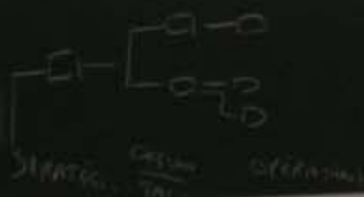
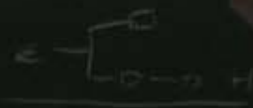


## outline

- 1 introduction
- 2 scheduling problem
- 3 compatible set problem
- 4 matching inequalities
- 5 numerical study
- 6 conclusions



VALUES





Outline

- Introduction
- Learning objectives
- Competences and outcomes
- Assessment and quality
- Academic work
- Assessment



## transmission scheduling problem

- $C_i, i \in I$  – a given list of compatible sets
- binary constants  $c_{ai}$  – arc  $a$  is in compatible set  $i$
- $n(a)$  – number of routes through arc  $a$
- binary variables  $u_{it}$ ;  $u_{it} = 1$  iff set  $i$  is used in time slot  $t$

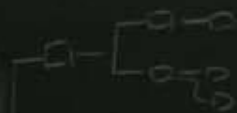
### integer program

maximize  $f$

$$\sum_{i \in I} u_{it} = 1, t \in T$$

$$C_a = B \cdot T \cdot \sum_{i \in I} \sum_{t \in T} c_{ai} u_{it}, a \in A$$

$$n(a) \cdot f \leq C_a, a \in A$$

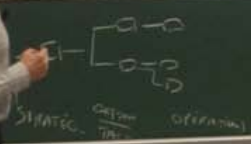


# linear relaxation

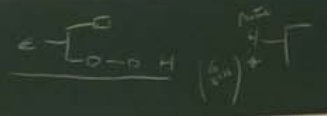
$$\begin{aligned} & \text{maximize } f \\ & \sum_{i \in I} u_i = 1, i \in I; \quad C_a = B \cdot T \cdot \sum_{i \in I} \sum_{j \in I} C_{ij} u_i, a \in A \\ & n(a) \cdot f \leq C_a, a \in A; \quad 0 \leq u_i \leq 1, i \in I, I \in T \end{aligned}$$

$$\begin{aligned} & \text{maximize } f \\ & \sum_{i \in I} x_i = T \cdot T \\ & [ \pi_a \geq 0 ] \quad n(a) \cdot f \leq B \cdot \sum_{i \in I} C_{ij} x_i, a \in A \\ & x_i \geq 0, i \in I \end{aligned}$$

where:  $x_i = T \cdot \sum_{j \in I} u_j$       $\pi_a$  - dual variables



VALUES

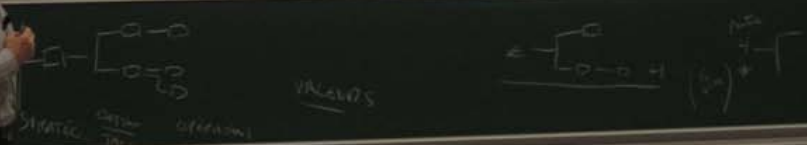


## linear relaxation

$$\begin{aligned} & \text{maximize } f \\ & \sum_{i \in I} u_{it} = 1, t \in T; \quad C_a = B \cdot \tau \cdot \sum_{i \in I} \sum_{t \in T} c_{ai} u_{it}, a \in A \\ & n(a) \cdot f \leq C_a, a \in A; \quad 0 \leq u_{it} \leq 1, i \in I, t \in T \end{aligned}$$

$$\begin{aligned} & \text{maximize } f \\ & \sum_{i \in I} x_i = T \cdot \tau \\ & [\pi_a \geq 0] \quad n(a) \cdot f \leq B \cdot \sum_{i \in I} c_{ai} x_i, a \in A \\ & x_i \geq 0, i \in I \end{aligned}$$

• where:  $x_i = \tau \cdot \sum_{t \in T} u_{it}$        $\pi_a$  - dual variables



## Strong relaxation

Assumption 1

$$\sum_{i \in I} \mu_i = 1, \quad \mu_i \geq 0, \quad \mu_i = 0 \Rightarrow \sum_{j \in J} \mu_{ij} = 0, \quad \mu_{ij} \geq 0, \quad \mu_{ij} = 0 \Rightarrow \mu_i = 0$$

Assumption 2

$$\sum_{i \in I} \mu_i = 1, \quad \mu_i \geq 0, \quad \mu_i = 0 \Rightarrow \sum_{j \in J} \mu_{ij} = 0, \quad \mu_{ij} \geq 0, \quad \mu_{ij} = 0 \Rightarrow \mu_i = 0$$

\* where:  $\mu_i = \sum_{j \in J} \mu_{ij}$ ,  $\mu_{ij} = \mu_i \mu_{ij}$





Generative text networks

- autoregressive generation (e.g. GPT-2)
- self-supervised learning (e.g. BERT)
- GPT-3 (text completion)
- applications: text generation, summarization, translation, etc.





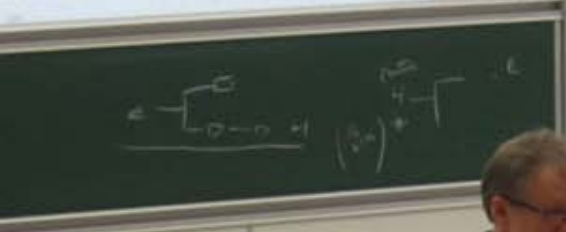
Handwritten text on the chalkboard, including the Greek letter  $\epsilon$  and some illegible symbols.

Two electrical outlets on the wall.

A red fire alarm pull station on the wall.

A plastic water bottle on a desk.

scheduling and channel  
oc Networks 8 (2010)  
compatible sets, MWCSP)  
n Fair Flow Optimization in  
Ad Hoc Networks 1 (2011)  
compatible sets)  
computational approach for  
wireless networks under the  
Wireless Commun. 10 (2011)  
for MWCSP)





Integer programming consists of maximizing  
or minimizing a linear function, subject to  
linear constraints. It is a special case of  
linear programming. It is NP-hard.  
Applications: scheduling, assignment,  
facility location, transportation, etc.  
The branch-and-bound algorithm is the  
most common method for solving integer  
programming problems. It is based on  
branching the problem into smaller  
problems and bounding the objective  
function.

