Toky v sítích

Zdeněk Hanzálek a Přemysl Šůcha
hanzalek@fel.cvut.cz

ČVUT FEL Katedra řídicí techniky

29. března 2011

1 Toky

- Problém maximálního toku
  - Ford-Fulkersonův Algoritmus
  - Problém minimálního řezu
  - Celočíselnost
- Rozhodovací problém přípustného toku v síti
  - Nalezení počátečního přípustného toku pro Ford-Fulkersonův alg.
- Nejlevnější tok v síti

2 Párování

- Maximální párování v bipartitním grafu
- Přiřazovací úloha - Nejlevnější perfektní párování v úplném bipart. gr.
  - Maďarský algoritmus

2 Multikomoditní toky

- Nejlevnější multikomoditní toky v síti

2 Vyjadřovací schopnosti problému maximálního toku

Síť a tok

Co je síť

Jako síť bývá označována pětice (G, l, u, s, t) kde G je orientovaný graf s hranami o horním omezení $u : E(G) \to \mathbb{R}_0^+$ a dolním omezení $l : E(G) \to \mathbb{R}_0^+$ a dvěma vrcholy s (zdroj) a t (spotřebič).

Tok

Tok v síti G je takové ohodnocení hran $f : E(G) \to \mathbb{R}_0^+$, kde pro každý vrchol $v \in V(G) \setminus \{s, t\}$ platí Kirchhofův zákon $\sum_{e \in \delta^-(v)} f(e) = \sum_{e \in \delta^+(v)} f(e)$.

$\delta^+(v)$ je množina hran opouštějících v
$\delta^-(v)$ je množina hran vstupujících do v

Přípustný tok

Pro přípustný tok platí $f(e) \in \langle l(e), u(e) \rangle$.
Přípustný tok nemusí existovat, pokud $l(e) > 0$.

Problém maximálního toku

Maximální tok

Je dána usp. pětice (G, l, u, s, t). Úkolem je najít takový **přípustný tok** f od zdroje ke spotřebiči, že $\sum_{e \in \delta^+(s)} f(e) - \sum_{e \in \delta^-(s)} f(e)$ je maximální (tj. chceme transportovat co nejvíce jednotek z s do t.).

$\delta^+(s)$ je množina hran opouštějících s
$\delta^-(s)$ je množina hran vstupujících do s (ty často neuvažujeme).

Příklad - dopravní úloha: Maximální množství produktu má být dopraveno z s do t. Problém je popsán sítí (grafem) kde hrany grafu odpovídají dopravním linkám (úsekům potrubí, silnic, železnic, atd.) s příslušným horním a dolním omezením. Tok v hraně je ustálený a beze ztrát.
Příklad omezení:

- $u_i = 10$ - linka i přepraví maximálně 10 jednotek
- $l_j = 3$ - linka j přepraví minimálně 3 jednotky
- $l_k = u_k = 20$ - linka k přepraví právě 20 jednotek

Př. - Rozvrhování na paralelních procesorech s pmtn, r_j, $\widetilde{d_j}$

Problém $P \left| \text{pmtn}, r_j, \widetilde{d_j} \right| C_{max}$ - máme n **úloh**, které je potřeba přiřadit na R **paralelních identických zdrojů** (procesorů). Každá úloha má svoji **dobu trvání** p_j, **termín dostupnosti** r_j a **termíny dokončení** $\widetilde{d_j}$. Při přiřazování úloh na procesory je povoleno úlohy **přerušovat** (včetně jejich migrace z jednoho procesoru na druhý). Příklad pro 3 zdroje:

úloha	T_1	T_2	T_3	T_4
p_j	1.5	1.25	2.1	3.6
r_j	3	1	3	5
$\widetilde{d_j}$	5	4	7	9

Cíl

Všechny úlohy přiřadit na procesory tak, aby každý zdroj vykonával v daný časový okamžik maximálně jednu úlohu a aby každá úloha byla v daný časový okamžik prováděna maximálně na jednom procesoru.

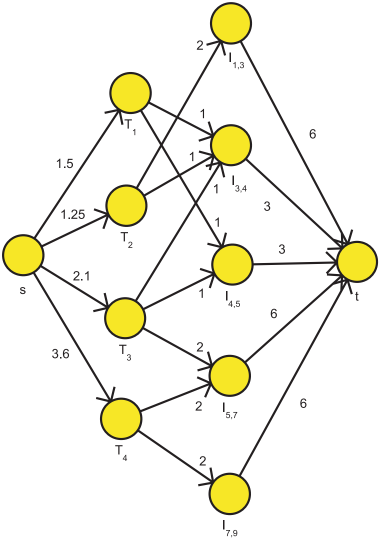
Formulujeme jako problém **maximálního toku**.

