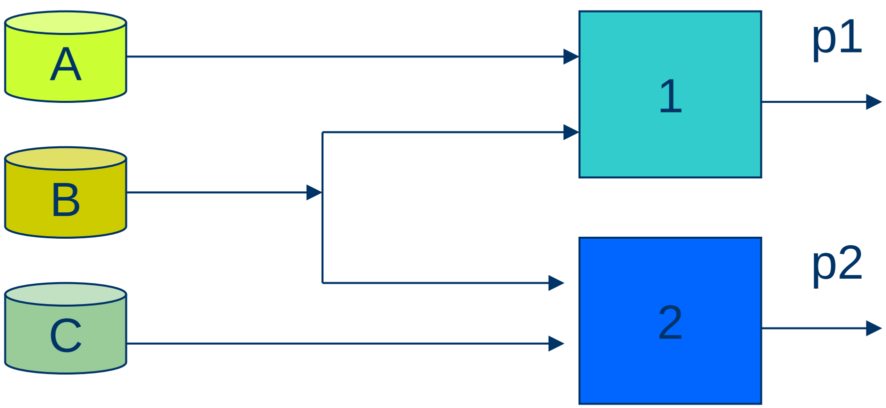
Variable z_i is 1 if unit i has been started up in the morning or in the afternoon

Blending with discrete batch sizes

Mixing unit	Capacity kg/day
1	8000
2	10000

Raw materials required to manufacture one Kg of	A Kg	B Kg	C Kg	Profit € / Kg
Product p1	0.4	0.6	0	0.16
Product p2	0	0.3	0.7	0.2
Availability	∞	6000	∞	

Each unit works with batches of 2000Kg



Which amounts of p1 and p2 should be manufactured in order to maximize profits?

Prof. Cesar de Prada ISA-UVA

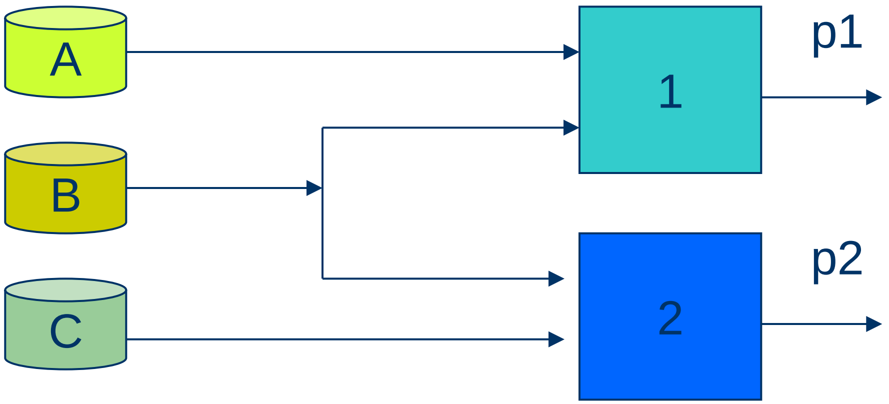
Blending with discrete batch sizes

Variables:

x_1 Kg of p1 manufactured per day

x_2 Kg of p2 manufactured per day

$$\max \; 0.16x_1 + 0.2x_2$$
$$0.6x_1 + 0.3x_2 \leq 6000$$
$$x_i = 2000y_i \quad i = 1, 2$$
$$0 \leq y_1 \leq 4 \qquad 0 \leq y_2 \leq 5$$
$$y_i \quad \text{integer}$$



x_i must be a multiple of 2000 Kg, the batch size

Branch and Bound (B&B) algorithm

$$\max \; 0.16x_1 + 0.2x_2$$
$$0.6x_1 + 0.3x_2 \le 6000$$
$$x_i = 2000y_i \quad i = 1, 2$$
$$0 \le y_1 \le 4 \qquad 0 \le y_2 \le 5$$
$$y_i \quad \text{integer}$$

1
Relaxation

$0 \le y_1 \le 4$
$0 \le y_2 \le 5$
$y^* = (2.5, 5)$
$J_r^* = 2800$

LP

$y_1 \le 2$

2 Branching

$y_1 \ge 3$

The relaxed problem is an LP one and its solution provides un upper bound Jr* of J*:
$J^* \le 2800$

Next, the two possible alternatives for y_1, the only real variable of the relaxed solution, will be examined

Branch and Bound (B&B) algorithm

$$\max \; 0.16x_1 + 0.2x_2$$
$$0.6x_1 + 0.3x_2 \le 6000$$
$$x_i = 2000y_i \quad i = 1, 2$$
$$0 \le y_1 \le 4 \qquad 0 \le y_1 \le 5$$
$$y_i \quad \text{integer}$$

Node nº

LP

1

$0 \le y_1 \le 4$
$0 \le y_2 \le 5$
$y^*=(2.5, 5)\ x_i$
$J_r^*=2800$

1 Relaxation

$y_1 \le 2$

2 Branching

$y_1 \ge 3$

3 Relaxation

LP

2

$0 \le y_1 \le 2$
$0 \le y_2 \le 5$
$y^*=(2,5)\ x_i$
$J_r^*=2640$

Current best feasible solution (incumbent):

Lower bound of J*

4 Fathoming

No more branching is possible at node 2. The B&B finish if the gap between the upper and lower bounds is less then a certain desired accuracy

B&B

1 Relaxation

LP

1

$0 \le y_1 \le 4$
$0 \le y_2 \le 5$
$y^*=(2.5, 5)$
$J_r^*=2800$

2800
J*
-∞

$y_1 \le 2$

2 Branching

$y_1 \ge 3$

2800
J*
2640

2

$0 \le y_1 \le 2$
$0 \le y_2 \le 5$
$y^*=(2, 5)\ x_i$
$J_r^*=2640$

Relaxations

3

$3 \le y_1 \le 4$
$0 \le y_2 \le 5$
$y^*=(3,4)$
$J_r^*=2560$

Another feasible solution, but with lower cost function than the incumbent

Incumbent

No more branching is made in this node as a feasible solution of the MILP is found

