# Problemi ben posti

Ricerca Operativa [035IN]

Lorenzo Castelli

9 Novembre 2021

UNIVERSITÀ DEGLI STUDI DI TRIESTE

# Setting the context

A natural starting point in solving linear integer programs

$$(IP) \quad \max\{cx : Ax \leq b, x \in \mathbb{Z}^n_+\}$$

with integral data $(A, b)$ is to ask when one will be so lucky that the linear programming relaxation

$$(LP) \quad \max\{cx : Ax \leq b, x \in \mathbb{R}^n_+\}$$

will have an optimal solution that is integral.

# Total unimodularity

**Definition**

A matrix $A$ is totally unimodular (TU) if every square submatrix of $A$ has determinant $+1$, $-1$ or $0$

**Observation**

If $A$ is TU, $a_{ij} \in \{+1, -1, 0\}$

# Examples

Matrices not TU

$$A_1 = \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}$$

$$|A_1| = 2$$

$$A_2 = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$$

$$|A_2| = 2$$

TU matrix

$$A_3 = \begin{pmatrix} 1 & -1 & -1 & 0 \\ -1 & 0 & 0 & 1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$$|A_3| = 0$$

**Proposition 1**

A matrix $A$ is TU if and only if

- the transpose matrix $A^T$ is TU
- the matrix $[A|I]$ is TU

**Proposition 2 (Sufficient condition)**

A matrix $A$ is TU if

(i) $a_{ij} \in \{+1, -1, 0\}$

(ii) Each column contains at most two nonzero coefficients, i.e, $\sum_{i=1}^{m} |a_{ij}| \leq 2$

(iii) There is a partition $(M_1, M_2)$ of the $M$ rows such that each column $j$ containing two nonzero coefficients satisfies $\sum_{i \in M_1} a_{ij} - \sum_{i \in M_2} a_{ij} = 0$.

Condition $(iii)$ means that if the nonzeros are in row $i$ and $k$, and if $a_{ij} = -a_{kj}$, then $\{i, k\} \in M_1$ or $\{i, k\} \in M_2$, whereas if $a_{ij} = a_{kj}, i \in M_1$ and $k \in M_2$ or vice versa.

# Example - TU matrix

$$A = \begin{array}{c} \left[\begin{array}{ccccccc} -1 & 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 \end{array}\right] \end{array} \begin{array}{l} row\,1 \\ row\,2 \\ row\,3 \\ row\,4 \\ row\,5 \end{array}$$

1. rows 1 and 3 are not in the same class
2. rows 2 and 3 are not in the same class
3. rows 1 and 4 are not in the same class
4. rows 2 and 5 are not in the same class
5. rows 1 and 2 are in the same class
6. rows 4 and 5 are in the same class
7. rows 2 and 4 are not in the same class

Hence $M_1 = \{1, 2\}$ and $M_2 = \{3, 4, 5\}$

# Examples - TU matrices

Condition is sufficient

$$\begin{pmatrix} 1 & -1 & -1 & 0 \\ -1 & 0 & 0 & 1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

1. rows 1 and 2 are in the same class
2. rows 1 and 3 are in the same class
3. rows 1 and 4 are in the same class
4. rows 2 and 3 are in the same class

$M_1 = \{1, 2, 3, 4\}$, $M_2 = \emptyset$

Condition is not necessary

$$\begin{pmatrix} 1 & 0 & 1 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & -1 & -1 & 0 \\ 0 & -1 & -1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

**Proposition 3**

The linear programming problem $\max\{cx : Ax \leq b, x \in \mathbb{R}^n_+\}$ has an integer optimal solution for all integer vectors $b$ for which it has a finite optimal value if and only if $A$ is totally unimodular.