Př. - Rozvrhování na paralelních procesorech s preempcí, release date a deadline

pro 3 paralelní identické zdroje

T_j	1	2	3	4
p_j	1.5	1.25	2.1	3.6
r_j	3	1	3	5
$\widetilde{d}_j$	5	4	7	9

1) Vrcholy $I_{1,3}, I_{3,4}, I_{4,5}, I_{5,7}$ a $I_{7,9}$ odpovídají časovým intervalům uvnitř kterých může být vykonána jedna podmnožina úloh (intervaly jsou dány hodnotami r a $\widetilde{d}$). Například $T_{1,3}$ odpovídá intervalu $\langle 1, 3 \rangle$.
2) Horní omezení hrany $(T_j, I_{x,y})$ je dáno délkou intervalu (neboli $y - x$) jelikož úlohy nejsou vnitřně paralelní.
3) Horní omezení hrany $(I_{x,y}, t)$ je dáno délkou intervalu a počtem zdrojů.



Formulace maximálního toku pomocí lineárního programování

$f(e) \in \mathbb{R}_0^+$ je proměnná udávající tok hranou $e \in E(G)$.

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

$$\begin{array}{lll} \text{s.t.} & \sum_{e \in \delta^-(v)} f(e) = \sum_{e \in \delta^+(v)} f(e) & v \in V(G) \setminus \{s, t\} \\ & l(e) \le f(e) \le u(e) & e \in E(G) \end{array}$$

Všimněme si, že pro libovolnou množinu A obsahující zdoj s a neobsahující spotřebič t platí:
$\sum_{e \in \delta^+(s)} f(e) - \sum_{e \in \delta^-(s)} f(e) = \sum_{e \in \delta^+(A)} f(e) - \sum_{e \in \delta^-(A)} f(e)$.

Dú: jednoduše dokažte z platnosti Kirchhofova zákona.

Ford-Fulkersonův Algoritmus

Za průkopníky v oblasti toků v sítích jsou považováni L. R. Ford, Jr. a D. R. Fulkerson (na obrázku). Jejich jména nese neznámější **algoritmus na výpočet maximálního toku v síti**, který publikovali v roce 1956.

Ford-Fulkersonův Algoritmus

Princip: Algoritmus je založen na **postupném zvětšování (zlepšování) toku** při zachování jeho přípustnosti.

Hrany vpřed a vzad

Hranu nazveme *hranou vpřed*, je-li orientována ve směru průchodu cestou od zdroje ke spotřebiči. *Hrana vzad* je orientována proti směru průchodu.

Zlepšující cesta

Zlepšující cesta vzhledem k toku f je taková **neorientovaná** cesta ze zdroje s ke spotřebiči t, jejíž každá hrana splňuje:

- je-li e hranou vpřed, pak $f(e) < u(e)$ ... tok můžeme zvětšit
- je-li e hranou vzad, pak $f(e) > l(e)$ ... tok můžeme zmenšit

Kapacita zlepšující cesty

Kapacita zlepšující cesty je maximální hodnota γ, o kterou lze změnit tok na zlepšující cestě.

Ford-Fulkersonův Algoritmus

Vstup: Síť (G, l, u, s, t).
Výstup: Maximální přípustný tok f z s do t.

1. Najdi přípustný tok $f(e)$ pro všechny $e \in E(G)$
2. Najdi zlepšující cestu P. Pokud neexistuje, ukonči hledání.
3. Spočítej γ, kapacitu zlepšující cesty P. Zlepši tok z s do t a jdi na 2.

Na hranách vpřed zvýšíme tok o γ a na hranách vzad snížíme tok o γ - přípustnost toku i Kirchhofův zákon tím zůstaly zachovány, ale celková velikost toku stoupla o γ.
Tuto cestu již nelze použít, jelikož v jedné hraně byl tok změně "nadoraz".

Tok z s do t je **maximální** právě tehdy, když neexistuje zlepšující cesta.

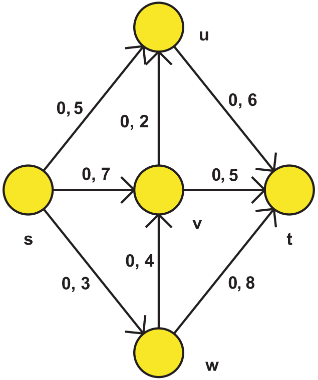
Hledání zlepšující cesty (značkovací procedura) pro Ford-Fulkersonův Algoritmus

Vstup: Síť (G, l, u, s, t), přípustný tok f.
Výstup: Zlepšující cesta P.

