4

Loot distribution

k loupežníků ⇒ k skrýší

Alibaba and his k men have raided an ancient tomb. Inside, there were n treasures $T = \{t_1, \dots, t_n\}$. Each treasure t_i has some weight $w_i \in \mathbb{R}^+$ and price $p_i \in \mathbb{R}^+$. Each treasure belongs to one of four groups $G = \{1, 2, 3, 4\}$, where $g_i \in G$ denotes the group of treasure t_i. Each of Alibaba's k men will move part of the loot to one of the Alibaba's hideouts. The maximal weight that man $j \in \{1, 2, \dots, k\}$ can carry is $u_j \in \mathbb{R}^+$.

Alibaba wants to:

- a) transport at least one treasure from group 1 and at least one treasure from group 2 to any of his hideouts;
- b) transport any treasure from group 4 ($trans_4$) only if at least some treasure from group 3 is being transported ($trans_3$) to any of the hideouts;
- c) distribute the loot such that the maximal difference in the total prices transported by the individual men is at most $P_{max} \geq \max\{p_i\} - \min\{p_i\}$; $\forall z$ k nese p_i
- d) maximize the total price of the transferred treasures

Design an ILP model to solve the given loot distribution problem.

$t_i \Rightarrow g_i$

předpočítám si binární vektory q: $q_{i,1}; q_{i,2}, q_{i,3}, q_{i,4}$ ... ith treasure patří skupině 1,2,3 nebo 4.

Vars:

~~$m = [m_1, \dots, m_k] \in \{0,1\}^k$ ... will ith man move~~

$m^{z,i} \in \{0,1\}^{k \times n}$ ... k-th man will move n-th part

~~$trans_4, trans_3 \in \{1,0\}$~~

$y, x \in \{0,1\}$

~~Param:~~

Constr:

~~a) $\sum_{z \in k} \sum_{i \in n} (g_i = 1) \cdot m^{z,i} \geq 1$~~

max. weight

$$\sum_{i=1..n} m^{z,i} \cdot w_i \leq u_z, \forall z \in 1..$$

✓ 1.6

$$a) \sum_{z=1..k} \sum_{i=1..n} q_{i,1} \cdot m^{z,i} \geq 1$$ ✓

✓ 1.6

$$\sum_{z=1..k} \sum_{i=1..n} q_{i,2} \cdot m^{z,i} \geq 1$$

$$b) \sum_{z} \sum_{i} q_{i,4} \cdot m^{z,i} \Big/ \geq \cancel{trans_4}\ 1 \; +y \cdot M ; \quad \sum_{z} \sum_{i} q_{i,3}\, m^{z,i} \Big/ \geq \cancel{trans_3}\ 1 \; +x \cdot M$$

~~$trans_4 = trans_3$~~ $\quad y = x$

0 $\quad z = 1 \dots y = 1$ ale musí nést G4 i bez ...

Objective

d) maximize $\sum_{z=1..k} \sum_{i=1..n} m^{z,i} \cdot p_i$ 1.6

pokračování 2. strana 1. list

Correctness of Dijkstra's Algorithm

Prove the correctness of Dijkstra's Algorithm

Input: digraph G, weights $c : E(G) \to \mathbb{R}_0^+$ and node $s \in V(G)$.

Output: Vectors l and p. For $v \in V(G)$, $l(v)$ is the length of the shortest path from s and $p(v)$ is the previous node in the path. If v is unreachable from s, $l(v) = \infty$ and $p(v)$ is undefined.

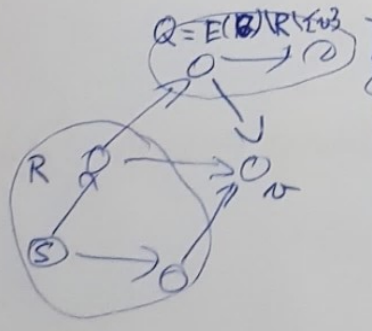
```
l(s) := 0; l(v) := ∞ for v ≠ s; R := ∅ ;
while R ≠ V(G) do
    Find v ∈ V(G) \ R such that l(v) = min_{w∈V(G)\R} l(w);
    R := R ∪ {v} ;
    // calculate l(w) for all nodes on border of R
    for w ∈ V(G) \ R such that (v, w) ∈ E(G) do
        if l(w) > l(v) + c(v, w) then
            l(w) := l(v) + c(v, w); p(w) := v;
        end
    end
end
```

Dokážeme indukcí:

První krok: Pokud $R_0 = \emptyset$, tak R obsahuje optimální vzdálenosti z s.

