Mixed integer linear modeling

Andrés Ramos (Andres.Ramos@comillas.edu)
Pedro Sánchez (Pedro.Sanchez@comillas.edu)
Sonja Wogrin (Sonja.Wogrin@comillas.edu)

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
DEPARTAMENTO DE ORGANIZACIÓN INDUSTRIAL

CONTENTS

- **PROBLEM CLASSIFICATION**
- SEVERAL CHARACTERISTIC PROBLEMS
- FIXED COST PROBLEM
- LOGICAL PROPOSITIONS
- MINIMUM, MAXIMUM AND ABSOLUTE VALUE
- MULTICRITERIA DECISION MAKING

1

Problem Classification

IP Problem classification

- *Linear* problems where several or all the variables are *integer*. A particular case of integer variables are *binary* variables (0/1).

1. PIP (*pure integer programming*) all integer
2. BIP (*binary integer programming*) all binary
3. MIP (*mixed integer programming*) some integer o binary

Justification of optimization problem with integer variables

- Investments are discrete variables (generation or transmission expansion planning, singular equipment acquisition, people hiring)

- Decisions are binary variables (plant or store location)

Binary representation of discrete variables

- x integer variable
- y_i binary variable (0/1)

$$x = \sum_{i=0}^{N} 2^i y_i$$

$$0 \leq x \leq u$$

$$2^N \leq u \leq 2^{N+1}$$

2

Several Characteristic Problems

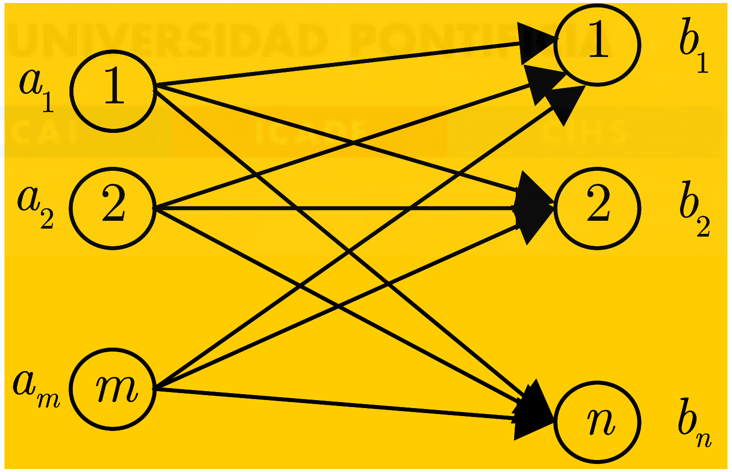
Several LP and BIP characteristic problems

- They have been exhaustively studied. They have limited importance in practice but may appear as part of other problems.
- *Linear Programming LP*
  - Transportation
  - Transshipment
  - Assignment
- *Binary Integer Programming BIP*
  - Knapsack
  - Covering
  - Packing
  - Partitioning
  - Traveling salesman

Transportation problem

- Minimize the total transportation cost of a specific product from origin to destination, satisfying the destination demand without exceeding the origin offer.

- a_i product offer in origin i — m origins
- b_j product demand in destination j — n destinations
- c_{ij} unitary transportation cost from i to j



Transportation problem formulation

$$\min_{x_{ij}} \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}$$

- Offer available in each origin i $\quad \sum_{j=1}^{n} x_{ij} = a_i \quad \forall i = 1, \dots, m$
- Demand in each destination j $\quad \sum_{i=1}^{m} x_{ij} = b_j \quad \forall j = 1, \dots, n$
- $x_{ij} \geq 0$ units of product transported from i to j $\quad \forall i, j$
- Hypothesis: offer equals demand of the product

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$$

- If $\sum_{i=1}^{m} a_i > \sum_{j=1}^{n} b_j$ add a universal sink with zero cost
- If $\sum_{i=1}^{m} a_i < \sum_{j=1}^{n} b_j$ add a universal source with a very high cost

Transportation problem structure

$$
\begin{array}{l|cccc|cccc|c|cccc|}
 & x_{11} & x_{12} & \dots & x_{1n} & x_{21} & x_{22} & \dots & x_{2n} & \dots & x_{m1} & x_{m2} & \dots & x_{mn} \\
\hline
 & 1 & 1 & \dots & 1 & & & & & & & & & \\
m \text{ restricciones de oferta} & & & & & 1 & 1 & \dots & 1 & & & & & \\
 & & & & & & & & & \ddots & & & & \\
 & & & & & & & & & & 1 & 1 & \dots & 1 \\
\hline
 & 1 & & & & 1 & & & & \dots & 1 & & & \\
n \text{ restricciones de demanda} & & 1 & & & & 1 & & & \dots & & 1 & & \\
 & & & \ddots & & & & \ddots & & \dots & & & \ddots & \\
 & & & & 1 & & & & 1 & \dots & & & & 1 \\
\hline
\end{array}
$$

If a_i and b_j are integer $\Rightarrow x_{ij}$ are integers because the matrix is totally unimodular (i.e., every square submatrix has determinant 0, 1 or –1)

Transshipment problem

- Determine in a network of n nodes the cheapest routes to carry product units from their origins to their destinations through intermediate transshipment locations.

- Each *origin* generates $b_i > 0$ units.
- Each *destination* consumes $b_i < 0$ units.
- Each *transshipment* neither generates nor consumes units $b_i = 0$.
- c_{ij} transportation unit cost from i to j in this direction.

