Scuola universitaria professionale
della Svizzera italiana
SUPSI

Università
della
Svizzera
italiana

Istituto Dalle Molle di studi sull'intelligenza artificiale

Applied Operations Research (C-B4051E.1) Bachelor in Data Science and AI

Lecture 3

ILP - Modelling

Matteo Salani
matteo.salani@idsia.ch

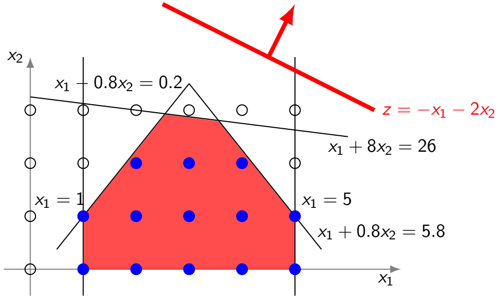
Integer Linear Programming (ILP)

$$\begin{aligned} min\ z &= c^T x \\ s.t.\ Ax &\geq b \\ x &\in \mathbb{Z}^n \end{aligned}$$

- The integrality constraints are non-linear and non-convex
- $x \geq 0,\ x' \subseteq x \in \mathbb{Z}^n$ Mixed Integer Linear Programming (MILP)
- $x \in \{0, 1\}$ Binary Programming (BP)

Example

$$\begin{aligned} \max\ z &= -x_1 - 2x_2 \\ s.t.\ x_1 &\geq 1 \\ x_1 &\leq 5 \\ x_1 + 0.8x_2 &\leq 5.8 \\ x_1 - 0.8x_2 &\geq 0.2 \\ x_1 + 8x_2 &\leq 26 \\ x &\in \mathbb{Z}^2 \end{aligned}$$



- The feasible region consists only of the **blu** points.

Modelling - the use of binary variables

Variables that can only take on binary values $\in \{0, 1\}$ are particularly useful for the mathematical modelling of problems.

$$x_i = \begin{cases} 1 & \text{event } i \text{ occurs} \\ 0 & \text{event } i \text{ does not occur} \end{cases}$$

- $\sum_{i=1}^{n} x_i \leq 1$ at most 1 event occurs
- $\sum_{i=1}^{n} x_i = 1$ exactly 1 event occurs
- $\sum_{i=1}^{n} x_i \geq 1$ at least 1 event occurs
- $x_1 = x_2$ events 1 and 2 both or none of them occur.
- $x_1 \leq x_2$ if event 1 occurs, then event 2 occurs too.

Modelling - Implication and big-M constraint

In practice, it often happens that continuous variables are logically related to binary variables.
Let $x \geq 0$ and $y \in \{0, 1\}$, We want to model the following logical relation:

$$x > 0 \Rightarrow y = 1$$

$$x = 0 \Rightarrow y = 0$$

Knowing a majorant of the variable x (known as "bigM", M) the above implication can be expressed as

$$x \leq M \cdot y$$

On the other hand, the implication $y = 1 \Rightarrow x = 0$ can be modelled by the linear constraint

$$x \leq M \cdot (1 - y)$$

Modelling - Logical conditions - Implication

We have already seen big-M constraints when modelling fixed costs (lot sizing))

$$\min z = \sum_{t \in T} (c_t x_t + \mathrm{f_t} y_t + h_t I_t)$$

and when modelling resource constraints with activation

$$\left.\begin{array}{rl} 0 \leq x_t \leq q & if \;\; y_t = 1 \\ x_t = 0 & if \;\; y_t = 0 \end{array}\right\} \Rightarrow 0 \leq x_t \leq q \cdot y_t$$

Basic models - Knapsack problem

- N items
- v_n value of item n, $\forall n \in 1, \cdots, N$.
- w_n weight of item n, $\forall n \in 1, \cdots, N$.
- q capacity of the knapsack
- $x_n \in \{0, 1\}$, $\forall n \in 1, \cdots, N$ takes value 1 if item n is selected , 0 otherwise.

$$\begin{aligned} \max z = \sum_{n=1}^{N} v_n \cdot x_n & \\ s.t. \sum_{n=1}^{N} w_n x_n \leq q & \\ x_n \in \{0, 1\} & \qquad \forall n \in 1, \cdots, N \end{aligned}$$

Basic models - Knapsack problem

In some cases, we have multi-dimensional knapsacks when items have multiple attributes (e.g. volume and weight) and limits are defined over each attribute.