Indukční předpoklad: ~~Když R~~ R_n obsahuje pouze uzly jejichž vzdálenost od s je optimální.

Chci dokázat: Když ↗, tak R_{n+1} je také optimální.



R je optimální, tady platí Bellmanova rovnice $l(s, v) = \min_{w \in pred(v)} \{ l(s, w) + c(w, v) \}$

cena hrany (w, v)

pro předchůdce z Q: ~~[illegible] je [illegible]~~, protože hrany musí mít nezáporné c, tak cesta vedoucí přes uzly z Q, nebude optimální

$\Rightarrow R_{n+1} = R_n \cup \{v\}$ bude optimální.

6

Multiprocessor Scheduling problem with preemption, release date and deadline

We have n **tasks** which we want to assign to R **identical resources** (processors). Each task has its own **processing time** p_j, **release date** r_j and **deadline** $\tilde{d}_j$. **Preemption** is allowed (including migration from one resource to another). Every processor will execute no more than one task at a moment and no task will be executed simultaneously on more than one processor.

2 parallel identical resources:

M =

úkol	T_1	T_2	T_3	T_4	T_5
p_j	2.1	3.2	4.1	1.6	2
r_j	0	2	0	5	5
$\tilde{d}_j$	5	6	6	7	7

Formulate as Maximum flow problem.

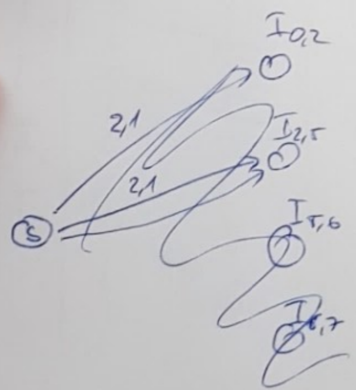
$R = 2$

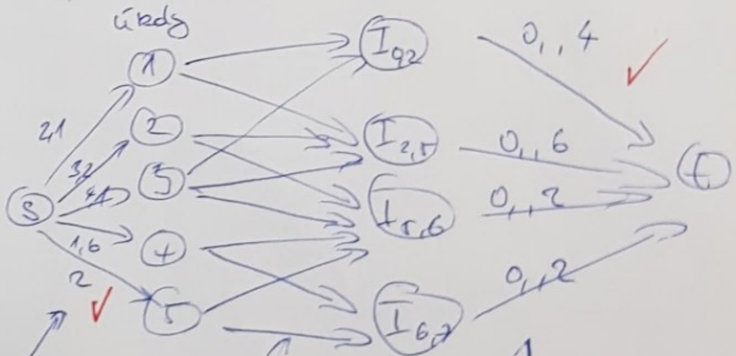
Důležité body určující intervaly: set$(M[r_j : d_j, :])$ =

$= \{0, 2, 5, 6, 7\}$ ✓

Intervaly: $\langle 0; 2)$, $\langle 2; 5)$, $\langle 5, 6)$, $\langle 6; 7)$ ✓

$I_{0,2}$, $I_{2,5}$, $I_{5,6}$, $I_{6,7}$





✓

stejný UB i LB

✓... feasible

opt. flow → schedule

✓

LB = 0; UB = vstupní tok do úkolu.

upper bounds: množství volných jednotek v intervalu. → délka I · R

Knapsack

Using dynamic programming, solve the following instance of Knapsack Problem:

- number of items: $n = 7$
- knapsack capacity: $W = 5$
- costs $\mathbf{c} = (2, 2, 2, 2, 4, 3, 1)$
- weights: $\mathbf{w} = (1, 1, 2, 2, 3, 4, 1)$