Transshipment problem formulation

$$\min_{x_{ij}} \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}$$

- Balance or flow conservation in each node i

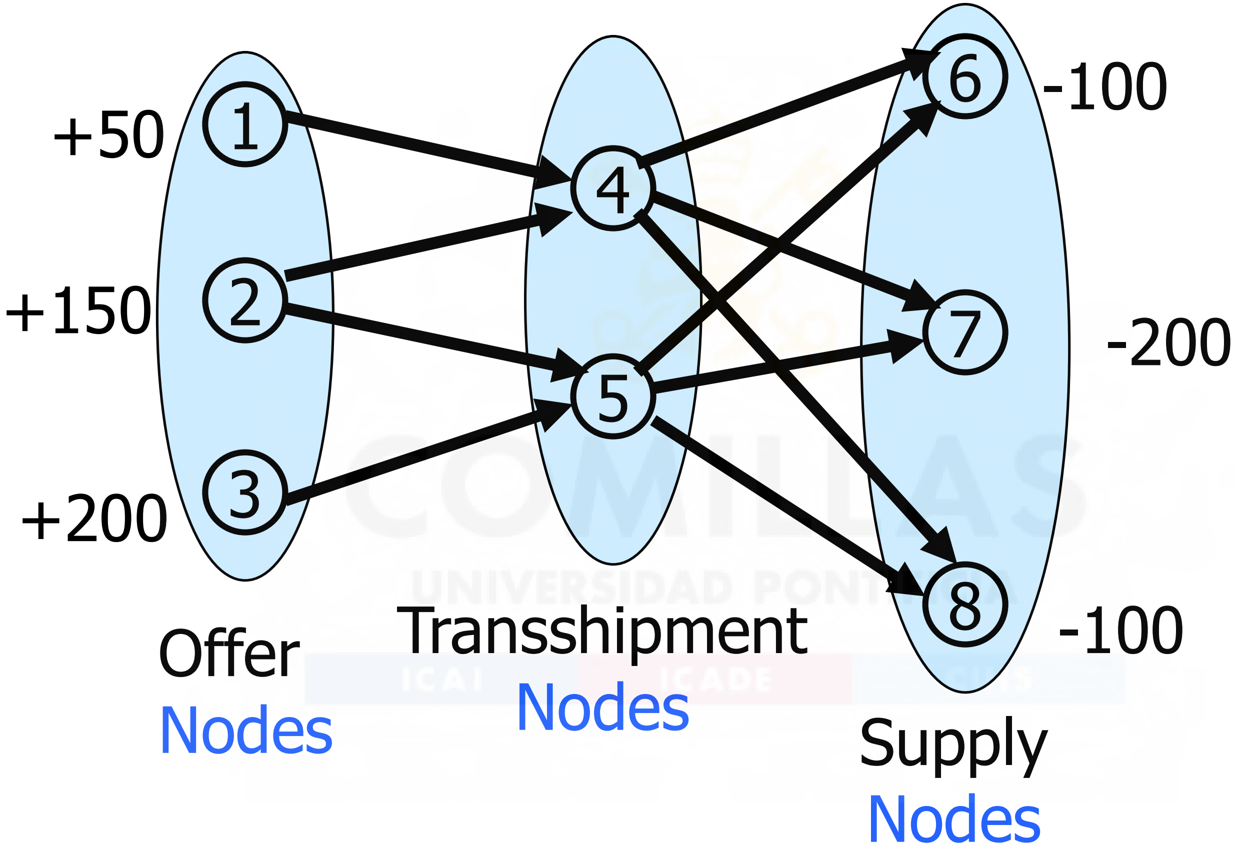
$$\sum_{j=1}^{n} x_{ij} - \sum_{k=1}^{n} x_{ki} = b_i \quad \forall i = 1, \dots, n$$

- x_{ij} units of product transported from i to j $\forall ij$
- Hypothesis: offer equals demand

$$\sum_{i=1}^{n} b_i = 0$$

Transshipment Example

$b_i \geq 0$



$b_i \leq 0$

Task assignment problem

- n tasks
  n persons (machines, etc.) to do them
- It is a particular case of a transportation problem.
- Minimize the total cost of doing the tasks, knowing that each person does one task, and each task is done by one person.
- c_{ij} cost of doing task i by person j

$$x_{ij} = \begin{cases} 1 & \text{if task } i \text{ is done by person } j \\ 0 & \text{otherwise} \end{cases}$$

- *Although it is not necessary to declare them as binary variables*.

Task assignment problem formulation

$$\min_{x_{ij}} \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij} x_{ij}$$

- Each task i is done by one person

$$\sum_{j=1}^{n} x_{ij} = 1 \quad \forall i = 1, \ldots, n$$

- Each person j does one task

$$\sum_{i=1}^{n} x_{ij} = 1 \quad \forall j = 1, \ldots, n$$

$$x_{ij} \geq 0 \quad \forall i, j$$

Task sequencing in one machine (i)

- Given several tasks to do, their duration, and an estimated problem due date, state the mixed integer programming problem to find the sequence that minimizes the mean delay of the tasks with the following data:

Task	A	B	C	D
Processing time	9	12	7	14
Due date	15	19	23	31

Task sequencing in one machine (ii)

- Let d_j be the processing time of task j and r_j the due date of task j.
- Define the problem variables as

$$x_{ij} = \begin{cases} 1 & \text{if task } j \text{ is done in step } i \\ 0 & \text{otherwise} \end{cases}$$

- The objective function will minimize the mean delay

$$\min \frac{1}{4} \sum_i p_i$$

- Subject to these constraints:
  - Each task is done once

$$\sum_i x_{ij} = 1 \quad \forall j$$

  - In each step only one task

$$\sum_j x_{ij} = 1 \quad \forall i$$

Task sequencing in one machine (iii)

- For each step i a task is done, and its due date is $\sum_j r_j x_{ij}$
- On the other hand, task j done in this step end at time $\sum_j d_j \sum_{k\leq i} x_{kj}$

- Variables n_i and p_i, consider if the task ends before due date (prompted) or after (delayed), therefore p_i, is the delay, that appears in the objective function

$$\sum_j d_j \left(\sum_{k\leq i} x_{kj} \right) + n_i - p_i = \sum_j r_j x_{ij} \quad \forall i$$

$$n_i, p_i \geq 0 \quad x_{ij} \in \{0,1\}$$

Knapsack Problem

- n projects
- Maximize the total value of selecting a set of projects without exceeding the available budget.
- c_j cost of project j
- v_j value of project j
- b available budget

$$x_j = \begin{cases} 1 & \text{if project } j \text{ is done} \\ 0 & \text{otherwise} \end{cases}$$