Fathoming

Fathoming

Hence, the solution is: $y^*=(2, 5)$, $x^*=(4000, 10000)$

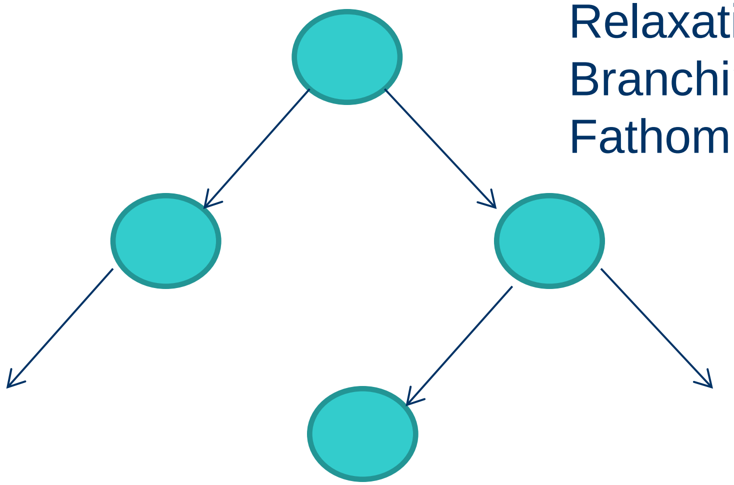
And the optimal profit 2640 €

Solving MINLP: Branch and bound

$$\min_{x,y} \quad J(x,y)$$
$$h(x,y) = 0$$
$$g(x,y) \le 0$$
$$x \in X, \quad y \in \{0,1\}$$

Relaxed to an NLP at each node

$$\min_{x,y} \quad J(x,y)$$
$$h(x,y) = 0$$
$$g(x,y) \le 0$$
$$x \in X, \quad 0 \le y_i \le 1$$

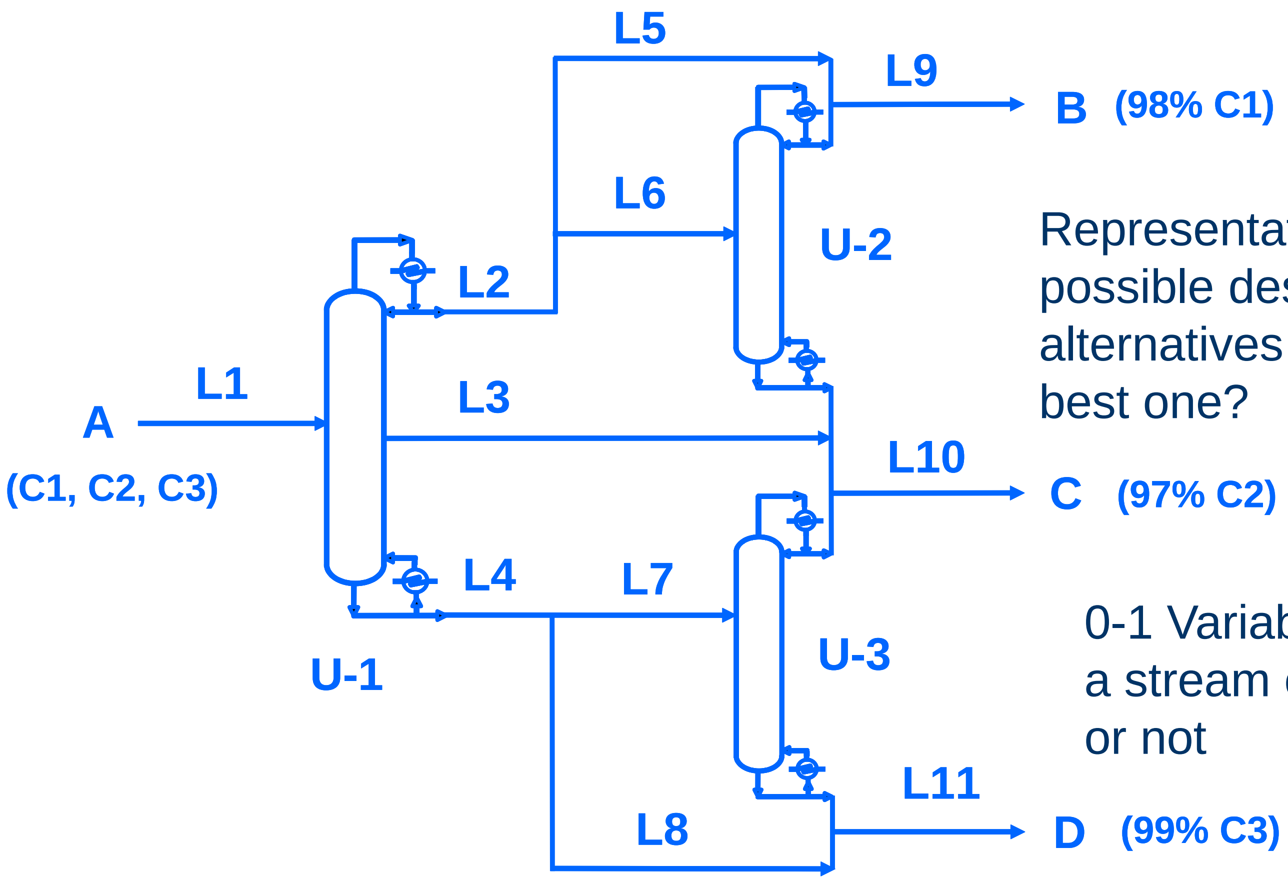


Relaxation
Branching
Fathoming

NLPs provides Lower bounds

Integer solutions in y provide Upper bounds

Super-structures



Representation of all possible design alternatives. Which is the best one?