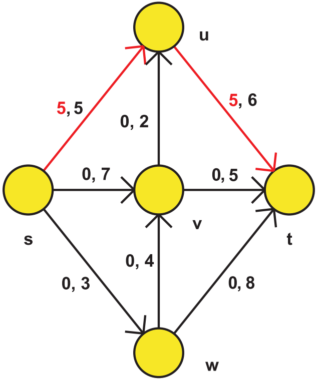
1. $m_v = FALSE \ \forall v \in V(G)$, $m_s = TRUE$ (označkuj vrchol s)
2. Existuje-li $e \in E(G)$ (přičemž v_i je počáteční a v_j je koncový vrchol hrany e) taková, že platí $m_i = TRUE$, $m_j = FALSE$ a $f(e) < u(e)$, pak označkujeme $m_j = TRUE$.
3. Existuje-li $e \in E(G)$ (přičemž v_i je počáteční a v_j je koncový vrchol hrany e) taková, že platí $m_i = FALSE$, $m_j = TRUE$ a $f(e) > l(e)$, pak označkujeme $m_i = TRUE$.
4. Pokud byl označkován t, hledání končí (zlepšující cesta P byla nalezena). Pokud nelze označit další vrchol, P neexistuje. V ostatních případech pokračuj body 2 a 3.

Ford-Fulkersonův Algoritmus
Příklad s nulovým dolním omezením toku

značení hran: $f(e), u(e)$

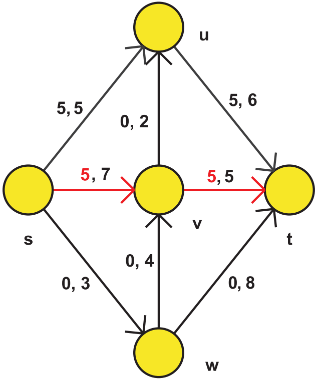


značení hran: $f(e), u(e)$



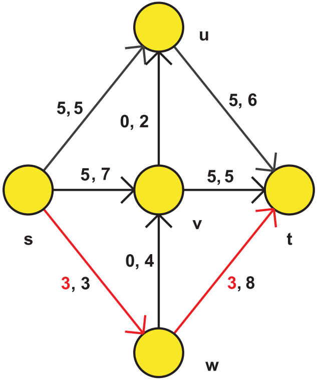
Ford-Fulkersonův Algoritmus
Příklad s nulovým dolním omezením toku

značení hran: $f(e), u(e)$



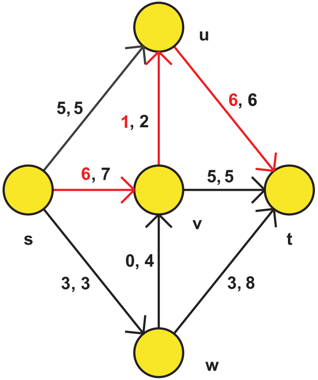
Ford-Fulkersonův Algoritmus
Příklad s nulovým dolním omezením toku

značení hran: $f(e), u(e)$



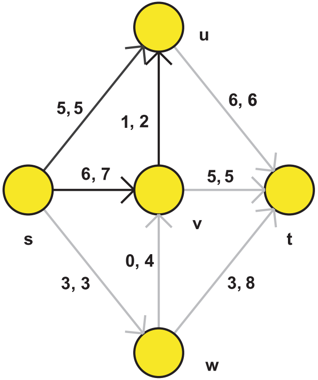
Příklad s nulovým dolním omezením toku

značení hran: $f(e), u(e)$



Ford-Fulkersonův Algoritmus
Příklad s nulovým dolním omezením toku

značení hran: $f(e), u(e)$

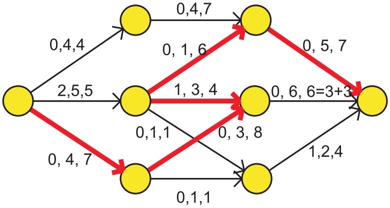


množina A=$\{s, u, v\}$ charakterizuje **řez s minimální kapacitou**

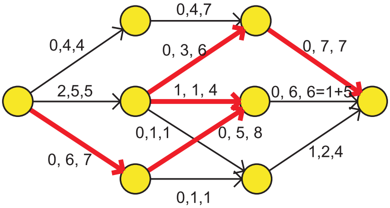
Ford-Fulkersonův Algoritmus
Příklad s nenulovým dolním omezením toku

Tento příklad zároveň ukazuje, že **značkování proti směru hran nelze v algoritmu vynechat**. Jinak by v tomto případě z výchozího toku (vlevo) nebylo možné dosáhnout maximálního toku (vpravo).

Značení hran: $l(e), f(e), u(e)$.
Kapacita zlepšující cesty je rovna 2.



Výsledný tok je maximální - nalezněte řez o minimální kapacitě.



Problém minimálního řezu

Řez

Řez v grafu G je množina hran $\delta^{(}A)$, kde $s \in A$ a $t \in V(G) \setminus A$ (neboli řez odděluje s a t). **Minimální řez** je řez s minimální kapacitou $C(A) = \sum_{e \in \delta^+(A)} u(e) - \sum_{e \in \delta^-(A)} l(e)$.

Ford-Fulkersonova věta [1956]

Hodnota maximálního toku z s do t v libovolné síti je rovna **kapacitě minimálního řezu**. Vyplývá z LP duality.