# Minimum Cost Network Flow

Given a digraph $D = (V, A)$ with arc capacities $h_{ij}$ for all $(i, j) \in A$, demands $b_i$ (positive inflows or negative outflows) at each node $i \in V$, and unit flow costs $c_{ij}$ for all $(i, j) \in A$, the minimum cost network flow problem is to find a feasible flow that satisfies all the demands at minimum cost. This has the formulation

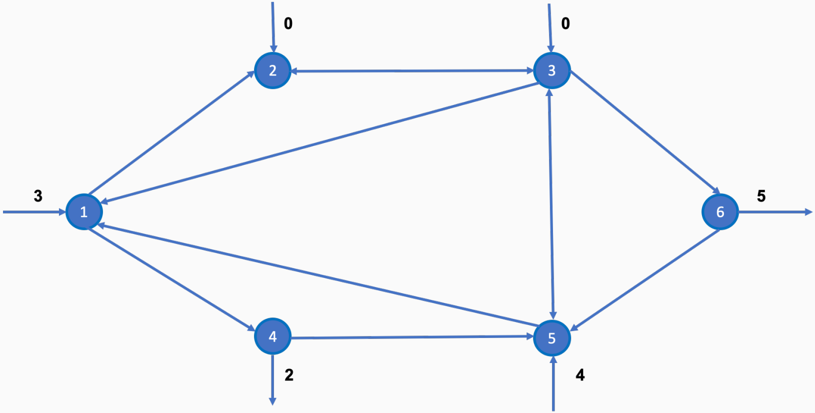
$$\min \sum_{(i,j)\in A} c_{ij} x_{ij} \tag{1}$$

$$\sum_{k\in V^+(i)} x_{ik} - \sum_{k\in V^-(i)} x_{ki} = b_i \text{ for } i \in V \tag{2}$$

$$0 \leq x_{ij} \leq h_{ij} \text{ for } (i, j) \in A \tag{3}$$

where $x_{ij}$ denotes the flow in arc $(i, j), V^+(i) = \{k : (i, k) \in A\}$ and $V^-(i) = \{k : (k, i) \in A\}$

# Minimum Cost Network Flow



# Minimum Cost Network Flow

| $x_{12}$ | $x_{14}$ | $x_{23}$ | $x_{31}$ | $x_{32}$ | $x_{35}$ | $x_{36}$ | $x_{45}$ | $x_{51}$ | $x_{53}$ | $x_{65}$ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | -1 | 0 | 0 | 0 | 0 | -1 | 0 | 0 | = | 3 |
| -1 | 0 | 1 | 0 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | = | 0 |
| 0 | 0 | -1 | 1 | 1 | 1 | 1 | 0 | 0 | -1 | 0 | = | 0 |
| 0 | -1 | 0 | 0 | 0 | 0 | 0 | -1 | 0 | 0 | 0 | = | -2 |
| 0 | 0 | 0 | 0 | 0 | -1 | 0 | -1 | 1 | 1 | -1 | = | 4 |
| 0 | 0 | 0 | 0 | 0 | 0 | -1 | 0 | 0 | 0 | 1 | = | -5 |

The additional constraints are the capacity constraints $0 \leq x_{ij} \leq h_{ij}$

**Proposition 4**

The constraint matrix $A$ arising in a minimum cost network flow problem is totally unimodular.

**Proof** The matrix $A$ is of the form $\begin{pmatrix} C \\ I \end{pmatrix}$ where $C$ comes from the flow conservation constraints and $I$ from the capacity constraints. Therefore it suffices to show that $C$ is TU. The sufficient conditions of Proposition 2 are satisfied with $M_1 = M$ and $M_2 = \emptyset$.

**Corollary**

In a minimum cost network flow problem, if the demands $\{b_i\}$ and the capacities $\{h_{ij}\}$ are integral

- Each extreme point is integral.
- The constraints (2) and (3) describe the convex hull of the integral feasible flows.

This corollary means that the linear relaxation of the minimum cost network flow problem always provides an integer solution provided that all capacities $\{h_{ij}\}$ and demands $\{b_i\}$ are integral.

# Special minimum cost flows

The Shortest Path Problem Given a digraph $D = (V, A)$, two distinguished nodes $s, t \in V$, and non-negative arcs costs $c_{ij}$ for $(i, j) \in A$, find a minimum cost $s - t$ path.