- N items
- M attributes
- v_n value of item n, $\forall n \in 1, \cdots, N$.
- w_n^m weight of item n, $\forall n \in 1, \cdots, N$ related to attribute m.
- q^m capacity of the knapsack for the $m-th$ attribute
- $x_n \in \{0, 1\}$, $\forall n \in 1, \cdots, N$ takes value 1 if item n is selected , 0 otherwise.

$$\max z = \sum_{n=1}^{N} v_n \cdot x_n$$

$$s.t. \ \sum_{n=1}^{N} w_n^m x_n \le q^m \qquad \forall m \in M$$

$$x_n \in \{0, 1\} \qquad \forall n \in 1, \cdots, N$$

Basic models - Assignment problem

Let J be a set of jobs and P a set of persons ($|J| = |P|$). Assign each job to a person. Let $c_{j,p}$ be the cost of assigning job j to person p.

Minimize the cost of the assignment

We define $|J| \times |P|$ binary variables. Variable $x_{j,p} \in \{0, 1\}$ equals 1 if job j is assigned to person p, 0 otherwise.

$$
\begin{aligned}
\min z = \sum_{j=1}^{J} \sum_{p=1}^{P} c_{j,p} \cdot x_{j,p} & \\
s.t. \ \sum_{j=1}^{J} x_{j,p} = 1 & \qquad \forall p \in 1, \cdots, P \\
\sum_{p=1}^{P} x_{j,p} = 1 & \qquad \forall j \in 1, \cdots, J \\
x_{j,p} \in \{0, 1\}, & \qquad \forall j \in 1, \cdots, J, p \in 1, \cdots, P
\end{aligned}
$$

Modelling - Completion time problem

Let J be a set of jobs and P a set of persons ($|J| > |P|$). Assign zero or more tasks to each person. Let $t_{j,p}$ be the time taken by person p to complete *job*.

Minimize the time taken to complete all jobs

We define $|J| \times |P|$ binary variables. Variable $x_{j,p} \in \{0,1\}$ equals 1 if job j is assigned to person p, 0 otherwise.
We can compute the time taken by person p to complete all his/her jobs:

$$t_p = \sum_{j=1}^{J} t_{j,p} \cdot x_{j,p} \quad \forall p \in P$$

The time taken to complete all jobs is equal to the longest completion time:

$$t = \max_{p \in P} t_p = \max_{p \in P} \sum_{j \in J} t_{j,p} \cdot x_{j,p}$$

Modelling - Completion time problem

We want to minimize t :

$$\min t = \max_{p \in P} t_p = \max_{p \in P} \sum_{j \in J} t_{j,p} \cdot x_{j,p}$$

We introduce an artificial variable w which must be greater than the completion time of each person:

$$w \geq t_p \quad \forall p \in P$$

that is:

$$w \geq \sum_{j \in J} t_{j,p} \cdot x_{j,p} \quad \forall p \in P$$

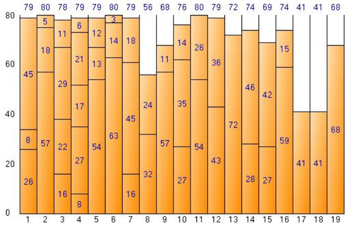
Modelling - Completion time problem

Complete Model:

$$\begin{aligned}
\min z = w& \\
s.t.\ w \geq \sum_{j \in J} t_{j,p} \cdot x_{j,p} \quad \forall p \in P& \\
\sum_{p \in P} x_{j,p} = 1& \qquad \forall j \in P \\
x_{j,p} \in \{0,1\},& \qquad \forall j \in 1, \cdots, J, p \in 1, \cdots, P
\end{aligned}$$

Basic models - Bin packing problem

A set N of different objects of size p_n must be placed in a set C of bins of fixed capacity q. The task is to minimize the number of containers necessary to fit all items.



We define:

- $x_{n,m} \in \{0,1\}$ if object n is placed in bin m, 0 otherwise.
- $y_m \in \{0,1\}$ equal 1 if bin m contains at least one object (i.e. it is used), 0 otherwise.

