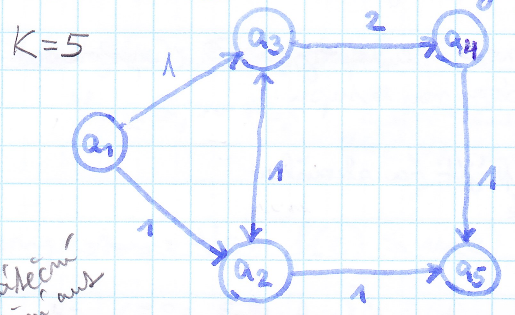
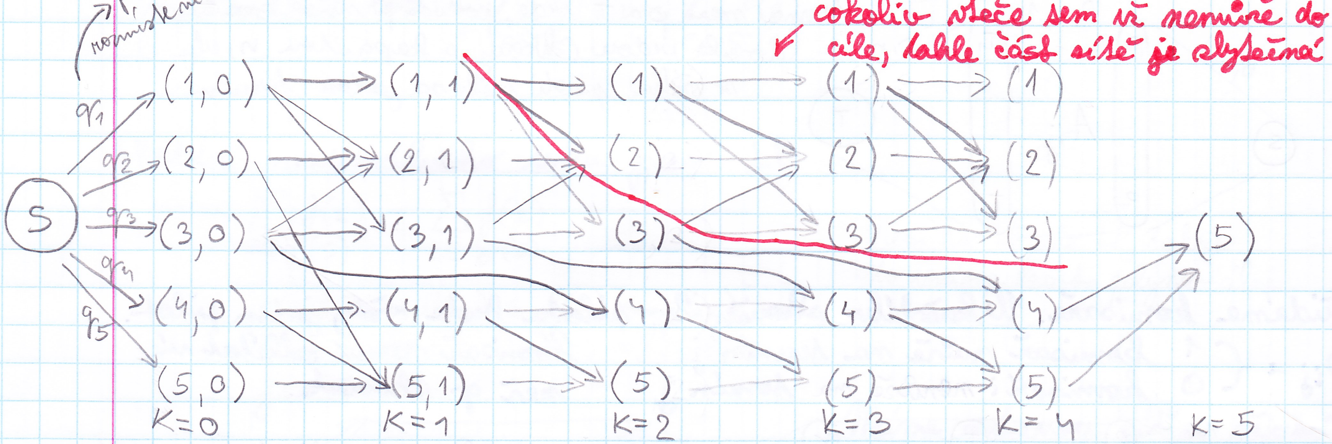
Přednáška 2.4.2019

Dynamický tok: mění se v čase, horní omezení u_i se změní na horní omezení hrany i v čase j
- např: data v síti, auta ve městě

Př: Máme města $a_1, a_2, \dots, a_n$ s počtem aut $q_1, q_2, \dots, q_n$. Všechna auta se během K hodin chtějí přesunout do města a_n.



ceny hran jsou časy potřebné k přejezdu, proměnnou kapacitu raději neřešíme



hrana z (i,j) do (i',j') vede, pokud je cena cesty z i do i' přesně $j'-j$.
dolní omezení tam není, horní omezení je kapacita hrany v daném čase
hrany $(i,j) \to (i,j')$ mají horní omezení $\sum q_k$, auta stojí a nikam nejedou

Tok jako ILP

$$\max \sum_{e\in\delta^+(s)} f(e) - \sum_{e\in\delta^-(s)} f(e)$$

s.t. $\sum_{e\in\delta^-(v)} f(e) = \sum_{e\in\delta^+(v)} f(e) \quad \forall v \in V(G)\setminus\{s,t\}$

$l(e) \leq f(e) \leq u(e) \quad \forall e \in E(G)$

Ford-Fulkersonův algoritmus

hrana dopředu: cesta ji prochází ve směru hrany
hrana vzad: cesta hranou prochází proti směru hrany

