Tutorium #10

20/01/2023

1601. Maximum Number of Achievable Transfer Requests

Hint

Hard 265 28

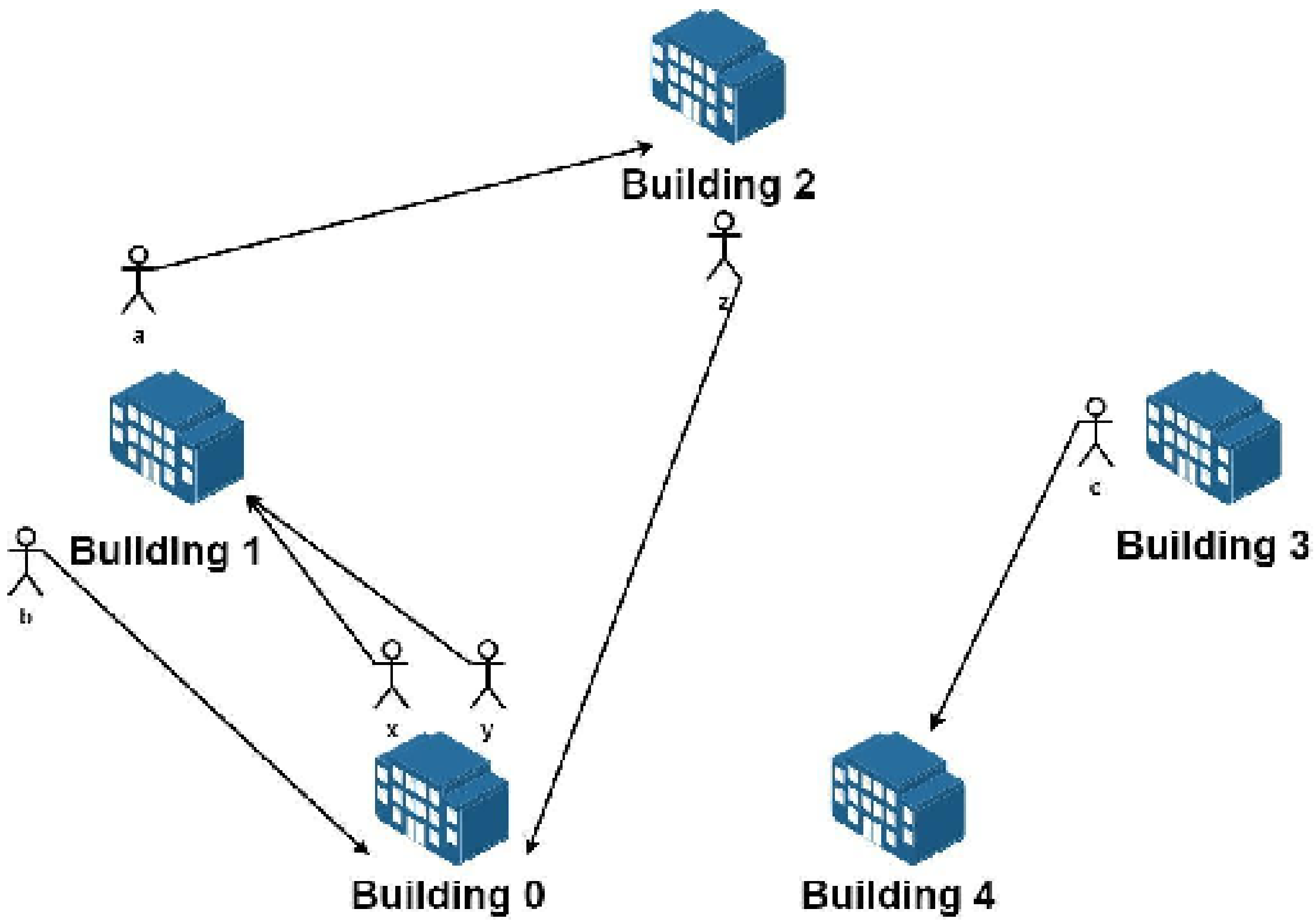
Companies

We have `n` buildings numbered from `0` to `n - 1`. Each building has a number of employees. It's transfer season, and some employees want to change the building they reside in.

You are given an array `requests` where $requests[i] = [from_i, to_i]$ represents an employee's request to transfer from building $from_i$ to building to_i.

All buildings are full, so a list of requests is achievable only if for each building, the **net change in employee transfers is zero**. This means the number of employees **leaving is equal** to the number of employees **moving in**. For example if `n = 3` and two employees are leaving building `0`, one is leaving building `1`, and one is leaving building `2`, there should be two employees moving to building `0`, one employee moving to building `1`, and one employee moving to building `2`.