6b

a) Compute the optimal solution (objective value and items in knapsack) of this instance of Knapsack Problem. Write down all iterations of the algorithm. Is this solution unique and why?

b) What can you say about the computational complexity of the algorithm for instances where $W \leq 10n$?

možné předmět buďto vložit, nebo nevložit

beru maximum

item \ w	0	1	2	3	4	5
	0	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$
I_1	0	2	$-\infty$	$-\infty$	$-\infty$	$-\infty$
I_2	0	2	4	$-\infty$	$-\infty$	$-\infty$
I_3	0	2	4	4	6	$-\infty$
I_4	0	2	4	4	6	6
I_5	0	2	4	4	6	8
I_6	0	2	4	4	6	8
I_7	0	2	4	5	6	8

Když zde nepíšete konkrétní čísla, je nutné uvést, že opt. řešení nemusí být v pravém dolním rohu.

→ x

← maximální cena

a) řešení: ~~$\{I_1, I_2, I_5,$~~ $\{I_1, I_2, I_5\}$, $c_{opt} = 6$

Řešení je unikátní: existuje jen jedna cesta z $(I_1, 0)$ do optima

b) $O(n \cdot W) = O(n \cdot 10n) = O(n^2)$

Složitost bude záviset jen na počtu těsně, ale stále bude polynomiální.

Project Scheduling with Temporal Constraints

46

A construction company has obtained a contract on a project consisting of 4 non-preemptive activities A, B, C and D. The following temporal constraints apply:

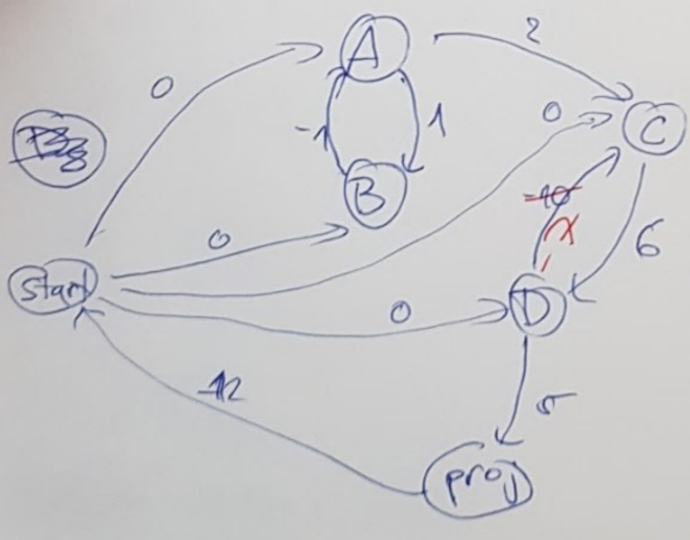
- a) Activity durations are $p_A = 1$, $p_B = 2$, $p_C = 3$ and $p_D = 2$.
- b) B must start exactly 1 time unit after start of A.
- c) C can start at least 2 time units after start of A.
- d) After the completion of C, there are at least 3 more time units before D can start.
- e) D can be started not later than 7 units after start of C.
- f) We suppose that the project can be terminated 3 time units after the completion time of D.
- g) The maximum duration of the project is 12 time units.