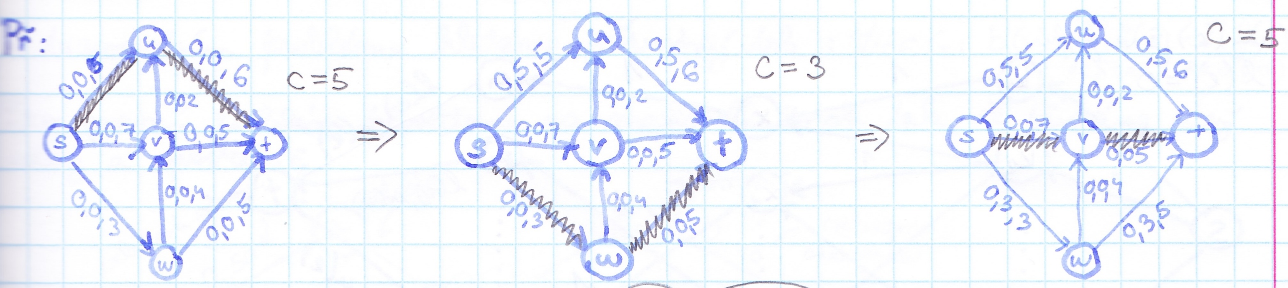
kapacita cesty C: $\sum_{e\in C} (u(e)-f(e),\ f(e)-l(e))$
pro hrany vpřed, pro hrany vzad

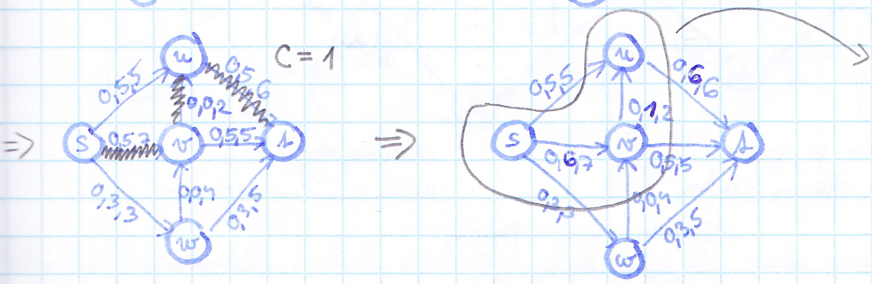
1) Máme nějaký přípustný tok v síti.
2) Najdeme zlepšující cestu, tj. cestu s kladnou kapacitou
3) Zvýšíme tok tak, abychom vyplnili celou kapacitu cesty.

Značkovací procedura

Vrchol v má značku m_v, na začátku m_s = TRUE, m_v = FALSE ($v \neq s$)
1) pokud máme $e \in G(E)$, $e = (v_i, v_j)$, m_i = TRUE, M_j = FALSE, a navíc pokud $f(e) < u(e)$. Pak m_j := TRUE
2) obdobně když $e = (v_j, v_i)$, $f(e) > l(e)$, pak m_j := TRUE
3) Jakmile narazíme na hranu do T, máme nalezenou zlepšující cestu.

Př:

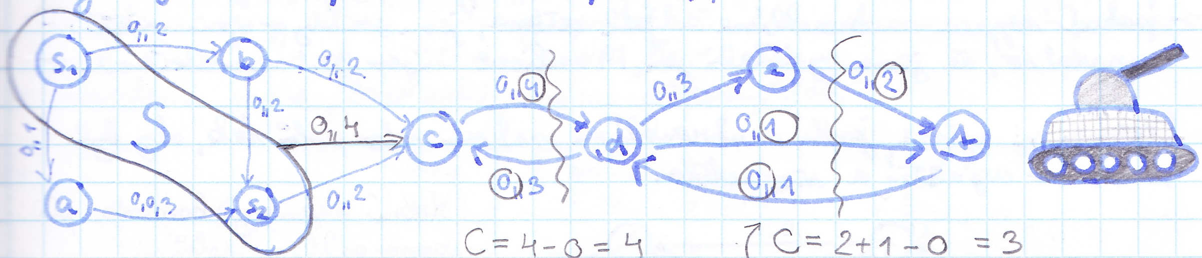




Tohle je minimální řez, už neexistuje zlepšující cesta

Řez grafu: Množina $\delta(A)$, $s \in A$, $t \notin A$, tj. t tam není. Minimální řez je takový řez, kde $C(A) = \sum_{e \in \delta^+(A)} u(e) - \sum_{e \in \delta^-(A)} l(e)$