Po zastavení značkovací procedury, které se nepodařilo nalézt zlepšující cestu, je minimální řez charakterizován označkovanými vrcholy. Minimální řez je roven množině hran, které nedovolují další značkování. Pro všechny $e \in \delta^+(A)$ platí $f(e) = u(e)$ a pro všechny $e \in \delta^-(A)$ platí $f(e) = l(e)$.

Velikost maximálního toku je rovna kapacitě minimálního řezu:
$\sum_{e \in \delta^+(A)} f(e) - \sum_{e \in \delta^-(A)} f(e) = \sum_{e \in \delta^+(A)} u(e) - \sum_{e \in \delta^-(A)} l(e)$.

Celočíselnost

Integral Flow Theorem (Danzing and Fulkerson [1956])

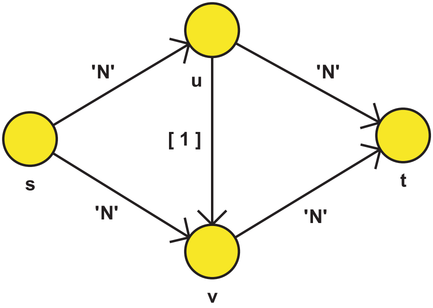
Pokud jsou kapacity sítě celočíselné, potom **existuje celočíselný maximální tok**.

Vyplývá z totální unimodularity incidenční matice orientovaného grafu G, která je přímo maticí $\mathbf{A}$ v LP formulaci $\mathbf{A} \cdot x \leq b$.

Lze též dokázat následovně:
Pokud jsou všechny kapacity celočíselné, pak je γ v kroku 3) Ford-Fulkersonova algoritmu vždy celočíselná. Algoritmus se zastaví po konečném počtu kroků, jelikož maximální tok má celočíselnou hodnotu

Ford-Fulkersonův Algoritmus - Časová složitost

Pokud hledáme zlepšující cestu **nevhodným způsobem**, může se tok zvyšovat jen po jednotkových krocích. Pro neceločíselná omezení toku a neceločíselný tok se **algoritmus nemusí vůbec zastavit**.



Edmonds a Karp [1972])

Pokud vždy vybíráme zlepšující cesty s **nejmenším počtem hran**, je časová náročnost algoritmu $O\left(m^2 \cdot n\right)$.

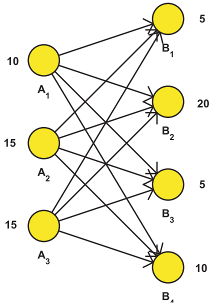
Rozhodovací problém přípustného toku v síti

Polynomiální transformací lze převést na problém maximálního toku.

Přípustný tok v síti

- **Instance**: Uspořádaná trojice (G, u, b) kde G je orientovaný graf s hranami o horním omezení $u : E(G) \rightarrow \mathbb{R}_0^+$ a dále s:
    - bilancí (**zdrojů/spotřebičů**) vrcholů $b : V(G) \rightarrow \mathbb{R}$ kde $\sum_{v \in V(G)} b(v) = 0$.
- **Cíl**: Rozhodnout, zda existuje přípustný tok f tak, aby platilo $\sum_{e \in \delta^+(v)} f(e) - \sum_{e \in \delta^-(v)} f(e) = b(v)$ pro všechny $v \in G(V)$.

Jeden produkt s danými **dodavateli** určitého množství (dodavatel reprezentován vrcholem s $b(v) > 0$) a **odběrateli** určitého množství (odběratel reprezentován vrcholem s $b(v) < 0$). Úkolem je rozhodnout, zda lze přepravit všechen produkt od dodavatelů k odběratelům v transportní síti s horním omezením linek u. Problém je popsán sítí (grafem) kde hrany grafu odpovídají úsekům potrubí, silnic, železnic, atd. s příslušnou kapacitou.



Cíl

Úkolem je rozhodnout, zda lze přepravit všechen produkt od dodavatelů A k odběratelům B v transportní síti s kapacitami linek u.

Rozhodovací problém přípustného toku v síti

Tento rozhodovací problém převedeme na **problém maximálního toku s nulovým dolním omezením**:

1. založíme zdroj s a přidáme hrany (s, v) s horním omezením $u_v = b(v)$ pro všechny vrcholy, pro které platí $b(v) > 0$
2. založíme spotřebič t a přidáme hrany (v, t) s horním omezením $u_v = -b(v)$ pro všechny vrcholy, pro které platí $b(v) < 0$
3. vyřešíme problém maximálního toku s nulovým dolním omezením (jako **počáteční přípustný tok** vezmeme nulový tok)
4. **pokud maximální tok saturuje** všechny hrany vycházející z s a/nebo vstupující do t, potom má problém přípustného toku v síti kladnou odpověď.