Knapsack Problem Formulation

$$\max_{x_j} \sum_{j=1}^{n} v_j x_j$$

- Limit on the available budget

$$\sum_{j=1}^{n} c_j x_j \leq b$$

$$x_j \in \{0,1\} \quad \forall j$$

Set Covering Problem

- m characteristics (flights)
- n set of characteristics (sequences of flights). If a set is selected, then all characteristics of this set should be done.
- Minimize the total cost of the selected sets so that all characteristics are covered at least once.
- c_j cost of selected set j
- Membership matrix

$$a_{ij} = \begin{cases} 1 & \text{if characteristic } i \text{ belongs to set } j \\ 0 & \text{otherwise} \end{cases}$$

- Decision variables

$$x_j = \begin{cases} 1 & \text{if set } j \text{ is selected} \\ 0 & \text{otherwise} \end{cases}$$

Set Covering Problem Formulation

$$\min_{x_j} \sum_{j=1}^{n} c_j x_j$$

- Each characteristic i should be selected at least once.

$$\sum_{j=1}^{n} a_{ij} x_j \geq 1 \quad i = 1, \dots, m$$

$$x_j \in \{0,1\} \quad j = 1, \dots, n$$

Set Covering Example: Crew Assignment

- An airline company needs to assign crews to cover all its flights. Specifically, it requires solving the set covering problem of three crews whose origin airport is San Francisco for all flights shown in the first column of the table. The rest of the columns show 12 feasible crew flight sequences (sets).
- It is necessary to choose three flight sequences (one for each crew) to cover all flights. It is possible to have more than one crew on the same flight (the extra crew is considered as passengers, although the personnel is paid normally). The assignment cost of a crew to a flight sequence is given in thousands of Euros in the last row.
- The objective is to minimize the total assignment cost of the three crews to cover all flights.

Feasible Flight Sequences

	1	2	3	4	5	6	7	8	9	10	11	12
SF - LA	1			1			1			1		
SF - Denver		1			1			1			1	
SF - Seattle			1			1			1			1
LA - Chicago				2			2		3	2		3
LA - SF	2					3				5	5	
Chicago - Denver				3	3				4			
Chicago - Seattle							3	3		3	3	4
Denver - SF		2		4	4				5			
Denver - Chicago					2			2			2	
Seattle - SF			2				4	4				5
Seattle - LA						2			2	4	4	2
Cost (M€)	2	3	4	6	7	5	7	8	9	9	8	9

Crew Assignment Formulation

$$\min \quad 2x_1 + 3x_2 + 4x_3 + 6x_4 + 7x_5 + 5x_6 + 7x_7 + 8x_8 + 9x_9 + 9x_{10} + 8x_{11} + 9x_{12}$$

- Flight Covering

$$\begin{aligned} x_1 + x_4 + x_7 + x_{10} &\geq 1 \\ x_2 + x_5 + x_8 + x_{11} &\geq 1 \\ x_3 + x_6 + x_9 + x_{12} &\geq 1 \\ &\vdots \end{aligned}$$

- Three crews Assignment

$$\sum_{j=1}^{12} x_j = 3 \qquad x_j \in \{0,1\} \quad j = 1, \dots 12$$

$$x_j = \begin{cases} 1 & \text{if the set } j \text{ is chosen} \\ 0 & \text{otherwise} \end{cases}$$

- Solution
  - $x_3 = x_4 = x_{11} = 1 \quad x_j = 0 \quad j \neq 3, 4, 11$ cost = 18 M€
  - $x_1 = x_5 = x_{12} = 1 \quad x_j = 0 \quad j \neq 1, 5, 12$ cost = 18 M€

Set Packing **Problem**

- m projects
- n project sets. If a set is selected, all projects of this set are done.
- Maximize the total benefit without doing one project more than once.
- c_j benefit of selecting the set j

$$a_{ij} = \begin{cases} 1 & \text{if the project } i \text{ is in the set } j \\ 0 & \text{otherwise} \end{cases}$$

$$x_j = \begin{cases} 1 & \text{if set } j \text{ is selected} \\ 0 & \text{otherwise} \end{cases}$$

Set Packing Problem Formulation

$$\max_{x_j} \sum_{j=1}^{n} c_j x_j$$

- Each project i of all sets cannot be selected more than once

$$\sum_{j=1}^{n} a_{ij} x_j \leq 1 \quad i = 1, \dots, m$$

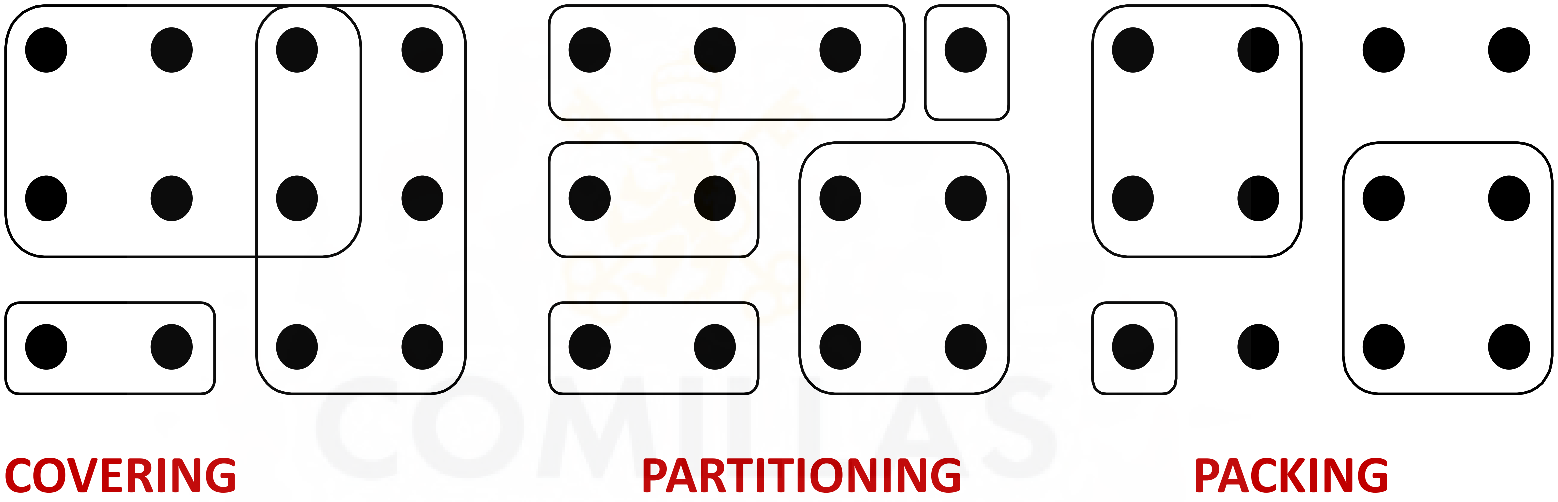
$$x_j \in \{0,1\} \quad j = 1, \dots, n$$