Př: Máme tank v bodě t, snažíme se do něj dopravit palivo sítí trubek, z několika zdrojů S. Záškodník chce přerušit napájení, přičemž zneškodnění trubky stojí $k \cdot \alpha(e)$, $\alpha(e) \in \langle l(e), u(e) \rangle$, $k \in \mathbb{Z}^+$

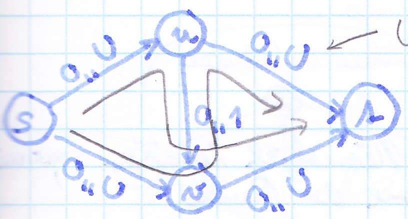


$C = 4 - 0 = 4$ $\quad$ $C = 2 + 1 - 0 = 3$

tenhle řez je minimální

Integral flow theorem: Pokud jsou všechna omezení celočíselná a existuje přípustný tok, pak existuje celočíselný maximální tok.

Časová náročnost Ford-Fulkersona



U je nějaké velké číslo

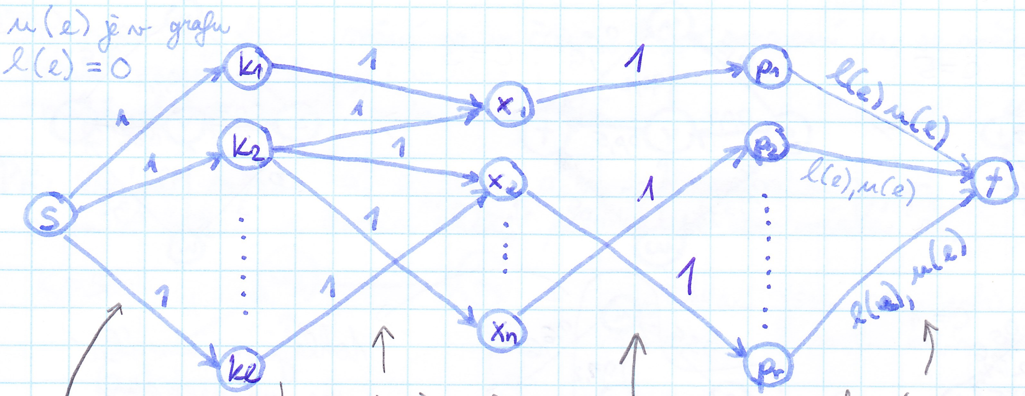
FF alg může najít $2 \cdot U$ velký tok, ale bude mu to trvat až $2U$ kroků − $O(|E|^2 \cdot U)$

Pro neceločíselná omezení nemusí nikdy skončit

Tohle řeší Edmonds Karp, který prohledává nejprve kratší cesty, a má časovou složitost $O(m^2 \cdot n)$

Př: Ve městě je n obyvatel, každý z nich chodí aspoň do jednoho z klubů $k_1 \dots k_l$, a každý patří přesně do jedné věkové skupiny $p_1 \dots p_r$. Každý klub musí nominovat jednoho zástupce do městské rady a tím, že počet členů rady s určitým věkem je shora i zdola omezen.

Řešení je na další straně



každý klub má jednoho člena rady

hrana ku je, pokud občan je členem klubu

každý občan patří do jedné kategorie

věková omezení v radě musí vyhovovat

Př: Zaokrouhlování

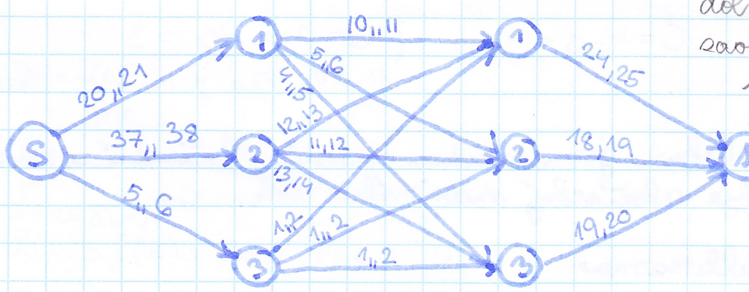
10.3	5.4	4.4	20.1
12.6	11.6	13.7	37.9
1.8	1.6	1.8	5.2
24.7	18.6	19.9	63.2