0-1 Variables indicate if a stream or unit exists or not

Turning off continuous variables

One can force the continuous variable q to have a value 0 or a positive one, as a function of a logic condition represented by a binary variable y:

q continuous variable , e.g.. flow
L lower bound
U upper bound

Never use the product yq because this is a non-convex term

$$Ly \leq q \leq Uy$$

$$\text{if} \quad y = 0 \quad 0 \leq q \leq 0 \Rightarrow q = 0$$

$$\text{if} \quad y = 1 \quad L \leq q \leq U$$

Multiperiod

Activation of the operation of a unit i at time periods t = 1, 2, …T using the binary variable y_{it}. The unit i can exists or not (using the binary variable z_i),

$$\sum_{t=1}^{T} y_{it} \le T z_i$$

If $z_i = 0$ then all y_{it} are zero

but

$$y_{it} \le z_i \quad t = 1,2,\ldots T$$

Is an equivalent, and usually tighter, alternative

Turning constraints on/off

Activation and deactivation of constraints associated to a stream or process unit

$$\text{constraints} \quad h(x) = 0 \qquad g(x) \le 0$$

$$\text{slack variables} \quad s, v$$

$$h(x) + s - v = 0$$

$$s + v \le U_1(1 - y) \qquad U, \text{large number}$$

$$g(x) - U_2(1 - y) \le 0$$

$$s \ge 0, v \ge 0$$

$$\text{if} \quad y = 0 \quad \text{then} \quad h(x) \text{ and } g(x) \text{ are not constrained}$$

$$\text{if} \quad y = 1 \quad \text{then} \quad s = 0,\ v = 0,\ h(x) = 0,\ g(x) \le 0$$

Switching constraints

The first or second constraint is activated as a function of the value of a binary variable y

$$\text{Either} \quad g_1(x) \le 0, \quad \text{or} \quad g_2(x) \le 0$$

$$g_1(x) - U(1-y) \le 0$$

$$g_2(x) - Uy \le 0$$

$$\text{if} \quad y = 0 \text{ then} \quad g_1(x) \le U,\ g_2(x) \le 0$$

$$\text{if} \quad y = 1 \text{ then} \quad g_1(x) \le 0,\ g_2(x) \le U$$

U large upper limit Big M

Conditional constraints

The second constraint is activated as a function of the value of the first one

$$\text{If} \quad g_1(x) \le 0, \quad \text{then} \quad g_2(x) \le 0$$

$$y_1, y_2 \ \text{associated with} \ P_1 \Rightarrow P_2$$

$$y_1 \le y_2$$

$$-M_1 y_1 \le g_1(x) \le M_1(1 - y_1)$$

$$-M_2 y_2 \le g_2(x) \le M_2(1 - y_2)$$

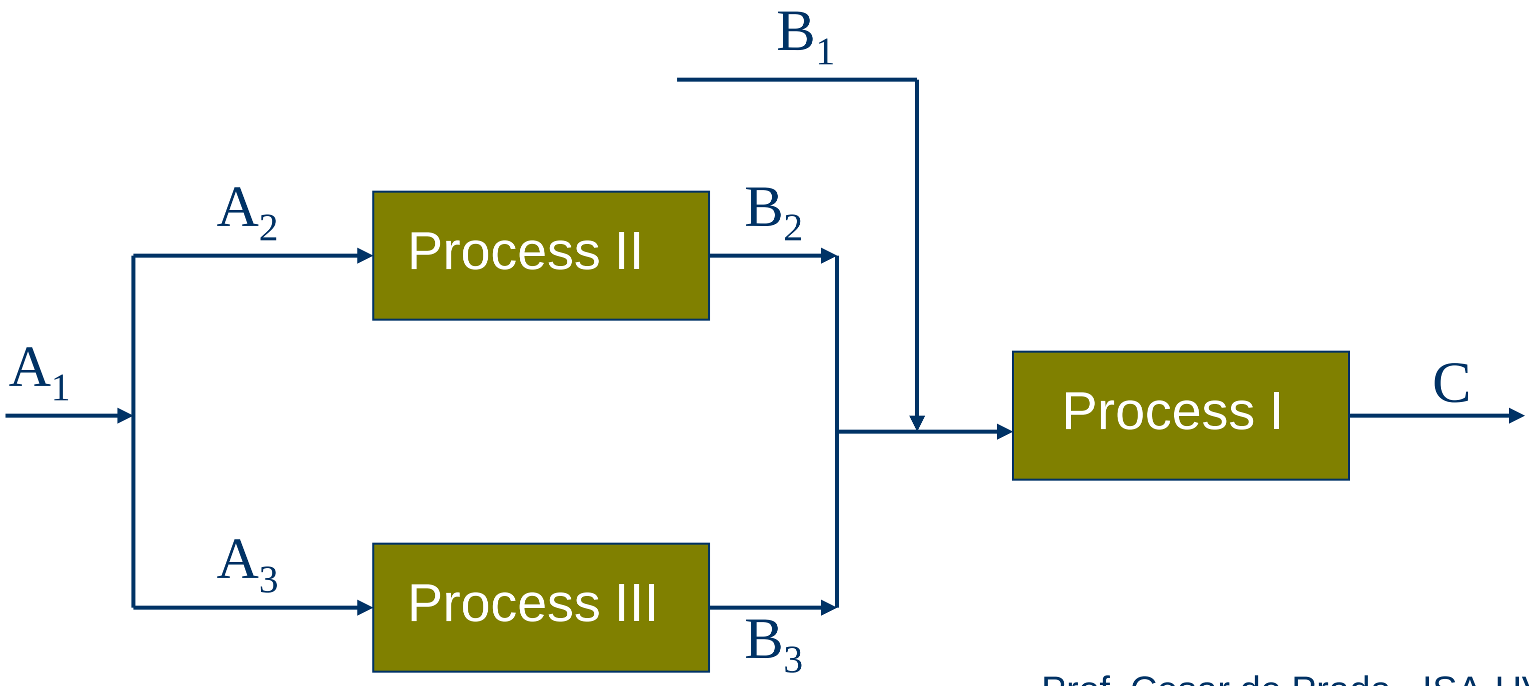
$$\text{if} \quad y_1 = 0 \ \text{then} \quad g_1(x) \le M_1, \ y_2, g_2(x) \ \text{any value}$$

$$\text{if} \quad y_1 = 1 \ \text{then} \quad g_1(x) \le 0, \ \ y_2 = 1, \quad g_2(x) \le 0$$