Nalezení počátečního přípustného toku pro Ford-Fulkersonův Algoritmus

Pro případ kdy $\forall e \in E(G)$; $l(e) = 0$ - triviální - lze vzít nulový tok, jelikož ten splňuje Kirchhofův zákon.
Pro případ kdy $\exists e \in E(G)$; $l(e) > 0$ převedeme hledání přípustného toku na **rozhodovací problém přípustného toku v síti** následujícím postupem:

1. Problém maximálního toku (s nenulovým dolním omezením) převedeme na **cirkulaci** přidáním hrany z t do s o nekonečném horním omezení, tím platí Kirchhofův zákon pro všechny vrcholy v síti (včetně s a t).
2. Hledaná přípustná cirkulace s dolním a horním omezením toku musí vyhovět následujícím omezením:

$$\sum_{e \in \delta^+(v)} f(e) - \sum_{e \in \delta^-(v)} f(e) = 0 \qquad v \in V(G)$$
$$l(e) \leq f(e) \leq u(e) \qquad e \in E(G)$$

Nalezení počátečního přípustného toku pro Ford-Fulkersonův Algoritmus

3. Substitucí $f(e) = f(e)' + l(e)$ obdržíme transformovaný problém:

   $$\sum_{e\in\delta^+(v)} f(e)' - \sum_{e\in\delta^-(v)} f(e)' = b(v) \qquad v \in V(G)$$
   $$0 \le f(e)' \le u(e) - l(e) \qquad e \in E(G)$$
   $$\text{kde } b(v) = \sum_{e\in\delta^-(v)} l(e) - \sum_{e\in\delta^+(v)} l(e) \qquad v \in V(G)$$

4. Toto je **rozhodovací problém přípustného toku v síti** jelikož $\sum_{v\in V(G)} b(v) = 0$ (povšimněte si, že $l(e)$ se nachází dvakrát v tomto součtu, jednou s kladným a jednou se záporným znaménkem).
5. Vyřešením tohoto rozhodovacího problému (t.j. přidáním s', t' a řešením maximálního toku s nulovým dolním omezením) zjistíme počáteční přípustnou cirkulaci/tok nebo rozhodneme, že neexistuje.

Závěr: problém nalezení **počátečního přípustného toku s nenulovým dolním omezením** jsme převedli na **rozhodovací problém přípustného toku v síti** a ten umíme převést na problém **maximálního toku s nulovým dolním omezením**.

Problém nejlevnějšího toku v síti

Rozšíření úlohy maximálního toku o ceny hran a exaktní rozdíl vstupního a výstupního toku ve zdrojích/spotřebičích.

Nejlevnější tok v síti

- **Instance**: Uspořádaná pětice (G, l, u, c, b) kde G je orientovaný graf s hranami o horním omezení $u : E(G) \rightarrow \mathbb{R}_0^+$ a dolním omezení $l : E(G) \rightarrow \mathbb{R}_0^+$ a dále s:
  - **cenami** hran $c : E(G) \rightarrow \mathbb{R}$
  - ohodnocením (**zdrojů/spotřebičů**) vrcholů $b : V(G) \rightarrow \mathbb{R}$ kde $\sum_{v \in V(G)} b(v) = 0$.
- **Cíl**: Nalézt přípustný tok f jehož cena $\sum_{e \in E(G)} f(e) \cdot c(e)$ je minimální (tj. chceme dopravit tok mezi uzly co nejlevněji) a zároveň platí $\sum_{e \in \delta^+(v)} f(e) - \sum_{e \in \delta^-(v)} f(e) = b(v)$ pro všechny $v \in G(V)$. Nebo rozhodnout, že přípustný tok neexistuje.

Nejlevnější tok v síti - formulace LP

Proměnná $f(e) \in \mathbb{R}_0^+$ reprezentuje tok hranou $e \in E(G)$.

$$\min \sum_{e \in E(G)} c(e) \cdot f(e)$$

$$\begin{aligned} \text{s.t.} \ & \sum_{e \in \delta^+(v)} f(e) - \sum_{e \in \delta^-(v)} f(e) = b(v) && v \in V(G) \\ & l(e) \leq f(e) \leq u(e) && e \in E(G) \end{aligned}$$

Maximální tok lze převést na nejlevnější tok:

- založ návratovou hranu z t do s s horním omezením ∞ a cenou -1
- ostatní hrany mají ceny rovny 0
- $b(v) = 0$ pro všechny vrcholy včetně s a t
- cirkulace s nejmenší (zápornou) cenou maximalizuje tok v návratové hraně

Párování - základní pojmy a problémy

Párování v grafu G je taková množina hran $P \subseteq E(G)$, že žádné dvě hrany z množiny P nemají společný vrchol.
Pokud všechny vrcholy G jsou incidentní s některou hranou P, potom P nazýváme **perfektním párováním**.
Problémy:

a) V daném grafu najít **maximální párování** (anglicky (Maximum) Cardinality Matching Problem), tj. párování které má největší počet hran.

b) **Maximální párování v bipartitním grafu** (spec. případ úlohy a).

c) V ohodnoceném grafu najít **nejlevnější maximální párování**, tj. nejlevnější párování ze všech, která jsou maximální.

d) V úplném ohodnoceném bipartitním grafu, jehož strany mají stejné počty vrcholů, najít **nejlevnější perfektní párování**. Tento problém se často nazývá **přiřazovací úloha** a je speciálním případem úlohy c) a speciálním případem **úlohy nejlevnějšího toku**.