Set Partitioning Problem

- Exactly each characteristic (project) of all sets should be chosen only once

$$\sum_{j=1}^{n} a_{ij} x_j = 1 \quad i = 1, \dots, m$$

Dot Graphs of Covering, Partitioning and Packing Problems



Traveling Salesman Problem (TSP)

- Given a list of cities and the distances between each pair of cities, this problem consists of finding what is the shortest possible route that visits each city exactly once and returns to the origin city
- *Formulation 1:* $x_{ij} = \begin{cases} 1 & \text{if the path between } i \text{ and } j \text{ is included in the route} \\ 0 & \text{otherwise} \end{cases}$

$$\min_{x_{ij}} \sum_i \sum_j c_{ij} x_{ij}$$

$$\sum_i x_{ij} = 1 \ \forall j$$

$$\sum_j x_{ij} = 1 \ \forall i$$

$$\sum_{i,j \in U} x_{ij} \leq \text{Card}(U) - 1 \quad \forall U \subset \{1, \ldots, n\} / 2 \leq \text{Card}(U) \leq n - 2$$

$$x_{ij} \in \{0,1\}$$

Traveling Salesman Problem (TSP)

- *Formulation 2:*

$$x_{ijk} = \begin{cases} 1 & \text{if the path between } i \text{ and } j \text{ is included in the route at stage } k \\ 0 & \text{otherwise} \end{cases}$$

$$\min_{x_{ijk}} \sum_{i,j,k} c_{ij} x_{ijk}$$

$$\sum_{j,k} x_{ijk} = 1 \ \ \forall i$$

We leave every city once

$$\sum_{i,k} x_{ijk} = 1 \ \ \forall j$$

We arrive at every city once

$$\sum_{i,j} x_{ijk} = 1 \ \ \forall k$$

We only make a connection in each stage

$$\sum_{i} x_{ijk} = \sum_{r} x_{jrk+1} \ \ \forall j, k$$

If we are in a city at a certain stage, we leave this city at the following stage

$$x_{ijk} \in \{0,1\}$$

3

Fixed Cost Problem

Fixed Cost Problem

- Objective Cost Function:

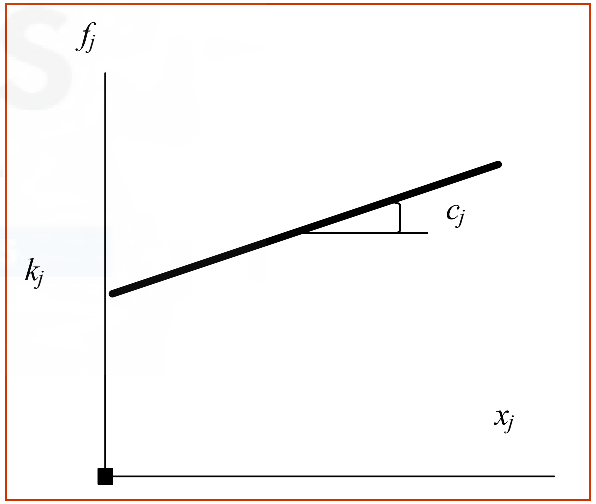
$$f_j(x_j) = \begin{cases} 0 & x_j = 0 \\ k_j + c_j x_j & x_j > 0 \end{cases}$$

- A binary variable y_j represents the binary decision on an activity realization x_j

$$y_j = \begin{cases} 1 & x_j > 0 \\ 0 & x_j = 0 \end{cases}$$

- Mathematical formulation:

$$\min \sum_{j=1}^{n} f_j(x_j) = \sum_{j=1}^{n} (k_j y_j + c_j x_j)$$
$$x_j \leq M_j y_j$$
$$x_j \geq 0 \quad j = 1, \ldots, n$$
$$y_j \in \{0,1\} \quad j = 1, \ldots, n$$



- M_j should have the lowest possible value

Fixed Cost Problem: Unit Commitment on Electric Systems

- Determine thermal generation units should be connected to the electric network each hour of the day (or week) in such a way that:
  - Variable Generation Costs (including fuel and startup/shutdown costs) are minimized.
  - Demand is supplied each hour
  - A specific level of spinning reserve is given
  - Technical limits are fulfilled (minimum/maximum outputs, ramp up/down)

Unit Commitment Problem. Data and Variables

DATA

D_h demand during hour *h* [MW]

R spinning reserve ratio related to demand [p.u.]

a_t **linear coefficient of fuel variable cost of unit *t* [€/MWh]**

b_t **fixed coefficient of fuel variable cost of unit *t* [€/h]**

ca_t startup cost of unit *t* [€]

cp_t shutdown cost of unit *t* [€]

$\bar{P}_t$ maximum output of unit *t* [MW]

$\underline{P}_t$ minimum output of unit *t* [MW]

rs_t ramp up of unit *t* [MW/h]

rb_t ramp down of unit *t* [MW/h]

VARIABLES

P_{ht} output of unit *t* during hour *h* [MW]