M large value

Process synthesis

A product C can be manufactured (Process I) from other B that can be purchased on the market or manufactured from product A in two different and excluding ways (Processes II and III). Represent the different alternatives and find the best way of producing it.



Process synthesis

Conversions:

Process I: C = 0.9B

Process II: B = ln(1 + A)

Process III: B = 1.2ln(1 + A)

Maximum capacity

Process I: 2 ton/h of C

Process II: 4 ton/h of B

Process III: 5 ton/h of B

Price

A: 1.800 €/ton

B: 7.000 €/ton

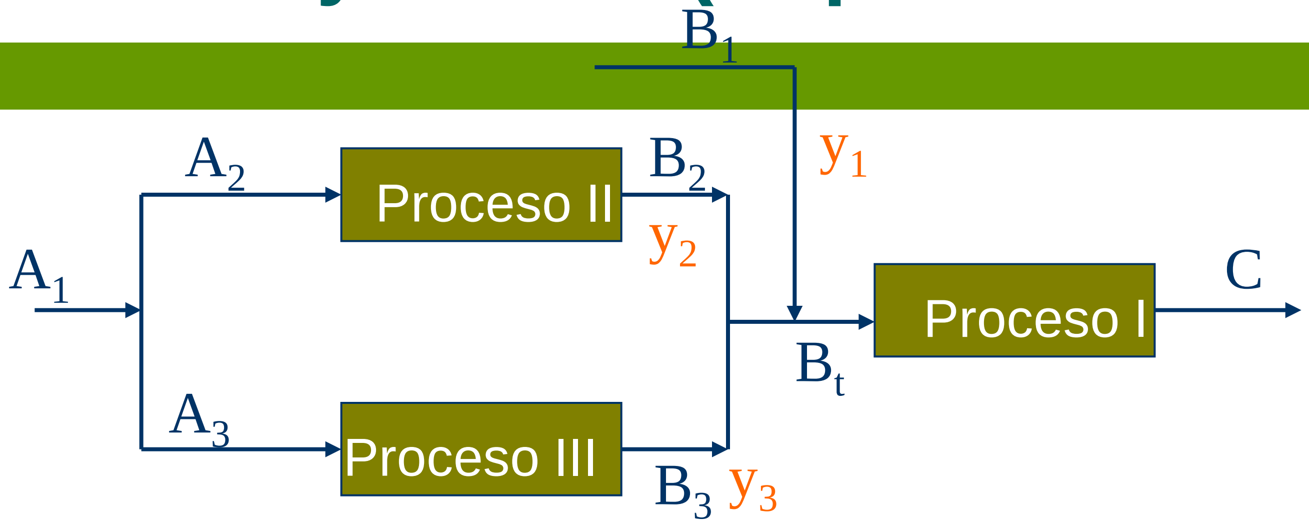
C: 13.000 €/ton

Costs

	Fixed (10^3 €/h)	Variable (10^3 €/ton)
Process I:	3.5	2
Process II:	1	1
Process III:	1.5	1.2

Market B maximum: 2 ton/h

Process synthesis (Superstructure)



max PR = 13C -1.8A$_2$ - 1.8 A$_3$ -7B$_1$ - 3.5 - 2C - 1.0y$_2$ - 1B$_2$ - 1.5y$_3$ - 1.2B$_3$

$$\max \; PR = 13C - 1.8A_2 - 1.8A_3 - 7B_1 - 3.5 - 2C - 1.0y_2 - 1B_2 - 1.5y_3 - 1.2B_3$$

s .a.

Balances

PI: $C - 0.9(B_1 + B_2 + B_3) = 0$

PII: $B_2 - \ln(1 + A_2) = 0$

PIII: $B_3 - 1.2\ln(1 + A_3) = 0$

$B_t = B_2 + B_3 + B_1$

$B_2 \le 4y_2$

$B_3 \le 5y_3$

$B_1 \le 2y_1$

$C \le 2$

Constraints

$C, A_2, A_3, B_1, B_2, B_3 >= 0$

$y_1, y_2, y_3 = 0, 1$

$y_2 + y_3 \le 1$

GAMS

```
Positive Variables
      a2 materia prima para el proceso 2
      a3 materia prima para el proceso 3
      b2 produccion de producto B en el proceso 2
      b3 produccion de producto B en el proceso 3
      b1 cantidad de producto B que se puede adquirir en el mercado
      bt cantidad de producto B que se consume en el proceso 1
      c1 capacidad de produccion del producto c en el proceso 1 ;
Binary Variables
      y1 existencia de compra exterior de B
      y2 existencia del proceso 2
      y3 existencia del proceso 3 ;
Variable
bene   beneficio total en millones de $ por ano ;
```