Tyto problémy jsou polynomiální. My ukážeme algoritmy pro bipartitní grafy, které patří k jednodušším.

Maximální párování v bipartitním grafu

Lze řešit například pomocí úlohy **Maximálního toku**:

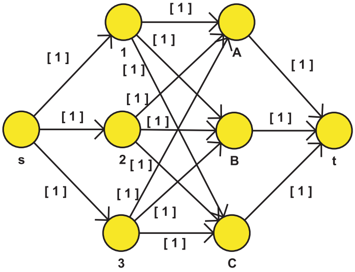
- založíme zdroj s a přidáme hrany (s, i) pro všechny $i \in X$
- založíme spotřebič t a přidáme hrany (j, t) pro všechny $j \in Y$
- orientaci hran zavedeme z s do X, z X do Y a z Y do t
- horní omezení všech hran jsou 1 a dolní omezení jsou 0
- vyřešíme problém maximálního toku z s do t a tím najdeme maximální párování

Příklad - Přiřazovací úloha

Máme *n* pracovníků a *n* úloh. Pro každou **dvojici pracovník-úloha** známe náklady na vykonání úlohy tímto pracovníkem.

Cíl

Každému pracovníkovi přiřadit jednu úlohu tak, aby celkové náklady byly minimální.



značení hran: $u(e)$
náklady na vykonání úloh
1,2,3 pracovníky A,B,C

	A	B	C
1	6	2	4
2	3	1	3
3	5	3	4

Vyřešíme buď jako problém **nejlevnějšího toku v síti** (viz. obrázek) nebo formulujeme jako problém **přiřazovací úlohy**.

Přiřazovací úloha - Nejlevnější perfektní párování v úplném bipartitním grafu, jehož strany mají stejné počty vrcholů

Popis:

- G - úplný neorientovaný bipartitní graf se stranami X, Y takový, že $|X| = |Y| = n$.
- Ceny hran uspořádáme do matice, jejíž prvek $c_{ij} \in \mathbb{R}_0^+$ je cenou hrany $(i, j) \in X \times Y$.

Základní myšlenka Maďarského algoritmu:

- Libovolné **ohodnocení vrcholů** reálnými čísly $p(v)$ pro $v \in V(G)$ definuje transformované ceny předpisem: $c_{ij}^p = c_{ij} - p_i^x - p_j^y$.
- Touto transformací se pro každé perfektní párování **změní cena o tutéž hodnotu** (každý vrchol se účastní právě jednou) a díky tomu je to nejlevnější stále dáno totožným výběrem hran.

Přiřazovací úloha - Maďarský algoritmus

Ohodnocení vrcholů p nazveme **přípustným ohodnocením**, jsou-li všechny transformované ceny nezáporné, tj. $c_{ij}^p \geq 0$.
Je-li p přípustným ohodnocením, pak **grafem rovnosti** G^p nazveme faktor grafu G, který obsahuje právě ty hrany, jejichž cena je nulová.

Věta

Jestliže graf rovnosti G^p obsahuje perfektní párování P, pak P je optimálním řešením přiřazovací úlohy

Párování P má v grafu rovnosti G^p nulovou cenu. Žádné jiné párování nemůže být levnější, protože jde o přípustné ohodnocení, kde $c_{ij}^p \geq 0$.

Maďarský algoritmus

Vstup: Úplný neorientovaný bipartitní graf G a váhy $c : E(G) \rightarrow \mathbb{R}_0^+$.
Výstup: Perfektní párování $P \subseteq E(G)$ jehož cena $\sum_{(i,j) \in P} c_{ij}$ je minimální.

1. Pro všechna $i \in X$ spočítej $p_i^x := \min_{j \in Y} \{c_{ij}\}$
   a pro všechna $j \in Y$ spočítej $p_j^y := \min_{i \in X} \{c_{ij} - p_i^x\}$
2. Sestroj gr. rovnosti G^p; $E(G^p) = \left\{(i,j) \in E(G); c_{i,j} - p_i^x - p_j^y = 0\right\}$
3. Nalezni maximální párování P v grafu G^p.
   Je-li toto párování perfektní, výpočet končí.
4. Není-li P perfektní, nalezni množinu $A \subseteq X$ a k ní v G^p incidentní množinu $B \subseteq Y$ takovou, že $|A| > |B|$. Spočítej
   $$d = min_{i \in A, j \in Y \setminus B} \left\{c_{i,j} - p_i^x - p_j^y\right\}$$
   a změň přípustné ohodnocení vrcholů takto:
   $$p_i^x := p_i^x + d \text{ pro všechna } i \in A$$
   $$p_j^y := p_j^y - d \text{ pro všechna } j \in B$$
   Pokračuj krokem 2.