A_{ht} commitment of unit *t* during hour *h* {0,1}

AR_{ht} startup of unit *t* during hour *h* {0,1}

PR_{ht} shutdown of unit *t* during hour *h* {0,1}

Unit Commitment Problem. Formulation

$$\min \sum_{h=1}^{H} \sum_{t=1}^{T} (a_t P_{ht} + b_t A_{ht} + ca_t AR_{ht} + cp_t PR_{ht})$$

$$\sum_{t=1}^{T} P_{ht} = D_h \qquad H$$

$$\sum_{t=1}^{T} (\bar{P}_t A_{ht} - P_{ht}) = RD_h \qquad H$$

$$\underline{P}_t A_{ht} \le P_{ht} \le \bar{P}_t A_{ht} \qquad 2HT$$

$$A_{ht} - A_{h-1t} = AR_{ht} - PR_{ht} \qquad (H-1)T$$

$$P_{ht} - P_{h-1t} \le rs_t \qquad (H-1)T$$

$$P_{h-1t} - P_{ht} \le rb_t \qquad (H-1)T$$

$$P_{ht} \ge 0 \qquad A_{ht}, AR_{ht}, PR_{ht} \in \{0,1\}$$

4

Logical Propositions

Logical Propositions

- How to model the proposition: "if product A is produced, then product B should be produced too". The condition of the production of product j is modeled as $x_j \geq 1$. Then, this proposition is mathematically written as

$$x_A \geq 1 \Rightarrow x_B \geq 1$$

- This proposition cannot be included directly (with arrows) in the linear problem. In this example, one constraint $x_B \geq 1$ is included or not depending on the value of variable $x_A \geq 1$ (**endogenous problem**), modifying the structure of the problem.

Disjunctive Propositions (i)

- A couple of constraints where only one (either of the two) should be met, while the other is not necessary. Then, it should meet one constraint but not necessarily both.

$$f(x) \leq 0 \quad \text{or} \quad g(x) \leq 0$$

Example of Disjunctive Propositions (ii)

- One of these two constraints should be met

$$3x_1 + 2x_2 \leq 18 \quad \text{or} \quad x_1 + 4x_2 \leq 16$$

- Adding M (high-value constant) is equivalent to relaxing the constraint (for positive variables with positive coefficients)
  - Relax constraint 1 and fulfill the 2

$$3x_1 + 2x_2 \leq 18 + M$$
$$x_1 + 4x_2 \leq 16$$

  - Relax constraint 2 and fulfill the 1

$$3x_1 + 2x_2 \leq 18$$
$$x_1 + 4x_2 \leq 16 + M$$

- Using an auxiliary binary variable, one of both is fulfilled, and the other one is relaxed

$$3x_1 + 2x_2 \leq 18 + M\delta$$
$$x_1 + 4x_2 \leq 16 + M(1 - \delta)$$

$$\delta = \begin{cases} 1 & \text{if constraint 1 is relaxed} \\ 0 & \text{if constraint 2 is relaxed} \end{cases}$$

Fulfillment at least k of N constraints

- At least k of N ($k < N$) constraints should be met

$$
\begin{aligned}
&f_1(x_1, \dots, x_n) \le d_1 \\
&f_2(x_1, \dots, x_n) \le d_2 \\
&\vdots \\
&f_N(x_1, \dots, x_n) \le d_N
\end{aligned}
$$

- Using $k = 1$ and $N = 2$ is the disjunctive case
- Formulation:

$$
\begin{aligned}
&f_1(x_1, \dots, x_n) \le d_1 + M\delta_1 \\
&f_2(x_1, \dots, x_n) \le d_2 + M\delta_2 \\
&\vdots \\
&f_N(x_1, \dots, x_n) \le d_N + M\delta_N \\
&\sum_{i=1}^{N} \delta_i = N - k \\
&\delta_i \in \{0,1\} \quad i = 1, \dots, N
\end{aligned}
$$

Selecting one from N values

- The equation should fulfill one of the possible values

$$f(x_1,\ldots,x_n)=\begin{cases} d_1 \\ d_2 \\ \vdots \\ d_N \end{cases}$$

- Formulation:

$$f(x_1,\ldots,x_n)=\sum_{i=1}^{N} d_i\delta_i$$

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

$$\delta_i \in \{0,1\} \quad i=1,\ldots,N$$

Simple Propositions (i)

- Using the previous constraint of the fixed cost $x \le M\delta$
- M is an upper bound of x and δ its associated binary variable.
  - If $\delta = 1$ the constraint is relaxed and is met by default $x \le M$
  - If $\delta = 0$ then $x \le 0$
- So, this constraint allows us to model the proposition $\delta = 0 \Rightarrow x \le 0$
- On the other hand, if $x > 0$ then $\delta = 1$. If $x \le 0$ the constraint does not imply anything $x > 0 \Rightarrow \delta = 1$
- Both are equivalent propositions because $P \to Q$ is equivalent to $\text{No } Q \to \text{No } P$

$$\left.\begin{array}{l} \delta = 0 \Rightarrow x \le 0 \\ x > 0 \Rightarrow \delta = 1 \end{array}\right\} x \le M\delta$$

Simple Propositions (ii)

- Analogously, the constraint $x \geq m\delta$
  being m a lower bound x and δ the binary variable.
  - If $\delta = 1$ the constraint does not imply anything as $x \geq m$ is fulfilled by default.
  - If $\delta = 0$ then $x \geq 0$. So, this constraint allows to model the proposition $\delta = 0 \Rightarrow x \geq 0$
- On the other hand, if $x < 0$ then $\delta = 1$. If $x \geq 0$ the constraint does not imply anything. $x < 0 \Rightarrow \delta = 1$
- Both propositions are equivalent as $P \rightarrow Q$ is equivalent to $\text{No } Q \rightarrow \text{No } P$

$$\left.\begin{array}{l} \delta = 0 \Rightarrow x \geq 0 \\ x < 0 \Rightarrow \delta = 1 \end{array}\right\} x \geq m\delta$$

Proposition of $\leq$ constraint (i)

- The proposition $\delta = 1 \rightarrow \sum_j a_j x_j \leq b$

  is modeled as $\sum_j a_j x_j \leq b + M(1 - \delta)$

  being M an upper bound of the constraint for any value x_j of $\sum_j a_j x_j - b \leq M$

  If $\delta = 1$ the original constraint is formulated and if $\delta = 0$ the original constraint is relaxed.