GAMS

- las restricciones inout2 e inout3 se han convexificado

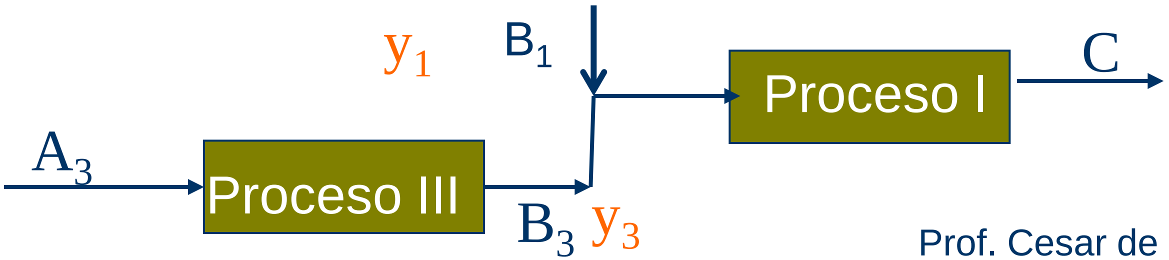
```
inout1.. c1 =e= 0.9*bt ;
inout2.. exp(b2) - 1 =e= a2 ;
inout3.. exp(b3/1.2) - 1 =e= a3 ;
mbalb.. bt =e= b2 + b3 + b1 ;
log1..   c1 = L= 2 ;
log2..   b2 = L= 4*y2 ;
log3..   b3 = L= 5*y3 ;
log4..   B1 = L= 2*y1
```

Rest.. $y_2 + y_3$ = L= 1

coste.. bene = E = 13*c1 -1.8*a2 - 1.8*a3 -7*b1 - 3.5 - 2*c1- y2 -b2 - 1.5*y3 - 1.2*b3;

GAMS

	LOWER	LEVEL	UPPER	MARGINAL
---- VAR a2	.	.	+INF	.
---- VAR a3	.	1.524	+INF	.
---- VAR b2	.	.	+INF	3.950
---- VAR b3	.	1.111	+INF	1.714
---- VAR b1	.	1.111	+INF	.
---- VAR bt	.	2.222	+INF	2.900
---- VAR c1	.	2.000	+INF	.
---- VAR y1	.	1.000	1.000	EPS
---- VAR y2	.	.	1.000	EPS
---- VAR y3	.	1.000	1.000	EPS
---- VAR bene	-INF	5.145	+INF	.



Modelling propositional logic expressions

P_i expression or logic variable with values false/true (0/1)

A logic proposition is a set of logic expressions linked by the logic operators:

$\wedge$ intersection $\vee$ union

$\overline{P}$ negation $\oplus$ exclusive or

The implication $P_1 \Rightarrow P_2$ is equivalent to $\overline{P_1} \vee P_2$

The logic expressions can be formulated as equations associating P (true / false) with y (1/0), and (no P) with 1-y

Logic operators

conjunction

AND	1	0
1	1	0
0	0	0

disjunction

OR	1	0
1	1	1
0	1	0

NOT	1	0
	0	1

EOR $\oplus$	1	0
1	0	1
0	1	0

q

p

$p \Rightarrow q$	1	0
1	1	0
0	1	1

Morgan Laws

$$\overline{(A+B)} = \overline{A}.\overline{B}$$

$$\overline{A.B} = \overline{A} + \overline{B}$$

Logic expressions / equations

$$P_1 \vee P_2 \vee P_3 \qquad y_1 + y_2 + y_3 \geq 1$$

$$P_1 \wedge P_2 \wedge P_3 \qquad y_1 \geq 1, \quad y_2 \geq 1, \quad y_3 \geq 1$$

$$P_1 \Rightarrow P_2 \qquad 1 - y_1 + y_2 \geq 1 \quad \text{ór} \quad y_1 \leq y_2$$

$$P_1 \text{ if and only if } P_2 \qquad y_1 = y_2$$

$$\text{one among} \quad P_1, P_2, P_3 \qquad y_1 + y_2 + y_3 = 1$$

$$P_1 \vee P_2 \Rightarrow P_3 \qquad y_1 \leq y_3 \quad y_2 \leq y_3$$

Using these equivalences, it is possible to convert any logic expression P to an associated set of equations in the binary variables *y*, if the logic expression is written in its normal conjunctive form

normal conjunctive form

$$Q_1 \wedge Q_2 \wedge \cdots \wedge Q_n$$

Where Q_i are logic expressions written as disjunctions

In order to transform any logic expression to this format:

1 Replace the implication by its equivalent expression

$$P_1 \Rightarrow P_2 \Leftrightarrow \overline{P_1} \vee P_2$$

2 Apply the Morgan's laws to move inside the negations

$$\overline{(P_1 \wedge P_2)} \Leftrightarrow \overline{P_1} \vee \overline{P_2} \qquad \overline{(P_1 \vee P_2)} \Leftrightarrow \overline{P_1} \wedge \overline{P_2}$$

3 Use the distributive property to arrive to normal conjunctive form

$$(P_1 \wedge P_2) \vee P_3 \Leftrightarrow (P_1 \vee P_3) \wedge (P_2 \vee P_3)$$

Example

$$(P_1 \wedge P_2) \vee P_3 \Rightarrow (P_4 \vee P_5)$$

Step 1

$$\overline{\left[(P_1 \wedge P_2) \vee P_3\right]} \vee (P_4 \vee P_5)$$

Step 2

$$\left[\overline{(P_1 \wedge P_2)} \wedge \overline{P_3}\right] \vee (P_4 \vee P_5) = \left[(\overline{P_1} \vee \overline{P_2}) \wedge \overline{P_3}\right] \vee (P_4 \vee P_5)$$

Step 3

$$\left[(\overline{P_1} \vee \overline{P_2}) \vee (P_4 \vee P_5)\right] \wedge \left[\overline{P_3} \vee (P_4 \vee P_5)\right]$$

$$\left[\overline{P_1} \vee \overline{P_2} \vee P_4 \vee P_5\right] \wedge \left[\overline{P_3} \vee P_4 \vee P_5\right]$$