Časová náročnost algoritmu je $O(n^4)$.

Maďarský algoritmus - příklad

Matice cen (pozor nejde o adjugovanou ani incidenční matici):

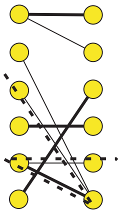
$$\begin{array}{rrrrrr}
5 & 3 & 7 & 4 & 5 & 4 \\
10 & 11 & 10 & 7 & 8 & 3 \\
18 & 7 & 6 & 6 & 6 & 2 \\
6 & 12 & 2 & 1 & 9 & 8 \\
8 & 4 & 4 & 4 & 1 & 1 \\
4 & 8 & 1 & 3 & 7 & 4
\end{array}$$

1. Nejdříve odečteme řádková minima od jednotlivých řádků - získáme ohodnocení vrcholů strany X.
   Ve vzniklé matici odečteme od každého sloupce sloupcové minimum - získáme ohodnocení vrcholů strany Y.

Maďarský algoritmus - příklad

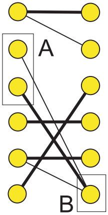
2. Vytvoříme matici transformovaných cen, ke každému řádku si poznamenáme hodnotu p_i^x a ke každému sloupci hodnotu p_j^y. Sestrojíme graf rovnosti G^p.
3. V G^p nalezneme maximální párování (hrany zvýrazněny tučně).
   - Nalezneme libovolné párování.
   - Nalezneme alternativní cestu (její hrany střídavě leží/neleží v párování) s neporkytými krajními vrcholy. Podél této cesty změníme (tj. zvětšíme) párování. Opakujeme dokud taková cesta existuje.

							p_i^x
	0	0	4	1	2	1	3
	5	8	7	4	5	0	3
	14	5	4	4	4	0	2
	3	11	1	0	8	7	1
	5	3	3	3	0	0	1
	1	7	0	2	6	3	1
p_j^y	2	0	0	0	0	0	



4. Jelikož výsledné párování není perfektní, nalezneme (modrou) množinu A a (zelenou) množinu B (začni značkovací proceduru z volného vrcholu v X). Z modrých prvků matice cen nalezneme minimum $d = 4$.

							p_i^x
	0	0	4	1	2	1	3
	5	8	7	4	5	0	3
	14	5	4	4	4	0	2
	3	11	1	0	8	7	1
	5	3	3	3	0	0	1
	1	7	0	2	6	3	1
p_j^y	2	0	0	0	0	0	

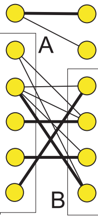


Hodnoty p_i^x snížíme o d, hodnoty p_j^y zvýšíme o d, přepočítáme matici cen.

Maďarský algoritmus - příklad

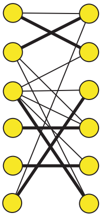
2. V transformované matici přibylo několik nul a v G^p několik hran. Hrana (5,6) naopak ubyla.
3. Párování nelze zvětšit.
4. Nalezneme množiny A (modře) a B (zeleně). Minimum $d = 1$.

							p_i^x
	0	0	4	1	2	5	3
	1	4	3	0	1	0	7
	10	1	0	0	0	0	6
	3	11	1	0	8	11	1
	5	3	3	3	0	4	1
	1	7	0	2	6	7	1
p_j^y	2	0	0	0	0	-4	



2. Nyní již v grafu existuje perfektní párování. Cena je rovna součtu ohodnocení vrcholů, 18.

							p_i^x
	0	0	5	2	3	6	3
	0	3	3	0	1	0	8
	9	0	0	0	0	0	7
	2	10	1	0	8	11	2
	4	2	3	3	0	4	2
	0	6	0	2	6	7	2
p_j^y	2	0	-1	-1	-1	-5	

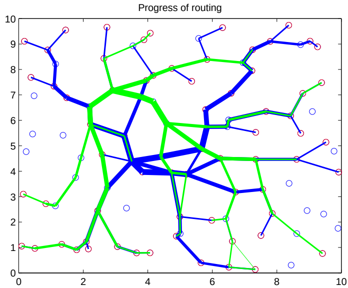


Multikomoditní toky

Doposud jsme předpokládali pouze jednu komoditu.
Zavedeme **množinu komodit** M transportované v téže síti.
Každá komodita má několik zdrojů a několik spotřebičů.
Proměnná $f^m(e) \in \mathbb{R}_0^+$ reprezentuje tok komodity $m \in M$ hranou $e \in E(G)$.