- Analogously to the previous case this constraint models the following proposition

$$\sum_j a_j x_j > b \rightarrow \delta = 0$$

Proposition of $\leq$ constraint (ii)

- The proposition $$\sum_j a_j x_j \leq b \rightarrow \delta = 1$$

  can be replaced by $$\delta = 0 \rightarrow \sum_j a_j x_j > b$$

  or also by $$\delta = 0 \rightarrow \sum_j a_j x_j \geq b + \varepsilon$$

  Both are equivalent to $$\sum_j a_j x_j \geq b + \varepsilon + (m - \varepsilon)\delta$$

  Being m a lower bound of the constraint for any value of x_j

$$\sum_j a_j x_j - b \geq m$$

Proposition of $\geq$ constraint (i)

- Symmetrically propositions of greater than or equal to can be modeled.
- The proposition

$$\delta = 1 \rightarrow \sum_j a_j x_j \geq b$$

is equivalent to

$$\sum_j a_j x_j \geq b + m(1 - \delta)$$

being m a lower bound of the constraint for any value of. x_j

$$\sum_j a_j x_j - b \geq m$$

If $\delta = 1$ the original constraint is formulated and if $\delta = 0$ the original constraint is relaxed.

- Analogously to the previous case this constraint models the next proposition

$$\sum_j a_j x_j < b \rightarrow \delta = 0$$

Proposition of $\geq$ constraint (ii)

- The proposition $\sum_j a_j x_j \geq b \rightarrow \delta = 1$

  can be replaced by $\delta = 0 \rightarrow \sum_j a_j x_j < b$

  or also by $\delta = 0 \rightarrow \sum_j a_j x_j \leq b - \varepsilon$

  Both are equivalent to $\sum_j a_j x_j \leq b - \varepsilon + (M + \varepsilon)\delta$

  Being M an upper bound of the constraint for any value of x_j

  $\sum_j a_j x_j - b \leq M$

Proposition of $=$ constraint (i)

- These equality constraints are replaced by constraints of upper than or equal to constraints and lower than and equal to constraints simultaneously.
- The proposition

$$\delta = 1 \rightarrow \sum_j a_j x_j = b$$

  is equivalent to

$$\delta = 1 \rightarrow \sum_j a_j x_j \leq b$$
$$\delta = 1 \rightarrow \sum_j a_j x_j \geq b$$

- The next two constraints model this equality

$$\sum_j a_j x_j \leq b + M(1-\delta)$$
$$\sum_j a_j x_j \geq b + m(1-\delta)$$

- Effectively when $\delta = 1$ both constraints are fulfilled and when $\delta = 0$ both constraints are relaxed.

Proposition of = constraint (ii)

- The proposition $\sum_j a_j x_j = b \rightarrow \delta = 1$

  is a combination of the previous cases
  $$\sum_j a_j x_j \leq b \rightarrow \delta' = 1$$
  $$\sum_j a_j x_j \geq b \rightarrow \delta'' = 1$$

  and besides $\delta' = 1 \ \text{y} \ \delta'' = 1 \rightarrow \delta = 1$

  The resulting formulation:
  $$\sum_j a_j x_j \geq b + \varepsilon + (m - \varepsilon)\delta'$$
  $$\sum_j a_j x_j \leq b - \varepsilon + (M + \varepsilon)\delta''$$

  and additional constraint that models the fulfillment of previous constraints $\delta' + \delta'' - \delta \leq 1$

Double propositions

- Double propositions are split into two simple propositions.

$$\delta = 1 \leftrightarrow \sum_j a_j x_j \leq b$$

is equivalent to

$$\begin{cases} \delta = 1 \rightarrow \sum_j a_j x_j \leq b \\ \sum_j a_j x_j \leq b \rightarrow \delta = 1 \end{cases}$$

and the same for other types of double propositions

Proposition Modeling Tables (i)

$\delta=1 \rightarrow \sum_j a_j x_j \le b$	$\sum_j a_j x_j \le b + M(1-\delta)$
$\sum_j a_j x_j \le b \rightarrow \delta=1$	$\sum_j a_j x_j \ge b + \varepsilon + (m-\varepsilon)\delta$
$\delta=1 \rightarrow \sum_j a_j x_j \ge b$	$\sum_j a_j x_j \ge b + m(1-\delta)$
$\sum_j a_j x_j \ge b \rightarrow \delta=1$	$\sum_j a_j x_j \le b - \varepsilon + (M+\varepsilon)\delta$
$\delta=1 \rightarrow \sum_j a_j x_j = b$	$\sum_j a_j x_j \le b + M(1-\delta)$ $\sum_j a_j x_j \ge b + m(1-\delta)$
$\sum_j a_j x_j = b \rightarrow \delta=1$	$\sum_j a_j x_j \ge b + \varepsilon + (m-\varepsilon)\delta'$ $\sum_j a_j x_j \le b - \varepsilon + (M+\varepsilon)\delta''$ $\delta' + \delta'' - \delta \le 1$

Double Proposition Modeling Tables (ii)

$\delta = 1 \leftrightarrow \sum_j a_j x_j \le b$	$\sum_j a_j x_j \le b + M(1-\delta)$ $\sum_j a_j x_j \ge b + \varepsilon + (m-\varepsilon)\delta$
$\delta = 1 \leftrightarrow \sum_j a_j x_j \ge b$	$\sum_j a_j x_j \ge b + m(1-\delta)$ $\sum_j a_j x_j \le b - \varepsilon + (M+\varepsilon)\delta$
$\delta = 1 \leftrightarrow \sum_j a_j x_j = b$	$\sum_j a_j x_j \le b + M(1-\delta)$ $\sum_j a_j x_j \ge b + m(1-\delta)$ $\sum_j a_j x_j \ge b + \varepsilon + (m-\varepsilon)\delta'$ $\sum_j a_j x_j \le b - \varepsilon + (M+\varepsilon)\delta''$ $\delta' + \delta'' - \delta \le 1$

Basket Team Problem

- A basket coach has 9 players that are ranked from 1 to 3 based on their skills in ball handle, shot, rebound, and defense