Example

$$\left[\overline{P_1} \vee \overline{P_2} \vee P_4 \vee P_5\right] \wedge \left[\overline{P_3} \vee P_4 \vee P_5\right]$$

$$Q_1 \wedge Q_2$$

$$Q_1 = \overline{P_1} \vee \overline{P_2} \vee P_4 \vee P_5 \rightarrow \quad 1 - y_1 + 1 - y_2 + y_4 + y_5 \geq 1$$

$$Q_2 = \overline{P_3} \vee P_4 \vee P_5 \rightarrow \quad 1 - y_3 + y_4 + y_5 \geq 1$$

Then $Q_1 \wedge Q_2$ is equivalent to

$$y_1 + y_2 - y_4 - y_5 \leq 1 \qquad (P_1 \wedge P_2) \vee P_3 \Rightarrow (P_4 \vee P_5)$$

$$-y_3 + y_4 + y_5 \geq 0$$

Acetone production

(Raman & Grossmann, CACHE)

One wishes to select the best way to produce acetone CH_3COCH_3 from alcohol (CH_3CH_2OH) and methane (CH_4). There are different pathways to obtain acetone that are listed next, for which the appropriate catalyser is available as well as the intermediate inorganic compounds, with the exception of CrO_3 y O_3. Formulate the feasibility of the chemical reactions in mathematical form.

$$CH_3CO_2C_2H_5 \xrightarrow{NaOC_2H_5/C_2H_5OH} CH_3COCH_2CO_2C_2H_5$$

$$CH_3COCH_2CO_2C_2H_5 \xrightarrow{H_3O^+} CH_3COCH_3 + C_2H_5OH + CO_2$$

$$CH_3CN + CH_3MgI \xrightarrow{Et_2O} CH_3C(NMgI)CH_3 \xrightarrow{H_2O/HCl} CH_3COCH_3$$

Acetone production

(Raman & Grossmann, CACHE)

$$CH_3CHO + CH_3MgI \xrightarrow{Et_2O/H_3O^+} CH_3CHOHCH_3$$

$$CH_3CHOHCH_3 \xrightarrow{CrO_3/H_2SO_4} CH_3COCH_3$$

$$CH_2 = C(CH_3)_2 \xrightarrow{O_3/H_2O/H_2O_2} CH_3COCH_3 + HCO_2H$$

$$CH_3I \xrightarrow{Mg/Er_2O} CH_3MgI \xrightarrow{CH_3CO_2CH_3} (CH_3)_3COH$$

$$(CH_3)_3COH \rightarrow CH_2 = C(CH_3)_2$$

$$CH_4 + I_2 \rightarrow CH_3I + HI$$

$$CH_4 + Cl_2 \rightarrow CH_3Cl + HCl$$

$$CH_3CH_2OH + O_2 \xrightarrow{Cr_2O_3/Cu} CH_3CHO$$

$$CH_3Cl + NaCN \xrightarrow{H_2O} NaCl + CH_3CN$$

$$CH_3COOH + C_2H_5OH \rightarrow CH_3CO_2C_2H_5$$

$$CH_3CHO + O_2 \rightarrow CH_3COOH$$

Acetone production

(Raman & Grossmann, CACHE)

Formulation

Among all those chemical reactions, one must select those that allow synthesizing acetone from the given row materials and catalysers.

In order to formulate a mathematical optimization problem, we will express all chemical reactions as propositional logic expressions using the operators:

$$\vee \text{ (OR)}, \wedge \text{ (and)}, \Rightarrow \text{ (implication), y } \neg \text{(negation)}$$

I.

$$A + B \rightarrow C + D$$

$$A \xrightarrow{M} B$$

They can be formulated as

$$A \wedge B \Rightarrow C \wedge D$$

$$A \wedge M \Rightarrow B$$

Acetone production

(Raman & Grossmann, CACHE)

Next, we will formulate these logic propositions in normal conjunctive form following these steps:

1. Remove implications

$$\mathrm{A} \Rightarrow \mathrm{B} \quad \text{is equivalent to} \quad \neg \mathrm{A} \vee \mathrm{B}$$

2. Displace negations inside

$$\neg(A \wedge B) \Leftrightarrow (\neg A) \vee (\neg B)$$

$$\neg(A \vee B) \Leftrightarrow (\neg A) \wedge (\neg B)$$

3. Use the distributive property

$$(A \wedge B) \vee C \Leftrightarrow (A \vee C) \wedge (B \vee C)$$

Example

$$A \wedge B \Rightarrow C \wedge D$$

$$\neg(A \wedge B) \vee (C \wedge D)$$

$$\neg A \vee \neg B \vee (C \wedge D)$$

$$(\neg A \vee \neg B \vee C) \wedge (\neg A \vee \neg B \vee D)$$

Acetone production

(Raman & Grossmann, CACHE)

Then, each component of the conjunction can be converted into a equation by assigning a binary variable y to every of its variables or 1-y if it is affected by a negation, and using the translation of the operators

$$(\neg A \vee \neg B \vee C) \wedge (\neg A \vee \neg B \vee D)$$

$$1 - y_A + 1 - y_B + y_C \geq 1$$

$$1 - y_A + 1 - y_B + y_D \geq 1$$

$$y_A + y_B - y_C \leq 1$$

$$y_A + y_B - y_D \leq 1$$

Acetone production

(Raman & Grossmann, CACHE)

$$CH_3CHO + O_2 \rightarrow CH_3COOH$$

$$CH_3CHO \wedge O_2 \Rightarrow CH_3COOH$$
$$\neg(CH_3CHO \wedge O_2) \vee CH_3COOH$$
$$(\neg CH_3CHO \vee \neg O_2) \vee CH_3COOH$$
$$(\neg CH_3CHO \vee CH_3COOH) \wedge (\neg O_2 \vee CH_3COOH)$$