Příklad: senzorová síť se dvěma komoditami a jedním spotřebičem pro každou z komodit:

- zdrojové vrcholy měří **teplotu(zelená)** a/nebo **vlhkost(modrá)** a zasílají je do jednoho koncentrátoru dat (spotřebiče) pro teplotu a jednoho pro vlhkost
- velikost toku (množství dat za čas)



Komunikační linky:

- kapacita (množství dat za čas)
- cena (energie na přenesení dat)

Nejlevnější multikomoditní toky v síti

- **Instance**: Uspořádaná pětice $(G, l, u, c, b^1 \dots b^m \dots b^{|M|})$ kde G je orientovaný graf s hranami o horním omezení $u : E(G) \to \mathbb{R}_0^+$, dolním omezení $l : E(G) \to \mathbb{R}_0^+$ a ceně $c : E(G) \to \mathbb{R}$ a dále s:
  - ohodnocením (**zdrojů/spotřebičů**) vrcholů $b^m : V(G) \to \mathbb{R}$ kde $\sum_{v \in V(G)} b^m(v) = 0$ **pro všechny komodity** $m \in M$.
- **Cíl**: Nalézt přípustný tok f jehož cena $\sum_{e \in E(G)} \sum_{m \in M} f^m(e) \cdot c(e)$ je minimální (tj. chceme dopravit tok mezi uzly co nejlevněji) nebo rozhodnout, že přípustný tok neexistuje.
  Pro přípustný tok platí $\sum_{e \in \delta^+(v)} f^m(e) - \sum_{e \in \delta^-(v)} f^m(e) = b^m(v)$ pro všechny $v \in G(V)$ a všechny komodity $m \in M$.

Nejlevnější multikomoditní toky v síti - formulace LP

Proměnná $f^m(e) \in \mathbb{R}_0^+$ reprezentuje tok komodity $m \in M$ hranou $e \in E(G)$.

$$
\begin{array}{lll}
\min & \sum_{e \in E(G)} \sum_{m \in M} f^m(e) \cdot c(e) & \\
 & & \\
\text{s.t.} & \sum_{e \in \delta^+(v)} f^m(e) - \sum_{e \in \delta^-(v)} f^m(e) = b^m(v) & v \in V(G), m \in M \\
 & l(e) \leq \sum_{m \in M} f^m(e) \leq u(e) & e \in E(G)
\end{array}
$$

- 1. Kirchhofův zákon platí v každém vrcholu pro **každou komoditu**
- Multikomoditní toky lze řešit pomocí LP - **polynomiální problém**
- Celočíselnost však není zaručena, jelikož matice A v LP **není totálně unimodulární**
- (Praktická zkušenost) ILP formulace, zaručující celočíselnost, je řešitelná pro velké instance v přijatelném čase

Příklad Flow1: Dynamický tok

Dynamický tok je tok, který mění svou velikost v závislosti na čase. Ukážeme, jak lze úlohy o dynamickém toku převést na vhodnou úlohu o běžném (statickém) toku zavedením diskrétního času.

Například: **Ve městech** $a_1, a_2, \ldots, a_n$ je k dispozici $q_1, q_2, \ldots, q_n$ **automobilů**, které mají být dopraveny **do města** a_n během K hodin. Jsou-li města a_i, a_j spojena přímou silnicí, označme d_{ij} **dobu jízdy** mezi těmito městy a dále označme u_{ij} **kapacitu této silnice**, neboli maximální počet, která tudy mohou projet za hodinu. Konečně označme p_i **kapacitu parkovišť** ve městě a_i. Úkolem je zorganizovat pohyby aut tak, aby jich co největší počet dorazil do města a_n během K hodin.

Příklad Flow2: Výběr reprezentantů - existence spodního omezení

Přiřazovací úloha s dalšími omezeními.

- Každý z n lidí je členem jednoho nebo více profesních klubů $k_1, \ldots, k_l$ a spadá do jedné věkové skupiny $p_1, \ldots, p_r$.
- Každý klub chce vybrat ze svých členů jednoho reprezentanta. Nikdo však nesmí reprezentovat více než jeden klub.
- Je potřeba dodržet zastoupení věkových skupin - pro každou je stanoven minimální a maximální počet reprezentantů.

Úkolem je najít co největší sbor reprezentantů nebo dokázat, že neexistuje. Zformulujte jako úlohu maximálního (přípustného) toku.

Příklad Flow3: Zaokrouhlování v tabulce - existence spodního omezení

Mějme matici $p \times q$ reálných čísel s vektorem řádkových součtů a vektorem sloupcových součtů. Pro každé z těchto čísel a máme rozhodnout, zda dané číslo zaokrouhlíme nahoru a nebo dolů a. Rozhodnutí však musí splňovat dvě podmínky:

- zaokrouhlený řádkový součet je roven součtu zaokrouhlených čísel v řádku
- zaokrouhlený sloupcový součet je roven součtu zaokrouhlených čísel ve sloupci

Úkolem je maximalizovat součet prvků v tabulce (díky podmínkám je tento součet roven sumě řádkových součtů a zároveň roven sumě sloupcových součtů).

Literatura

Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin.
Network Flows: Theory, Algorithms, and Applications.
Prentice Hall, 1993.

Jiří Demel.
Grafy a jejich aplikace.
Academia, 2002.

B. H. Korte and Jens Vygen.
Combinatorial Optimization: Theory and Algorithms.
Springer, fourth edition, 2008.