⇝

				jak to mělo vyjít
10	5	4	20	19
13	12	14	38	39
2	2	2	5	6
25	19	20	63	
25	19	20		64

- pokud zaokrouhlíme každou tabulku, součty nedávají smysl
- pokud nejprve zaokrouhlíme a pak sečteme, jsou součty o dost menší - vypadá to, že jsme vydělali jen 19 milionů místo 20.1 ve skutečnosti

Chceme si vybrat, jestli zaokrouhlíme nahoru nebo dolů tak, aby byl součet co největší a vše sedělo.



dolní hranice je hodnota při zaokrouhlení dolů

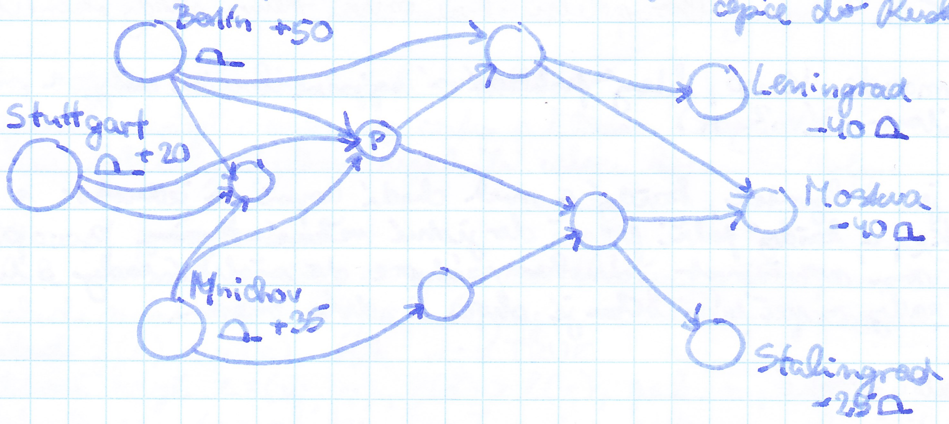
horní hranice je hodnota při zaokrouhlení nahoru

řádky

sloupce

hrana odpovídá jednomu políčku v tabulce

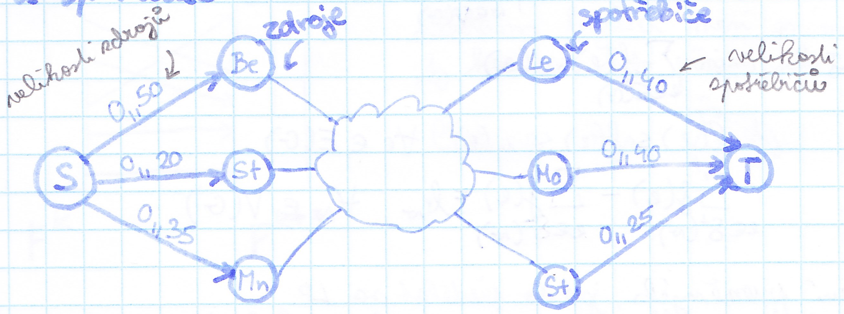
Př: Máme města spojená cestami. Potřebujeme z Německých továren dopravit čepice do Ruska.



b(v) je bilance vrcholu, kladná pro zdroje a záporná pro spotřebiče.
Říká se tomu Transportní problém.