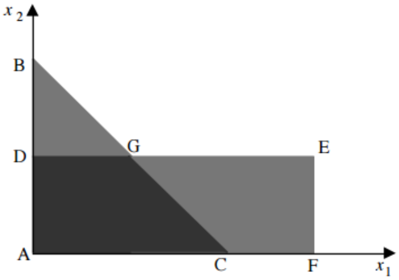
Basic models - Bin packing problem

Complete model:

$$\begin{aligned}
\min z = \sum_{m \in M} y_m & \\
s.t. \sum_{n \in N} p_n \cdot x_{n,m} \leq q \cdot y_m & \qquad \forall m \in M \\
\sum_{m \in M} x_{n,m} = 1 & \qquad \forall n \in N \\
x_{n,m} \in \{0, 1\}, & \qquad \forall n \in N, m \in M
\end{aligned}$$

Modelling - Disjunctive constraints

A decision problem may require to fulfill a disjoint set of constraints.



In formule:

$$\begin{cases} A_1 \cdot x \leq b_1 \vee A_2 \cdot x \leq b_2 \vee \cdots \vee A_i \cdot x \leq b_i \\ x \in \mathbb{R}^n \end{cases}$$

Modelling - Disjunctive constraints

Binary variables can be used!
We define as many binary variables as there are (groups of) constraints

$$y_i \begin{cases} 1 & \text{constraint } i \text{ is active} \\ 0 & \text{otherwise} \end{cases}$$

We estimate a majorant M_i for each constraint

$$A_i \cdot x - b_i \leq M_i \cdot (1 - y_i)$$

We impose the condition that exactly one of the constraints is active.

$$\sum_{i \in I} y_i = 1$$

It is possible to generalize (depending on the application) to satisfy p constraints

$$\sum_{i \in I} y_i = p$$

Modelling - Disjunctive constraints

Example:

$$\begin{cases} 5 \cdot x \leq 3 \vee -4 \cdot x \leq 4 \\ x \in \mathbb{R} \end{cases}$$

Becomes:

$$\begin{cases} 5 \cdot x - 3 \leq M(1 - y_1) \\ -4 \cdot x - 4 \leq M(1 - y_2) \\ y_1 + y_2 = 1 \\ x \in \mathbb{R}, y \in \{0, 1\} \end{cases}$$

With just two constraints we can give a meaning to value the value 0 of a binary value, saving on the number of variables:

$$\begin{cases} 5 \cdot x - 3 \leq M \cdot y \\ -4 \cdot x - 4 \leq M \cdot (1 - y) \\ x \in \mathbb{R}, y \in \{0, 1\} \end{cases}$$

Basic models - Job Sequencing - Scheduling

- We fine job sequencign problems in manufacturing and project planning.
- In general for this type of problems it is required to find the best sequence of jobs that access to limited resources (machines, operators, etc.)
- These problems often require to minimize the completion time or the delay with respect to a due date.

Basic models - Job Sequencing - Scheduling

Let N be the set of jobs to perform on a machine. Let p_n be the completion time of job $n \in N$. Determine the starting time of job n so that job do not overlap.

Decision variables are the starting times of jobs: $t_n \ \forall n \in N$.

Sequencing constraints (so that jobs do not overlap) are disjunctive constraints:

$$t_j \geq t_i + p_i \vee t_i \geq t_j + p_j \qquad \forall i, j \in N$$

Basic models - Job Sequencing - Scheduling

We need to explicitly represent the sequencing of jobs using binary variables:

$$y_{i,j} = \begin{cases} 1 & \text{job } i \text{ starts before job } j \\ 0 & \textit{otherwise} \end{cases}$$

We activate the respective constraint according to variables y_{ij}):

$$\begin{cases} t_j \geq t_i + p_i & \textit{is } y_{ij} = 1 \\ t_i \geq t_j + p_j & \textit{is } y_{ji} = 1 \end{cases}$$

Basic models - Job Sequencing - Scheduling

If $y_{ij} = 1$ it must be $y_{ji} = 0$. So we need additional contraints:

$$t_i + p_i - t_j \leq M(1 - y_{ij}) \qquad \forall i, j \in N$$

$$y_{ij} + y_{ji} = 1 \qquad \forall i, j \in N$$

we have seen in disjunctive programming that we can exploit the value 0 of binary variables. We define just ($N(N-1)$ constraints) restricting to job pairs (i, j) such that $i < j$:

$$t_i + p_i - t_j \leq M(1 - y_{ij}) \qquad \forall i, j \in N | i < j$$

$$t_j + p_j - t_i \leq M y_{ij} \qquad \forall i, j \in N | i < j$$

Basic models - Job Sequencing - Scheduling

So far we did not define any objective function. There are many:

- average completion time of all jobs

$$\min z = \frac{1}{N} \sum_{i \in N} t_i + p_i$$

- when a ready time r_j is defined for each job, we can minimize the average waiting time:

$$\min z = \frac{1}{N} \sum_{i \in N} (t_i - r_i)$$

Basic models - Job Sequencing - Scheduling

When a due time d_i is defined for each job we can have some objective functions:

- the average delay:

$$\min z = \frac{1}{N} \sum_{i \in N} (t_i + p_i - d_i)$$

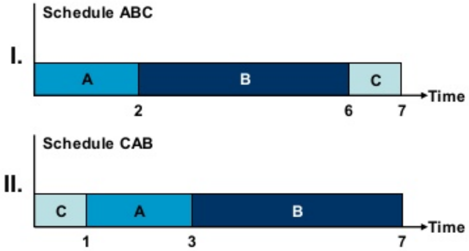
- the worst delay (exploiting min-max linearization):

$$\begin{aligned} &\min z = w \\ &\quad s.t.\ w \geq t_i + p_i - d_i \qquad \forall i \in N \end{aligned}$$

Basic models - Job Sequencing - Scheduling

Example:

- $N = \{A,B,C\}$
- $p_n = \{2,4,1\}$



Solution:

- $I: t_n = \{0, 2, 6\}, z = \frac{1}{3}((0+2)+(2+4)+(6+1)) = 5$
- $II: t_n = \{1, 3, 0\}, z = \frac{1}{3}((0+1)+(1+2)+(3+4)) = 3.6$

Basic models - Covering, Packing, Partitioning

Among the classical combinatorial optimization problems, we find generic problems defined on sets.

- Given a finite set of elements $N = \{1, 2, \cdots, n\}$
- Given a vector of costs $c^T = [c_1, c_2, \cdots, c_n]$
- Given a collection S of subsets of N, $S = \{F_1, F_2, \cdots, F_{|S|}\}$
- The cost of each subset is $c(F_j) = \sum_{i \in F_j} c_i$

We define the following optimization problems:

- Covering problem: Select the subsets F_j in such a way that they cover each element of N at a minimum total cost. That is $\cup_j F_j = N$
- Partitioning problem: Select the subsets F_j in such a way that each element belongs to exactly one selected set, i.e. $F_i \cap F_j = \emptyset$ e $\cup_j F_j = N$
- Packing problem: Select the subsets F_j in such a way that each element of N belongs to at most one subset and the total cost is maximized. That is $F_i \cap F_j = \emptyset$

Basic models - Covering, Packing, Partitioning

- For mathematical modelling, it is convenient to define a parameter $a_{ij}, i \in N, j \in F$ equal to 1 if the element i is in the subset j, 0 otherwise.
- We introduce binary variables $x_j \in \{0,1\}, \forall j \in S$.

We can model the problems:

- Covering:

$$\min z = \sum_{j \in S} c(F_j) \cdot x_j, s.t. \sum_{j \in S} a_{ij} \cdot x_j \geq 1 \quad \forall i \in N$$

- Partitioning:

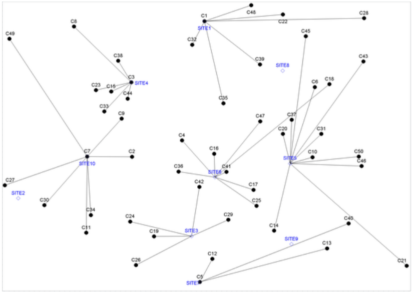
$$\max / \min z = \sum_{j \in S} c(F_j) \cdot x_j, s.t. \sum_{j \in S} a_{ij} \cdot x_j = 1 \quad \forall i \in N$$

- Packing:

$$\max z = \sum_{j \in S} c(F_j) \cdot x_j, s.t. \sum_{j \in S} a_{ij} \cdot x_j \leq 1 \quad \forall i \in N$$

Set Covering - Example

We have to decide where to locate mobile emergency stations (ambulances). Each station is able to serve a certain number of points in the city within a given time. An emergency station can be located anywhere in the city. Minimize the number of emergency stations to be distributed over the territory so that all points of the city can be reached from at least one station.