The Max Flow Problem Given a digraph $D = (V, A)$, two distinguished nodes $s, t \in V$, and non-negative capacities $h_{ij}$ for $(i, j) \in A$, find a maximum flow from $s$ to $t$ path.

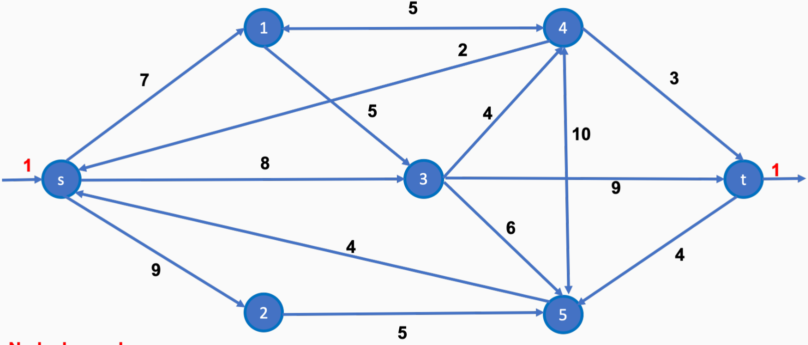
The Transportation Problem Let there be $m$ suppliers and $n$ consumers. The $i$th supplier can provide $a_i$ units of a certain good and the $j$th consumer has a demand for $b_j$ units. If $c_{ij}$ is the cost to transport one unit of good from the $i$th supplier to the $j$th consumer, the problem is to transport the goods from the suppliers to the consumers at minimum cost.

The Assignment Problem It is a special case of the transportation problem, where the number of suppliers is equal to the number of consumers, each supplier has unit supply, and each consumer has unit demand.

# The Shortest Path Problem

If we set $b_s = 1$ and $b_t = -1$, only one unit of flow can move from $s$ to $t$, and the problem is to find the sequence of arcs at minimum cost that this unit will traverse. An arc $(i, j) \in A$ if and only if $h_{ij} > 0$. Since we assume only integral values, $(i, j) \in A$ if and only if $h_{ij} \geq 1$. Since exactly one unit flows in the network, there is no need to explicitly include the capacity constraints.

Decision variables are such that $x_{ij} = 1$ if arc $(i, j)$ is in the minimum cost (shortest) $s - t$ path. For the total unimodularity, an optimal solution is always integer. Therefore, we can write $x_{ij} \geq 0$.

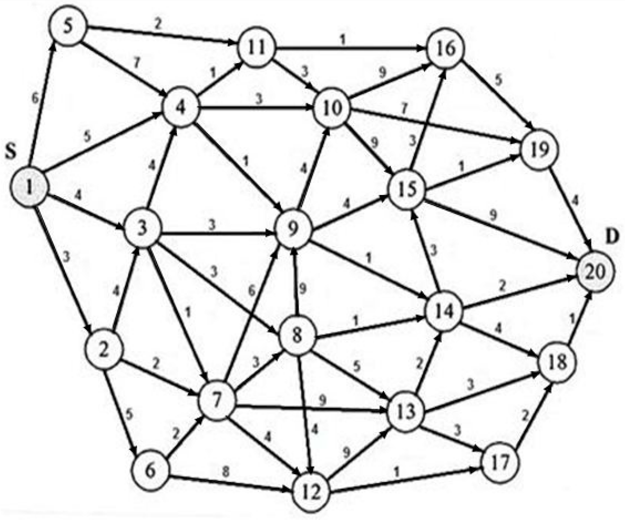



Node demand

Arc cost

# The Shortest Path Problem

$$
\begin{aligned}
z = \min \sum_{(i,j)\in A} c_{ij}x_{ij} & & \\
\sum_{k\in V^+(i)} x_{ik} - \sum_{k\in V^-(i)} x_{ki} = 1 & & \text{for } i = s \\
\sum_{k\in V^+(i)} x_{ik} - \sum_{k\in V^-(i)} x_{ki} = 0 & & \text{for } i \in V\backslash\{s,t\} \\
\sum_{k\in V^+(i)} x_{ik} - \sum_{k\in V^-(i)} x_{ki} = -1 & & \text{for } i = t \\
x_{ij} \geq 0 \text{ for } (i,j) \in A & &
\end{aligned}
$$