Return *the maximum number of achievable requests*.




```
Input: n = 5, requests = [[0,1],[1,0],[0,1],[1,2],[2,0],[3,4]]
Output: 5
Explantion: Let's see the requests:
From building 0 we have employees x and y and both want to move to building 1.
From building 1 we have employees a and b and they want to move to buildings 2 and 0 respectively.
From building 2 we have employee z and they want to move to building 0.
From building 3 we have employee c and they want to move to building 4.
From building 4 we don't have any requests.
We can achieve the requests of users x and b by swapping their places.
We can achieve the requests of users y, a and z by swapping the places in the 3 buildings.
```

$$\max \sum_{i=0}^{n} x_i$$

$$\text{st. } \forall i \in \{0, \ldots, n-1\} : \sum_{\substack{j=0 \\ \text{requests[j][1]=i}}}^{n} x_i - \sum_{\substack{j=0 \\ \text{requests[j][0]=i}}}^{n} x_i = 0$$

$$x \in \{0,1\}^n$$

or st. $Ax = 0$ with $A \in \{0, \pm 1\}^{m \times n}$

$\Rightarrow$ A is an incidence matrix

$\Rightarrow$ A is total unimodular $\boxed{\Rightarrow \text{LP}}$

Problem 1 (8 points)

Formulate the following partitioning problem as integer linear program (ILP): Given is a set $M = \{1, \ldots, m\}$, n subsets $M_i \subseteq M$, $1 \leq i \leq n$ and costs c_i for sets M_i. Problem: Find a set $F \subseteq \{1, \ldots, n\}$, such that $\bigcup_{i \in F} M_i = \{1, \ldots, m\}$, $\forall i, j \in F : M_i \cap M_j = \emptyset$ and the overall cost of the sets in F are minimized. If there is no solution, then the ILP shall be infeasible. Explain your solution in detail.

Solution:

Let x_j be a 0-1-variable with $x_j = 1$ iff $j \in F$, $1 \leq j \leq n$. Let $H_i = \{j | i \in M_j\}$ be the set of all M_j, that cover the i's. The ILP is then given as

minimize $\sum_{j=1}^{n} c_j x_j$
subject to $\sum_{j \in H_i} x_j = 1$ for $i \in \{1, \ldots, m\}$
$x_j \in \{0, 1\}$ for $j \in \{1, \ldots, n\}$

Problem 2 (8 points)

In this exercise you are asked to formulate the *maximum st-flow problem* as a linear program.

The input to the maximum st-flow problem is a directed graph $G = (V, E)$, two special vertices $s, t \in V$, and a function assigning a non-negative capacity c_{uv} to every edge $(u, v) \in E$.

A *flow* f assigns to every edge in E a non-negative flow value, i.e., $f : E \to \mathbb{R}$. To simplify the notation we use $f(u, v)$ to denote the flow on edge (u, v) (instead of $f((u, v))$). The flow *into a node* v is

$$\sum_{(u,v)\in E} f(u, v),$$

the flow *out of node* v is

$$\sum_{(v,x)\in E} f(v, x).$$

The *net-flow of a node* v is defined as

$$\sum_{(v,x)\in E} f(v, x) - \sum_{(u,v)\in E} f(u, v),$$

i.e., it is the difference between the flow out of v and the flow into v.

In a *feasible flow*

- the flow on each edge is at most its capacity, and
- for every vertex v except s and t, i.e., $v \in V \setminus \{s, t\}$, the flow into the vertex (i.e., over the incoming edges of this vertex) is the same as the flow out of the vertex, i.e., the net-flow of v is 0.

The goal is to maximize the net-flow of s.