V transportním problému hledáme přípustný tok : $\forall v \in G:\ b(v) = \sum_{e\in\delta^-(v)} l(e) - \sum_{e\in\delta^+(v)} l(e)$

Transportní problém převedeme na maximální tok tak, že přidáme nový zdroj a spotřebič



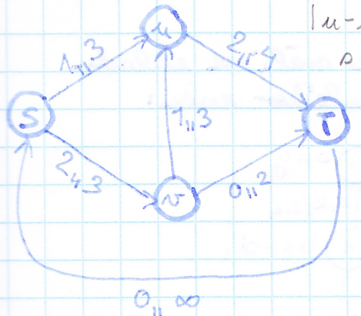
Pokud maximální tok saturuje hrany z S, pak má transportní problém řešení.

Jak hledat iniciální tok pro FF alg?

- pokud $\forall e: l(e) = 0$, je to přímo nulový tok
- v opačném případě
  - uděláme cirkulaci, tj. přidáme hranu $t \to s$, $u(e) = \infty$, $l(0)$, díky tomu nám Kirchhoffův zákon platí všude
  - nahradíme $f(e) = f(e)' + l(e)$

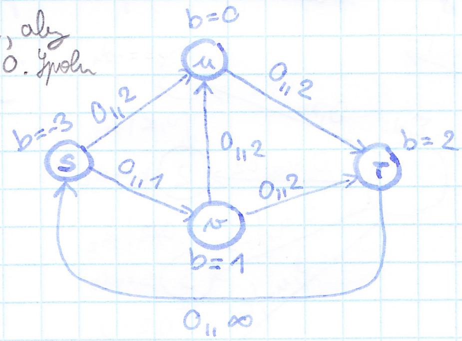
    $$\sum_{\delta^+} f(e)' - \sum_{\delta^-} f(e)' = \sum_{\delta^-} l(e) - \sum_{\delta^+} l(e) = b(v)$$
  - zavedeme nový zdroj a nový spotřebič, tím dostaneme síť, ve které je $l(e) = 0$, v ní najdeme max flow
  - převedeme max flow zpět, máme přípustný tok

Př.:

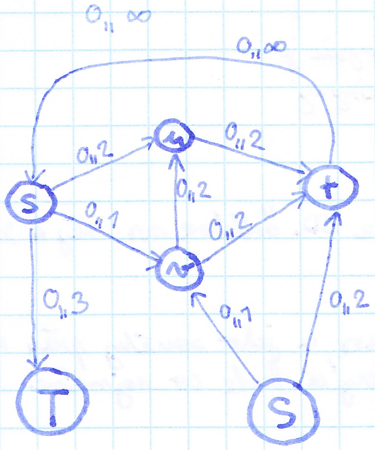


mezi u, l posuneme tak, aby $|u-l|$ se nezměnilo a $l=0$. Spolu s tím přepočítáme b

$\Rightarrow$



=



přidáme nové vrcholy S a T a hrany omezíme tak, aby odpovídaly bilancím

Cvičení
4.4.2019

Min cost flow

INPUT: (G, l, u, c, b)

$l, u: E(G) \to \mathbb{R}_0^+$

$c: E(G) \to \mathbb{R}$

$b: V(G) \to \mathbb{R}$, $\sum_{v \in V(G)} b(v) = 0$

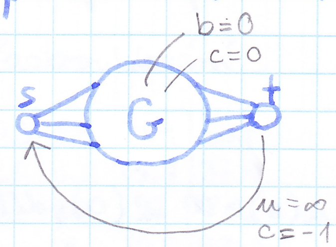
OUT: $f: E(G) \to \mathbb{R}_0^+$ | $\min \sum_{e \in E(G)} f(e) \cdot c(e)$

$$l(e) \le f(e) \le u(e) \quad \forall e \in E(G)$$

$$\sum_{e \in \delta^+(v)} f(e) - \sum_{e \in \delta^-(v)} f(e) = b_v \quad \forall v \in V(G)$$

Nikde nemám násobení proměnných, jde to předělat na LP.

Jak převedeme hledání MAX flow (G, l, u, s, t) na MIN cost flow $(G, l, u, c, $



Když v obrázku spustím min cost flow, bude se snažit protlačit co největší toku skrz novou hranu.
To je stejné, jako bychom pouštěli víc toku z S do t, takže jsme našli maximální tok.