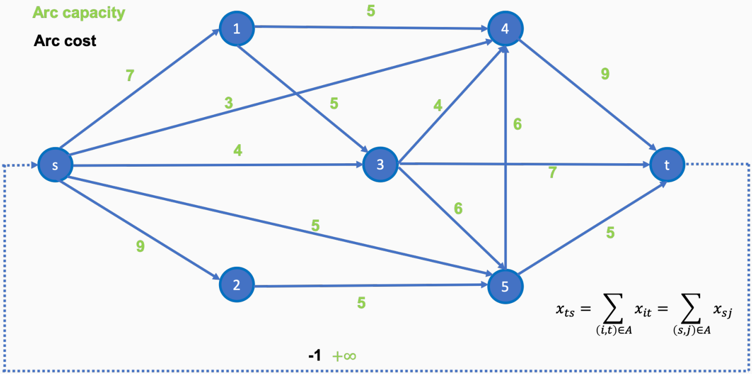

S
D
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20

Adding a backward arc from $t$ to $s$, the maximum $s - t$ flow problem can be formulated as

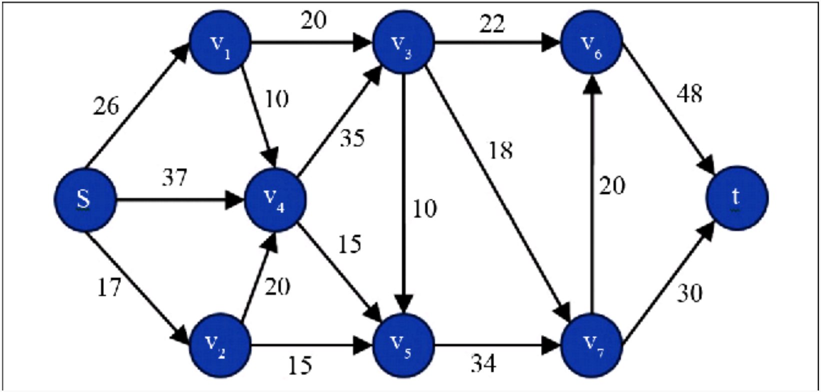
$$
\begin{aligned}
z = \max x_{ts} & \\
\sum_{k \in V^+(i)} x_{ik} - \sum_{k \in V^-(i)} x_{ki} = 0 & \qquad \text{for } i \in V \\
0 \leq x_{ij} \leq h_{ij} & \qquad \text{for } (i,j) \in A
\end{aligned}
$$

For the total unimodularity, an optimal solution is integer provided that all capacities $h_{ij}$ are integral.