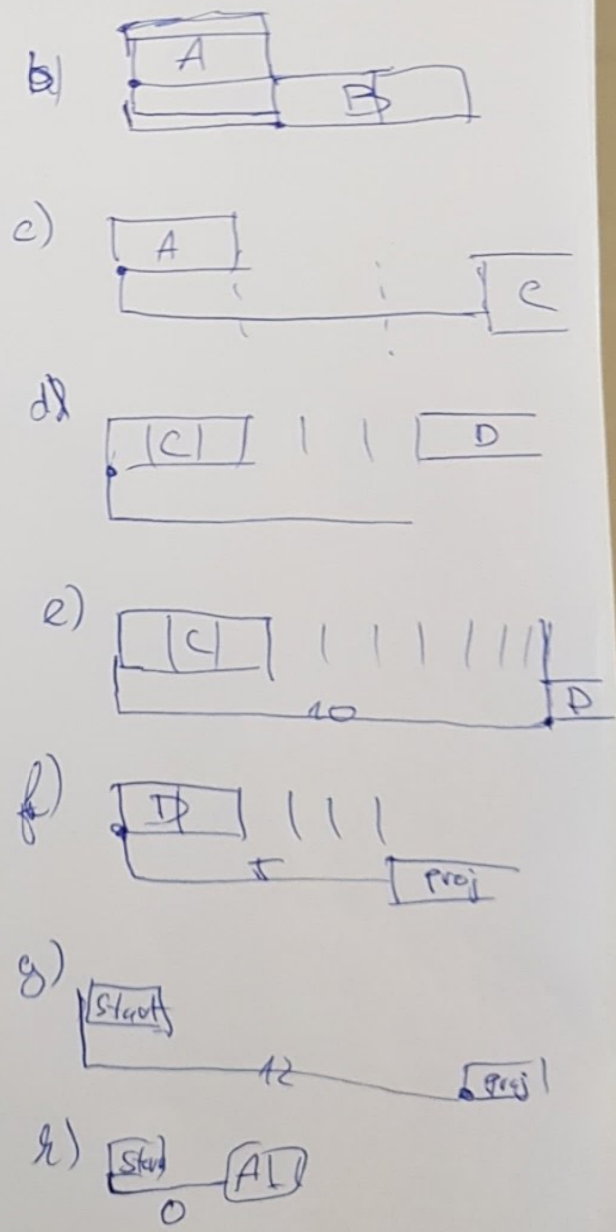
a) Draw a directed graph with temporal constraints for this problem.

b) Is this instance schedulable? Why? ✓

Hint: Add "dummy" activities with zero duration for beginning and end of the project. ~~B_s, D_e~~

	A	B	C	D
p_j	1	2	3	2





Maximization of absolute value

Consider the following mathematical model of four variables x_1, x_2, x_3 and x_4.

$|x_1| = \max\{x_1; -x_1\}$
$|x_2| = \max\{x_2; -x_2\}$

$$\text{Maximize } |x_1| + |x_2|$$

subject to the restrictions

4.5b

1. at least one of these two constraints must hold:
   - $x_1 + x_2 + x_3 + x_4 \leq 1000$
   - $3 \cdot x_1 + 5 \cdot x_2 + x_3 + 2 \cdot x_4 \leq 500$
2. x_1 is at least $\max\{2 \cdot x_2, x_3\}$.
3. $\frac{x_1}{x_4}$ equals to 4. $x_1 = 4x_4$; $(x_4 \neq 0) \rightarrow (x_4 > 0) \vee (x_4 < 0)$ není nutné díky
4. $x_{i \in \{1,2,3,4\}} \in \mathbb{Z}$
5. $-5000 \leq x_{i \in \{1,2,3\}}$, $x_4 \geq 1$, $x_{i \in \{1,2,3,4\}} \leq 1000$.

Formulate by the **Integer Linear Programming**.

konstanta M
$M = 4 \cdot 1000$

Vars:
$y \in \{0,1\}$
~~$z \in \{0,1\}$~~

4.) $x_{1,\dots,4} \in \mathbb{Z}$
$|x_1|; |x_2| \in \mathbb{Z}$

Constraints:

1.) $x_1 + x_2 + x_3 + x_4 \leq 1000 + My$ 1.5b
$3x_1 + 5x_2 + x_3 + 2x_4 \leq 500 + M(1-y)$

2.) $x_1 \geq 2 \cdot x_2$ 1b
$x_1 \geq x_3$

3.) $x_1 = 4 \cdot x_4$ 1b

~~$x_4 \Rightarrow M + x_4 > 0$~~
~~$x_4 < 0 + (1-y)M$~~

Obj cstr: $|x_1| \geq x_1$; $|x_1| \geq -x_1$; $|x_2| \geq x_2$; $|x_2| \geq -x_2$ X

5.) $-5000 \leq x_1$; $-5000 \leq x_2$; $-5000 \leq x_3$; $x_4 \geq 1$
$x_1 \leq 1000$; $x_2 \leq 1000$; $x_3 \leq 1000$; $x_4 \leq 1000$ 1b

Obj: maximize $|x_1| + |x_2|$