$$y_1 = CH_3CHO$$
$$y_2 = O_2$$
$$y_3 = CH_3COOH$$

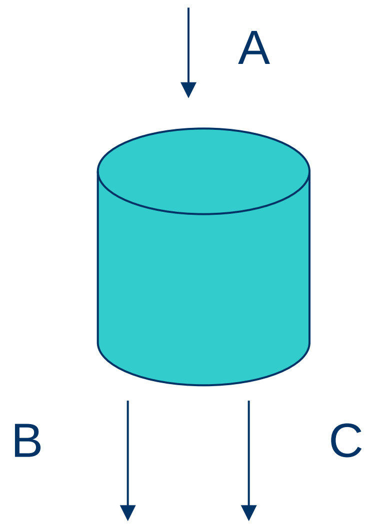
$$1 - y_1 + y_3 \geq 1$$
$$1 - y_2 + y_3 \geq 1$$

$\Leftrightarrow$

$$y_1 - y_3 \leq 0$$
$$y_2 - y_3 \leq 0$$

The optimization problem can be formulated as minimizing a cost under the set of constraints

A dynamic problem: batch reactor



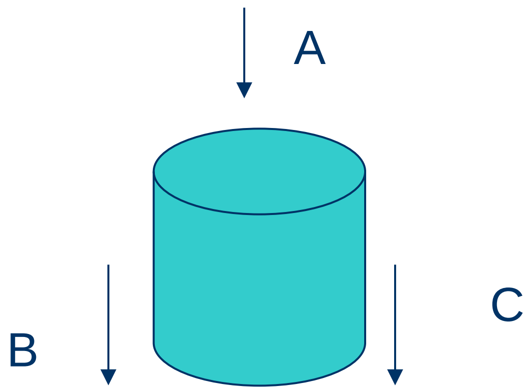
An endothermic batch reactor operates for one hour periods, with a load A according to the parallel reactions A $\rightarrow$ B and A $\rightarrow$ C, but only the B product has commercial value. The speeds of reaction are given by:

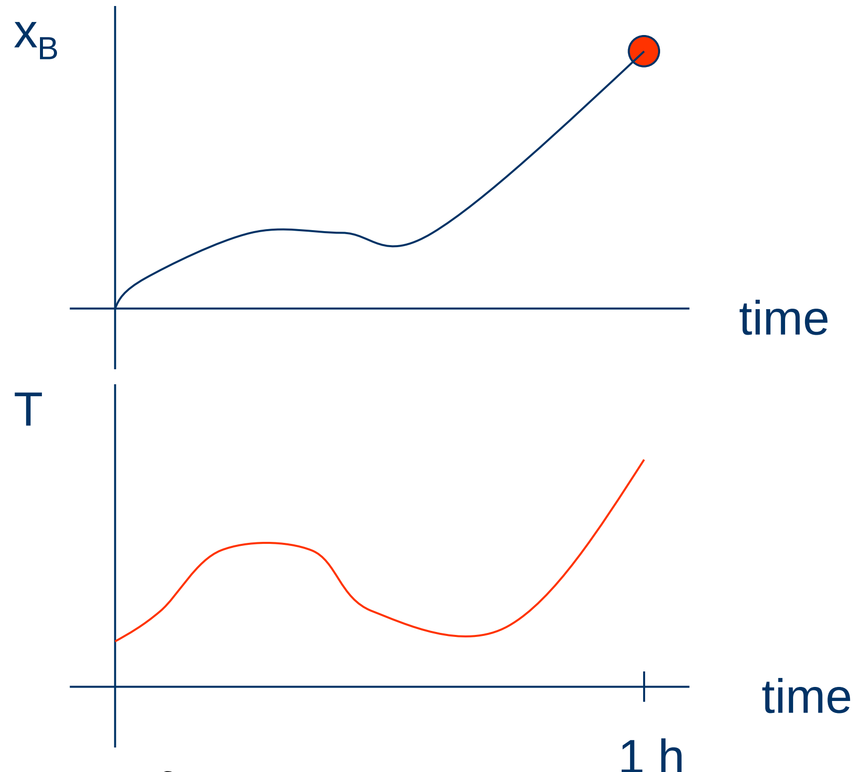
$$k_B = 10^6 \exp(10000 / RT)$$

$$k_C = 5*10^{11} \exp(20000 / RT)$$

Find the temperature profile that maximizes the final production of B, if the temperature must always be bellow 139 ºC

Dynamic Optimization (DO)





$$\max_{T(t)} \quad x_B(1)$$

$$\frac{dx_A}{dt} = -(k_B + k_C)x_A \qquad x_A(0) = A_0$$

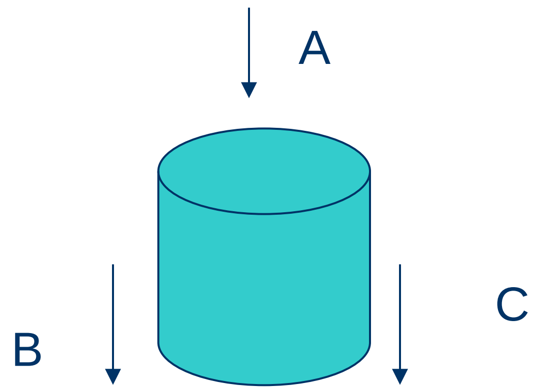
$$\frac{dx_B}{dt} = k_B x_A \qquad x_B(0) = 0$$