# The Maximum Flow Problem

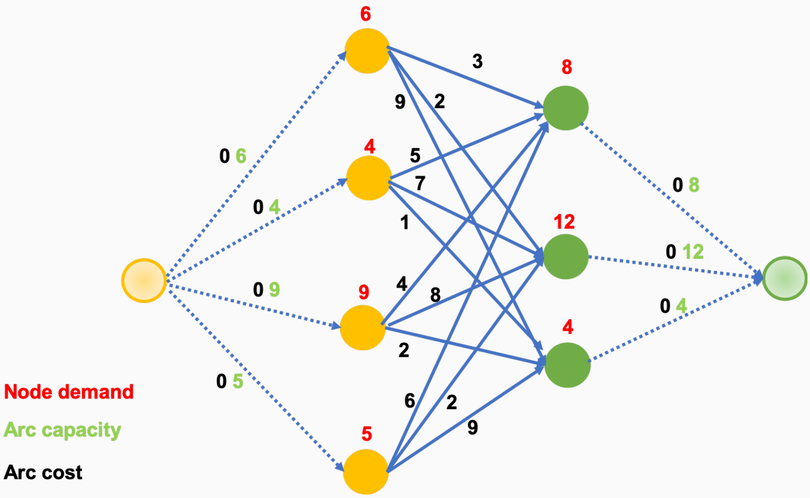



$$x_{ts} = \sum_{(i,t)\in A} x_{it} = \sum_{(s,j)\in A} x_{sj}$$


S
v1
v2
v3
v4
v5
v6
v7
t
26
37
17
20
10
20
15
35
15
10
22
18
34
20
48
30

It can be formulated as a minimum cost flow problem on a bipartite graph $D = (V_1 \cup V_2, A)$ where $V_1 = \{1, \dots, m\}$ is the set of sources, $V_2 = \{1, \dots, n\}$ is the set of sinks, and $A = \{(i, j) : i \in V_1, j \in V_2\}$. Without loss of generality, we assume there is an arc from each supply node to each demand node. The unit shipping cost from $i \in V_1$ to $j \in V_2$ is $c_{ij}$. If there is no arc from $i$ to $j$, we take $c_{ij}$ very large. Node $i \in V_1$ has a positive integral supply $a_i$ and $j \in V_2$ has a positive integral demand $b_j$. The flow out of a source is required to equal its supply, and the flow into a sink must equal its demand. Thus a necessary condition for feasibility is $\sum_{i \in V_1} a_i = \sum_{j \in V_2} b_j$

# The Transportation Problem

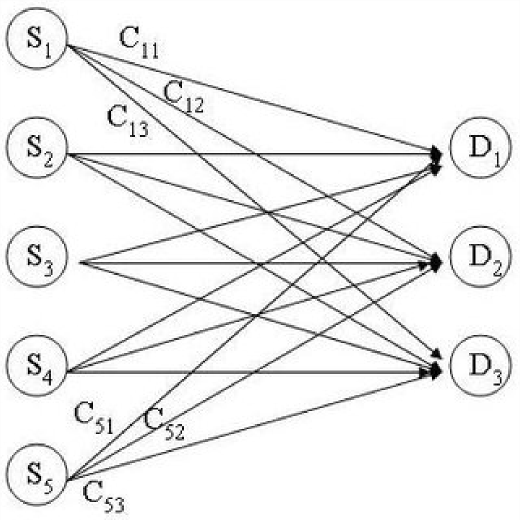


$$z = \min \sum_{i \in V_1} \sum_{j \in V_2} c_{ij} x_{ij}$$

$$\sum_{j \in V_2} x_{ij} = a_i \qquad \text{for } i \in V_1$$

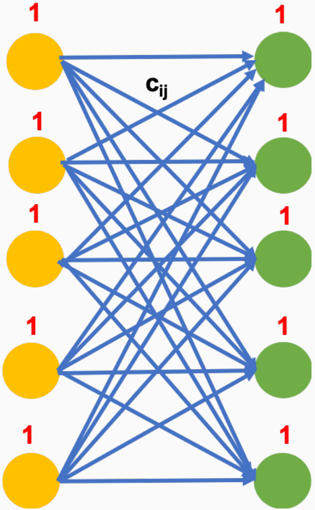
$$\sum_{i \in V_1} x_{ij} = b_j \qquad \text{for } j \in V_2$$

$$x_{ij} \geq 0 \qquad \text{for } (i, j) \in A$$


$S_1$
$S_2$
$S_3$
$S_4$
$S_5$
$D_1$
$D_2$
$D_3$
$C_{11}$
$C_{12}$
$C_{13}$
$C_{51}$
$C_{52}$
$C_{53}$

When $a_i = b_j = 1$ for all $i$ and $j$ and $m = n$, we have the assignment problem

$$
\begin{aligned}
z = \min \sum_{i \in V_1} \sum_{j \in V_2} c_{ij} x_{ij} & \\
\sum_{j \in V_2} x_{ij} = 1 & \qquad \text{for } i \in V_1 \\
\sum_{i \in V_1} x_{ij} = 1 & \qquad \text{for } j \in V_2 \\
0 \leq x_{ij} \leq 1 & \qquad \text{for } (i, j) \in A
\end{aligned}
$$