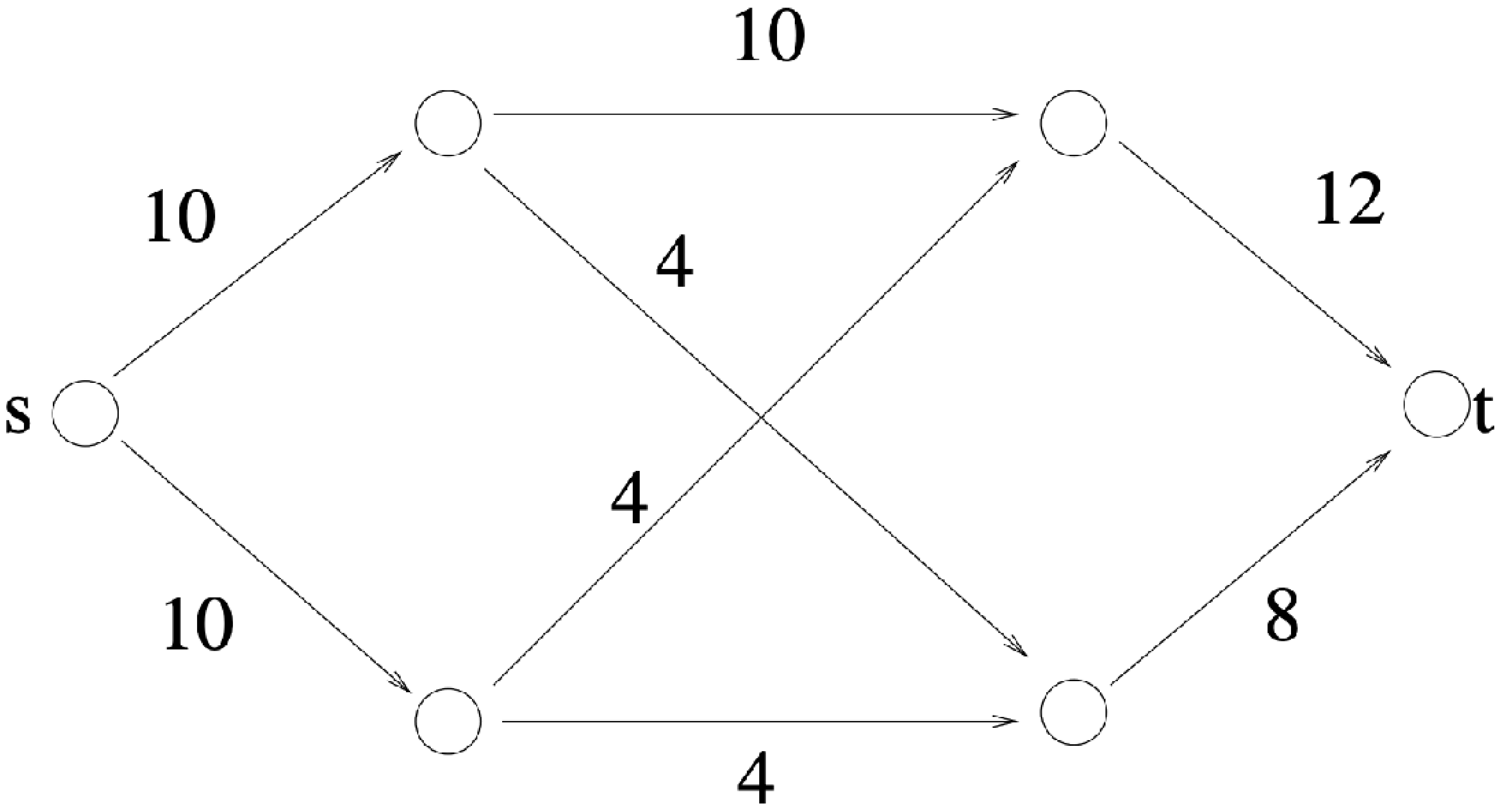
Formulate a linear program for the maximum st-flow problem (it does *not* have to be in canonical form). Clearly define the variables of your linear program, and then give the function that is maximized and the constraints.

Solution:
Optimize the following objective:

$$\max \left(\sum_{(s,x)\in E} f(s,x) - \sum_{(u,s)\in E} f(u,s) \right)$$

Subject to the following constraints:

$$\begin{aligned}
\forall (u,v) \in E: \quad & f(u,v) && \leq c_{uv} \\
\forall v \in V \setminus \{s,t\}: \quad & \textstyle\sum_{(v,x)\in E} f(v,x) - \sum_{(u,v)\in E} f(u,v)) && = 0 \\
\forall (u,v) \in E: \quad & f(u,v) && \geq 0
\end{aligned}$$



10
10
4
12
s
t
4
10
8
4

Problem 3 (4 points)

Given a graph $G = (V, E)$ the graph bipartitioning problem asks for a partition of the vertices into two blocks $V = V_1 \cup V_2$ with $V_1 \cap V_2 = \emptyset$ and $|V_i| \leq \lceil |V|/2 \rceil$. The objective is to minimize the number of edges that run between the two blocks V_1 and V_2. Give an ILP that solves the problem.

Solution:
Optimize the following objective:

$$\min \sum_{(i,j)\in E} y_{ij}$$

Subject to the following constraints:

$$\begin{aligned}
& \sum_{i\in V} x_i && \leq \lceil |V|/2 \rceil \\
& \sum_{i\in V} x_i && \geq \lfloor |V|/2 \rfloor \\
\forall (i,j) \in E: \quad & y_{ij} && \geq x_j - x_i \\
\forall (i,j) \in E: \quad & y_{ij} && \geq x_i - x_j \\
\forall i \in V: \quad & x_i && \in \{0, 1\} \\
\forall (ij) \in E: \quad & y_{ij} && \in \{0, 1\}
\end{aligned}$$