Player	Positions	Handle	Shot	Rebound	Defense
1	Pivot	2	1	3	3
2	Guard	3	3	1	2
3	Pivot, Forward	2	3	2	2
4	Forward, Guard	1	3	3	1
5	Pivot, Forward	1	3	1	2
6	Forward, Guard	3	1	2	3
7	Pivot, Forward	3	2	2	1
8	Pivot	2	1	3	2
9	Forward	3	3	1	3

Basket Team (Logical Constraints)

- The team should be composed by 5 players that should have the maximum defense value satisfying the following conditions:
  1. At least two players should be able to play as pivot, two as forward, and one as a guard. Each player only plays in one position.
  2. Their average value of ball handle, shot, and rebound should be upper or equal to 2.
  3. If player 3 is selected, then player 6 cannot be selected.
  4. If player 1 is selected, player 4 or player 5 should be selected, but not both. If player 1 is not selected, players 4 and 5 may be selected.
  5. The player 8 or 9, but not both, should be selected.

- Formulate a linear problem to optimize the basket team.

$\max 3x_1 + 2x_2 + 2x_3 + x_4 + 2x_5 + 3x_6 + x_7 + 2x_8 + 3x_9$

$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 = 5$

$x_1 + y_{3p} + y_{5p} + y_{7p} + x_8 \geq 2$

$y_{3a} + y_{4a} + y_{5a} + y_{6a} + y_{7a} + x_9 \geq 2$

$x_2 + y_{4b} + y_{6b} \geq 1$

$2x_1 + 3x_2 + 2x_3 + x_4 + x_5 + 3x_6 + 3x_7 + 2x_8 + 3x_9 \geq 10$

$x_1 + 3x_2 + 3x_3 + 3x_4 + 3x_5 + x_6 + 2x_7 + x_8 + 3x_9 \geq 10$

$3x_1 + x_2 + 2x_3 + 3x_4 + x_5 + 2x_6 + 2x_7 + 3x_8 + x_9 \geq 10$

$x_3 = 1 \Rightarrow x_6 = 0$ is equivalent to $x_3 \leq 0$ or $x_6 \leq 0$

$x_3 \leq y_1$

$x_6 \leq 1 - y_1$ or alternatively $x_3 + x_6 \leq 1$

$x_1 \geq 1 \rightarrow x_4 + x_5 = 1$ is equivalent to $x_1 \leq 0$ ó $(x_4 + x_5 \leq 1$ y $x_4 + x_5 \geq 1)$

$x_1 \leq y_2$

$x_4 + x_5 - 1 \leq (1 - y_2)$ or alternatively $x_4 + x_5 \leq 2 - x_1$, $x_4 + x_5 \geq x_1$

$x_4 + x_5 - 1 \geq -1(1 - y_2)$

$x_8 + x_9 = 1$

$y_{3p} + y_{3a} - x_3 = 0$

$y_{4a} + y_{4b} - x_4 = 0$

$y_{5p} + y_{5a} - x_5 = 0$

$y_{6a} + y_{6b} - x_6 = 0$

$x_{7p} + x_{7a} - x_7 = 0$

$x_i, y_i \in \{0,1\}$

- Two optimal solutions

$x_1 = x_2 = x_3 = x_5 = x_8 = 1$

$x_1 = x_4 = x_6 = x_7 = x_9 = 1$

and the rest of players = 0

Notation equivalence:

p: Pivot

a: Forward (alero)

b: Point Guard (base)

Product of Binary Variables

$\delta_1 \delta_2 = 0$ $\delta_i \in \{0,1\}$	$\delta_1 = 0$ o $\delta_2 = 0$	$\delta_1' + \delta_2' \geq 1$ $\delta_1 + \delta_1' = 1$ $\delta_2 + \delta_2' = 1$ $\delta_i, \delta_i' \in \{0,1\}$
$\delta_1 \delta_2$ $\delta_i \in \{0,1\}$	Reemplazar $\delta_1 \delta_2$ por δ_3 $\delta_3 = 1 \leftrightarrow \delta_1 = 1$ y $\delta_2 = 1$	$\delta_3 \leq \delta_1$ $\delta_3 \leq \delta_2$ $\delta_1 + \delta_2 \leq 1 + \delta_3$ $\delta_i \in \{0,1\}$
$x\delta$ $x \geq 0$ $\delta \in \{0,1\}$	Reemplazar $x\delta$ por y $\delta = 0 \rightarrow y = 0$ $\delta = 1 \rightarrow y = x$	$y \geq 0$ $y \leq M\delta$ $-x + y \leq 0$ $x - y + M\delta \leq M$ $x \leq M$

5

Minimum, Maximum and Absolute Value

Minimum or maximum of variables

$$\min z \qquad\qquad \min z$$

$$z = \max(x, y) \quad \rightarrow \quad z \geq x, \quad z \geq y$$

$$\max z \qquad\qquad \max z$$

$$z = \min(x, y) \quad \rightarrow \quad z \leq x, \quad z \leq y$$

Modeling the absolute value

$$z \leq |x| \Rightarrow -x \leq z \leq x$$

$$\begin{array}{ll} min|x| & \Rightarrow \min(x^+ + x^-) \\ \text{s.t.} & \text{s.t.} \\ x \in X & x = x^+ - x^- \\ & x \in X \\ & x^+, x^- \geq 0 \end{array}$$

- PROBLEM CLASSIFICATION
- SEVERAL CHARACTERISTIC PROBLEMS
- FIXED COST PROBLEM
- LOGICAL PROPOSITIONS
- MINIMUM, MAXIMUM AND ABSOLUTE VALUE
- **MULTICRITERIA DECISION MAKING**

6

Multicriteria Decision Making

Which is the best animal in terms of running, flying, and swimming simultaneously?

- The fastest runner?

Cheetah is the fastest running animal in the world.

- The fastest flying?

Peregrine falcon is the fastest bird in the world.

- The fastest swimmer?

Sailfish is the fastest fish in the world.