Set Partitioning - Example

A newly married couple has to organize the wedding dinner. Given the list of guests, they must be organised in groups of 8 persons. For each group it is possible to evaluate the degree of acquaintance of the members (kinship, friendship, etc.). Maximise the overall level of knowledge so that all guests have a place in the dining room and no guest is placed at more than one table.

Set Packing - Example

An administration has to decide on the location of some waste disposal centres. Each centre is close to a certain number of houses. According to the chosen location each waste disposal centre has a given capacity. Maximize the total capacity of the disposal centres, taking into account that each house is close to at most one waste disposal centre.

A funny example - Sudoku

Rules of Sudoku:

- A table with 9 rows and 9 columns is divided in 9 3×3 squares.
- Fill in each cell using the numbers between 1 and 9 according to the following rules:
  - each cell contains exactly one number;
  - each number appears exactly once:
    - in every row;
    - in every column;
    - in every 3×3 square;
- Certain numbers in some cells are fixed and can not be changed.

	7				2			
9				6		3		
			1				8	
		8		3				1
	9		7		4		6	
3				5		9		
	1				7			
		6		1				5
			4				3	

A funny example - Sudoku

A natural choice is to de
ne a binary decision variable for each cell, for each value $\in \{1, \cdots, 9\}$:

- $x_{i,j,k} \in \{0, 1\}$
  - equals 1 if cell i, j contains value k;
  - 0, otherwise
  - i is the row index $i \in \{1, \cdots, 9\}$
  - j is the column index $j \in \{1, \cdots, 9\}$
  - k is the value index $k \in \{1, \cdots, 9\}$
- There are exactly 9^3 variables.
- Some variables are fixed. In the example above:
  - $x_{2,1,9} = 1$
  - $x_{1,2,7} = 1$
  - $\ldots$

A funny example - Sudoku

Constraints:

- each cell has to contain exactly one value:

$$\sum_{k=1}^{9} x_{i,j,k} = 1 \qquad \forall i, j \in \{1, \cdots, 9\}$$

- each value must appear exactly once:
  - in each row of the table:

$$\sum_{j=1}^{9} x_{i,j,k} = 1 \qquad \forall i, k \in \{1, \cdots, 9\}$$

  - in each column of the table:

$$\sum_{i=1}^{9} x_{i,j,k} = 1 \qquad \forall j, k \in \{1, \cdots, 9\}$$

  - in each sub-table (indices of the sub-tables $m \in \{1, 2, 3\}$, $n \in \{1, 2, 3\}$):

$$\sum_{i=3m-2}^{3m} \sum_{j=3n-2}^{3n} x_{i,j,k} = 1 \qquad \forall k \in \{1, \cdots, 9\}, \forall m, n \in \{1, 2, 3\}$$

A funny example - Sudoku

Let's visualize the last set of constraints:

$$\sum_{i=3m-2}^{3m} \sum_{j=3n-2}^{3n} x_{i,j,k} = 1 \qquad \forall k \in \{1, \cdots, 9\}, \forall m, n \in \{1, 2, 3\}$$

- m=1,n=1:
$$\sum_{i=1}^{3} \sum_{j=1}^{3} x_{i,j,k} = 1 \qquad \forall k \in \{1, \cdots, 9\}$$
- m=1,n=2:
$$\sum_{i=1}^{3} \sum_{j=4}^{6} x_{i,j,k} = 1 \qquad \forall k \in \{1, \cdots, 9\}$$
- m=2,n=1:
$$\sum_{i=4}^{6} \sum_{j=1}^{3} x_{i,j,k} = 1 \qquad \forall k \in \{1, \cdots, 9\}$$
- ...

A funny example - Sudoku

Since Sudoku has no objective function (we are only looking for a feasible solution), we can choose any objective function:

$$
\begin{aligned}
&\min z = 0 \\
&s.t \sum_{k=1}^{9} x_{i,j,k} = 1 && \forall i, j \in \{1, \cdots, 9\} \\
&\sum_{j=1}^{9} x_{i,j,k} = 1 && \forall i, k \in \{1, \cdots, 9\} \\
&\sum_{i=1}^{9} x_{i,j,k} = 1 && \forall j, k \in \{1, \cdots, 9\} \\
&\sum_{i=3m-2}^{3m} \sum_{j=3n-2}^{3n} x_{i,j,k} = 1 && \forall k \in \{1, \cdots, 9\}, \forall m, n \in \{1, 2, 3\}
\end{aligned}
$$