Př: Zmrzlinář

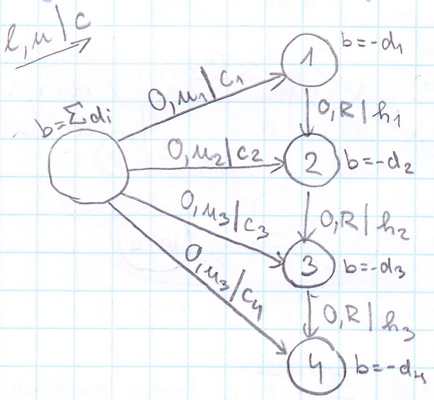
d_i .... poptávka po zmrzlině v období $i = 1 \dots 4$

c_i .... výrobní cena v období $i = 1 \dots 4$

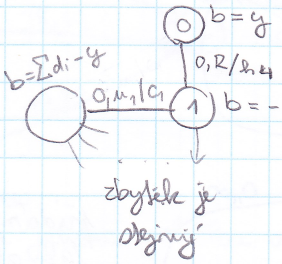
R .... kapacita skladu

h_j .... cena uskladnění z období j do $j+1$

u_j .... maximální kapacita produkce v období $j = 1 \dots 4$

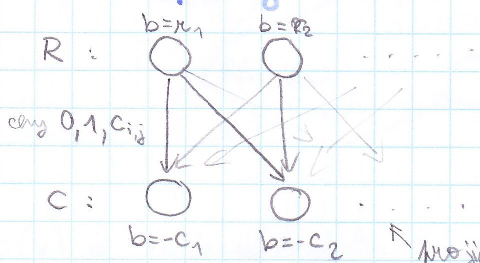


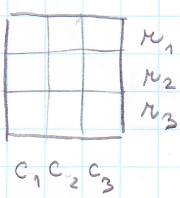
–kdybychom navíc měli ve skladu y zmrzlin z minulého roku:



Př: Máme krystal, který chceme zmapovat ve 3D. Svítíme na něj z různých směrů, a skládáme rekonstrukci zase ve 3D.

Zjednodušení: Máme černobílý obrázek $n \times n$ a jeho součty přes řádky R a sloupce C. Chceme najít původní obrázek, který se bude co nejméně lišit od odhadu I, který již máme.





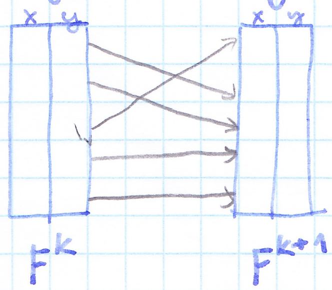
Pro obrázek velikosti $n \times n$ dostanu n^2 hran.

$c_{ij} = 1 - I_{ij}$

pro jiné projekce, třeba diagonální, bude mít graf jiný hran
ale stále zůstane n^2 hran

Úkol: Object tracking

Na vstupu je množina frames: $\{F^{(1)}, F^{(2)}, \dots, F^{(p)}\}$ pro časy $1 \dots p$,
a F^k je tabulka x, y souřadnic.

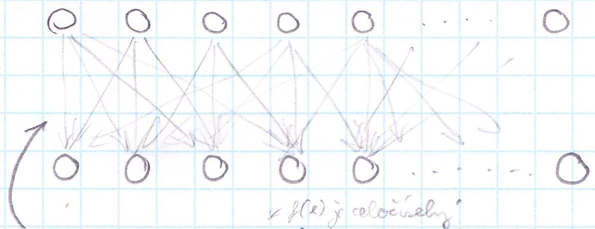


hledám bijekci mezi snímky, tj kam se kdo přesunul

Chceme min $\sum_{i=1}^{n} d_{ij}$

Budeme to řešit jako min cost pro dvojice sousedních snímků.

snímek k: vrchol pro každého hráče, $b = 1$



snímek k+1: vrchol pro každého hráče, $b = -1$

$f(e)$ je celočíselný

hrany mají omezení $\langle 0, 1 \rangle$, a cena je vzdálenost bodů.

Můžeme to dělat i Maďarským algoritmem, ale implementace tady toho bude možná v testu.