And the winner is … the DUCK

- Can run, although less than the cheetah

- Can fly, although less than the peregrine falcon

- Can swim, although less than the sailfish

Definition

- Decision: choosing the best of possible → To state the best and possible
- Possible solutions or feasible decisions:
  - Finite set: alternatives can be enumerated
  - Continuous set: alternatives are defined through constraints
- The Best:
  - One criterion (classic Optimization and classic Decision Theory)
  - Multiple criteria o multiple decision makers (Game Theory and **Multicriteria Decision Making**)
- General statement:

$$\begin{array}{l} \text{opt} \quad z = (z_1(x), \ldots, z_p(x)) \\ x \in F \end{array}$$

- F: feasible decision space
- $z(F)$: criterion space

Basic Concepts

- Attribute: observable "value" (measurable) of an alternative, independent of the decision-maker

- Objective: direction to improve an attribute (max. o min. if numerical; otherwise, preferences)

- Target: an acceptable level of achievement for an attribute

- Goal: a combination of an attribute with its target

- Criterion: relevant attributes, objectives, or goals to a decision problem

Solution Concept

Efficiency or Pareto optimality criterion

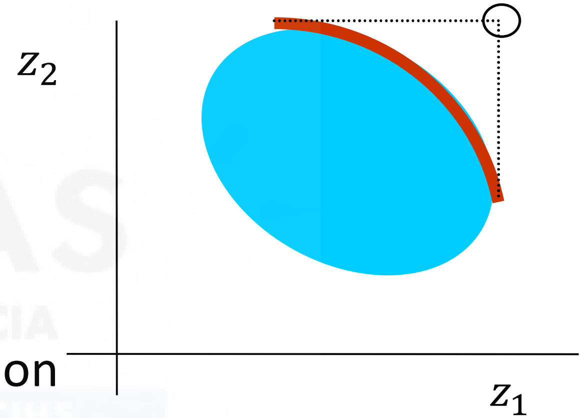
One feasible solution is efficient or Pareto optimal if no other feasible solution can improve one attribute without causing a degradation in at least another attribute.

Dominated or non-efficient alternative: there is another one with better attributes.

Criteria space $\max(z_1, z_2)$

Efficient set or Pareto frontier

Dominated alternatives



Numerical attributes, objectives maximisation

→ Efficient set:

$$\varepsilon = \{x \in F : \nexists x' \in F \text{ with } z_k(x') \geq z_k(x)\ \forall k \text{ and } \exists t \in \{1, \ldots, p\} \text{ with } z_t(x') > z_t(x)\}$$

Best compromise solution: efficient solution chosen by the decision maker

Multiobjective optimization

Weighted-Sum Method (Zadeh, 1963)

Multiplying any objective function with a weight or nonnegative factor and adding in a single composite objective function

Parametric programming (the efficient set is obtained by changing weights)

$$\left.\begin{array}{r} \max \sum_{i=1}^{p} \lambda_i z_i(x) \\ x \in F \\ \lambda \geq 0 \end{array}\right\}$$

Theorem: If $\lambda_i > 0 \quad \forall i$ then any optimal solution of $P(\lambda)$ is efficient.

The converse of the theorem is true under some assumptions (convexity, linearity)

Normalized criteria (units)

Compromise Programming

Ideal point: optimum values of each objective subject to problem constraints

$z^* = (z_1^*, \dots, z_i^*, \dots, z_p^*)$ $\quad z_i^* = \max_{x \in F} z_i(x)$ $\quad z_i^*$: anchor value

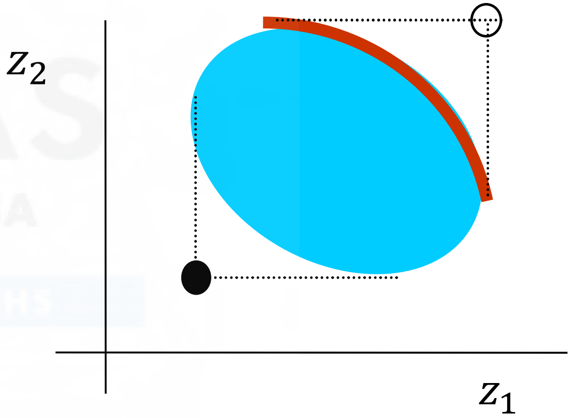
Optimum element or best-compromise solution: efficient solution closest to the **ideal point** (Zeleny's axiom of choice, 1976)

Degree of closeness between attribute i-th and its anchor value normalized

z_{*i} anti-ideal or nadir point (worst value criteria in the efficient set)

$$d_i(x) = \frac{z_i^* - z_i(x)}{z_i^* - z_{*i}}$$

$$\min_{x \in F} L_\pi = \left[\sum_{i=1}^{p} {w_i}^\pi \left(\frac{z_i^* - z_i(x)}{z_i^* - z_{*i}} \right)^\pi \right]^{1/\pi}$$



Weights, the importance of discrepancy of criteria (subjective → ordering, Saaty, ...)

π=∞ → Chebyshev or minimizing maximum distance (linear)

π=1 → lineal addition of weighted distances (linear)

Compromise set: varying π (usually $[L_1, L_\infty]$)

Goal Programming

Satisfying logic (Simon, 1955):

Decision-making behavior where, instead of attempting to optimize system performance, a **level of aspiration** is set either subjectively or heuristically, and no further effort is expended to exceed that level of performance → Goal programming (Charnes y Cooper(61), Lee (72) e Ignizio (76))

Attribute → mathematical expression: $z_i(x)$

Target or level of aspiration: acceptable **level of achievement** $\hat{z}_i$

Goal: $z_i(x) \geq \hat{z}_i$

With deviation variables: $z_i(x) + n_i - p_i = \hat{z}_i$

Deviation variables to be minimized:

if goal is "at least" one value, n_i; if it is "at most", p_i

Model (if at least): $\min_{x \in F \cap \text{goal constraints}} \sum_{i=1}^{p} n_i$

Andrés Ramos
https://www.iit.comillas.edu/aramos/
Andres.Ramos@comillas.edu

Pedro Sánchez
Pedro.Sanchez@comillas.edu

Sonja Wogrin
Sonja.Wogrin@comillas.edu