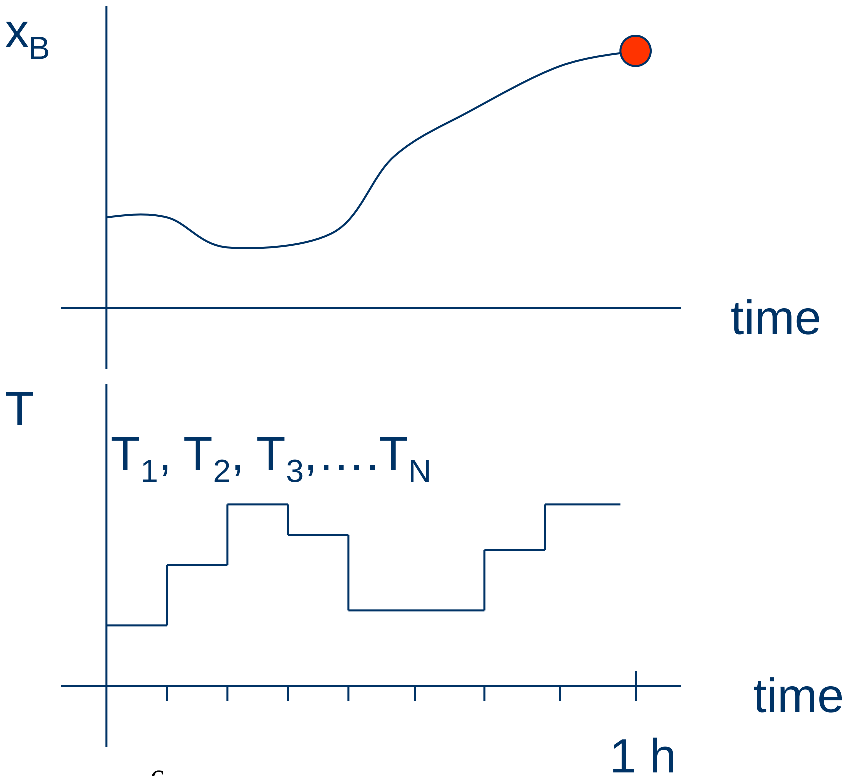
$$T(t) \leq 139$$

$$k_B = 10^6 \exp(10000/RT)$$

$$k_C = 5*10^{11} \exp(20000/RT)$$

Decision variables parameterization





$$\max_{T_i} \quad x_B(1)$$

$$\frac{dx_A}{dt} = -(k_B + k_C)x_A \qquad x_A(0) = A_0$$

$$\frac{dx_B}{dt} = k_B x_A \qquad x_B(0) = 0$$

$$T_i \le 139$$

$$k_B = 10^6 \exp(10000 / RT)$$

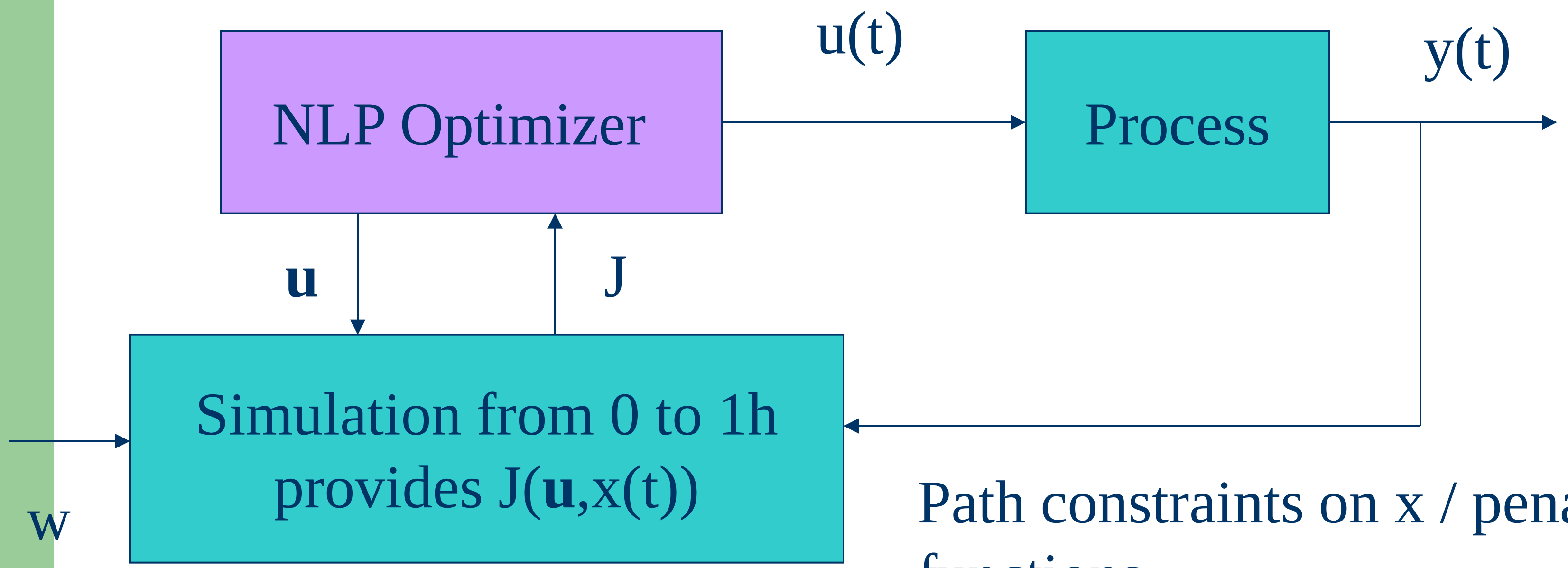
$$k_C = 5 * 10^{11} \exp(20000 / RT)$$

Sequential solution using simulation

$$\min_{u} J(x,u)$$

$$\dot{x}(t) = f(x(t),u(t)) \qquad y(t) = g(x(t), u(t))$$

$$\underline{y} \leq y(t) \leq \overline{y} \qquad \underline{u} \leq u(t) \leq \overline{u}$$



Path constraints on x / penalty functions