1
1
1
1
1
1
1
1
1
1
$c_{ij}$

Suppose there are $n$ people and $m$ jobs, where $n \geq m$. Each job must be done by exactly one person; also, each person can do, at most, one job. The cost of person $j$ doing job $i$ is $c_{ij}$. The problem is to assign the people to the jobs so as to minimise the total cost of completing all of the jobs. To formulate this problem, which is known as the assignment problem, we introduce 0-1 variables $x_{ij}, i = 1, \ldots, m, j = 1, \ldots, n$ corresponding to the $ij$th event of assigning person $j$ to job $i$.

Since exactly one person must do job $i$, we have the constraints

$$\sum_{j=1}^{n} x_{ij} = 1 \quad \text{for } i = 1, \dots, m \tag{4}$$

Since each person can do no more than one job, we also have the constraints

$$\sum_{i=1}^{m} x_{ij} \leq 1 \quad \text{for } j = 1, \dots, n \tag{5}$$

It is now easy to check that if $x \in \{0,1\}^{mn}$ satisfies (4) and (5), we obtain a feasible solution to the assignment problem. The objective function is

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

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

| | 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 |

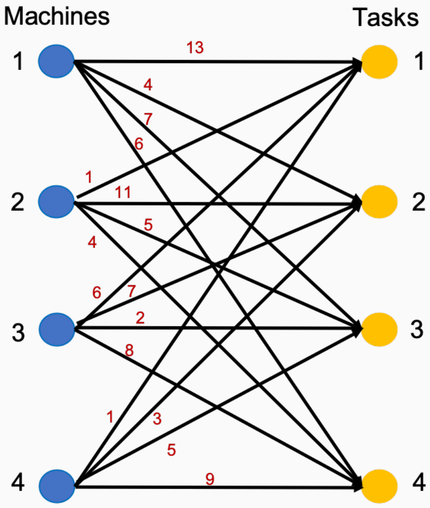
The company wants to minimise the total setup time needed for the processing of all four tasks.

If we think of the setup times as costs and define

$$x_{ij} = \begin{cases} 1 & \text{if machine } i \text{ is assigned to process task } j \\ 0 & \text{if machine } i \text{ is not assigned to process task } j \end{cases}$$

where $i = 1, 2, 3, 4$ and $j = 1, 2, 3, 4$, then it is easily seen that what we have is a problem with 4 sources (representing the machines), 4 sinks (representing the tasks), a single unit of supply from each source (representing the availability of a machine), and a single unit of demand at each sink (representing the processing requirement of a task). This particular class of problems is called the assignment problems.

# The Assignment Problem - Example (III)



# The Assignment problem - Example (IV)

$$
\begin{aligned}
\min 13x_{11} + 4x_{12} + 7x_{13} + 6x_{14} + \\
x_{21} + 11x_{22} + 5x_{23} + 4x_{24} + \\
6x_{31} + 7x_{32} + 2x_{33} + 8x_{34} + \\
x_{41} + 3x_{42} + 5x_{43} + 9x_{44}
\end{aligned}
$$

$$
\begin{aligned}
x_{11} + x_{12} + x_{13} + x_{14} = 1 \\
x_{21} + x_{22} + x_{23} + x_{24} = 1 \\
x_{31} + x_{32} + x_{33} + x_{34} = 1 \\
x_{41} + x_{42} + x_{43} + x_{44} = 1
\end{aligned}
$$

$$
\begin{aligned}
x_{11} + x_{21} + x_{31} + x_{41} = 1 \\
x_{12} + x_{22} + x_{32} + x_{42} = 1 \\
x_{13} + x_{23} + x_{33} + x_{43} = 1 \\
x_{14} + x_{24} + x_{34} + x_{44} = 1
\end{aligned}
$$

$x_{ij} \in \{0, 1\}$, for $i = 1, \ldots, 4, j = 1, \ldots, 4$

Thank you for your attention