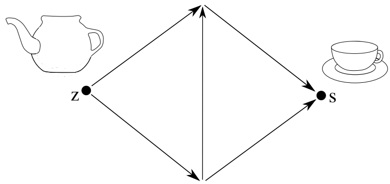
1. Toky v sítích

(zapsala Markéta Popelová)

První motivační úloha: Rozvod čajovodu do všech učeben.

Představme si, že by v budově fakulty na Malé Straně existoval čajovod, který by rozváděl čaj do každé učebny. Znázorněme si to orientovaným grafem, kde by jeden významný vrchol představoval čajovar a druhý učebnu, ve které sedíme. Hrany mezi vrcholy by představovaly větvící se trubky, které mají čaj rozvádět. Jak rozvést co nejefektivněji dostatek čaje do dané učebny?



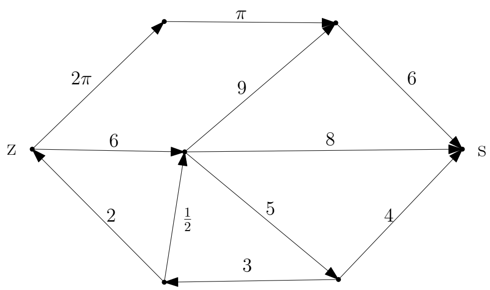
Čajovod

Druhá motivační úloha: Přenos dat.

Jiným příkladem může být počítačová síť na přenos dat, která se sestává z přenosových linek spojených pomocí routerů. Data se sice obvykle přenášejí po paketech, ale to můžeme při dnešních rychlostech přenosu zanedbat a považovat data za spojitá. Jak přenášet data mezi dvěma počítači v síti co nejrychleji?

Definice: *Síť* je uspořádaná pětice (V, E, z, s, c), kde platí:

- (V, E) je orientovaný graf.
- $c : E \to \mathbb{R}_0^+$ je *kapacita* hran.
- $z, s \in V$ jsou dva vrcholy grafu, kterým říkáme *zdroj* a *stok* (spotřebič).
- Graf je symetrický, tedy $\forall u, v \in V : uv \in E \Leftrightarrow vu \in E$ (tuto podmínku si můžeme zvolit bez újmy na obecnosti, neboť vždy můžeme do grafu přidat hranu, která v něm ještě nebyla, a dát jí nulovou kapacitu).



Příklad sítě. Čísla představují kapacity jednotlivých hran.

Definice: *Tok* je funkce $f : E \to \mathbb{R}_0^+$ taková, že platí:

1. Tok po každé hraně je omezen její kapacitou: $\forall e \in E : f(e) \le c(e)$.
2. Kirchhoffův zákon:

$$\forall v \in V \setminus \{z, s\} : \sum_{u:uv \in E} f(uv) = \sum_{u:vu \in E} f(vu).$$

   Neboli pro každý vrchol kromě zdroje a stoku platí, že to, co do něj přitéká, je stejně velké jako to, co z něj odtéká.

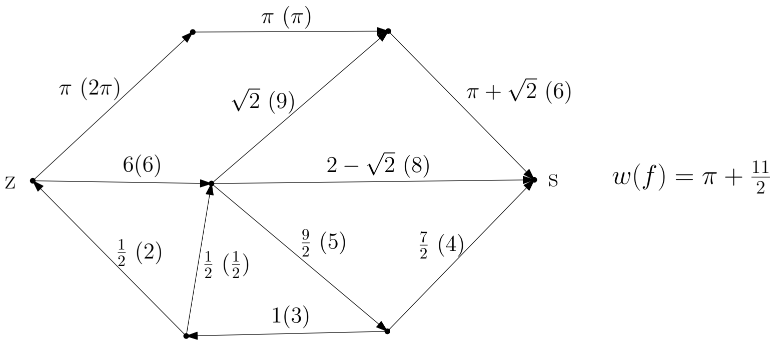
Poznámka: Pro zjednodušení zaveďme speciální značení:

- $f^+(v) = \sum_{u:uv \in E} f(uv)$ (to, co do vrcholu přitéká)
- $f^-(v) = \sum_{u:vu \in E} f(vu)$ (to, co z vrcholu odtéká)
- $f^\Delta(v) = f^+(v) - f^-(v)$ (rozdíl těchto hodnot)

   Pak můžeme Krichhoffův zákon zapsat jednoduše jako:

$$\forall v \in V \setminus \{z, s\} : f^\Delta(v) = 0.$$

Poznámka: V angličtině se obvykle zdroj značí s a stok t jako source a target.



Příklad toku. Čísla představují toky po hranách, v závorkách jsou kapacity.

Pozorování: Nějaký tok vždy existuje. V libovolné síti můžeme vždy zvolit funkci nulovou (po žádné hraně nic nepoteče). Tato funkce splňuje podmínky toku, a tedy takovýto nulový tok je zcela korektní.

Definice: *Velikost toku* f je rozdíl součtu velikostí toku na hranách vedoucích do s a součtu velikostí toku na hranách vedoucích z s. Neboli od toho, co do stoku přitéká odečteme to, co ze stoku odtéká.

$$|f| := f^\Delta(s).$$

Cíl: Budeme chtít najít v zadané síti tok, jehož velikost je maximální.

Otázka: Má vůbec smysl mluvit o maximálním toku? Bude vždy existovat? Nevybíráme zde totiž z konečně mnoha případů a na první pohled není jasné, že supremum množiny všech toků bude zároveň i maximum této množiny.

Odpověď: Ano, pro každou síť existuje maximální tok. Toto poměrně překvapivé tvrzení můžeme nahlédnout za pomoci matematické analýzy. Nástin důkazu je takový, že množina toků je kompaktní a velikost toku je spojitá (dokonce lineární) funkce z množiny toků do $\mathbb{R}$. Proto nabývá velikost toku na množině všech toků svého maxima.

Poznámka: Pro naše případy předpokládejme, že kapacity jsou racionální. Poměrně nám to zjednoduší práci a příliš nám to neublíží, neboť práce s reálnými čísly je stejně pro informatika poměrně zapeklitá.

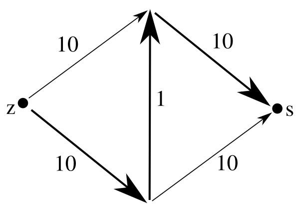
První řešení: Hledejme cestu P ze z do s takovou, že $\forall e \in P : f(e) < c(e)$ (po všech jejích hranách teče ostře méně, než jim dovolují jejich kapacity). Pak zjevně můžeme tok upravit tak, aby se jeho velikost zvětšila. Zvolme

$$\varepsilon := \min_{e \in P}(c(e) - f(e)).$$

Nový tok f' pak definujme jako $f'(e) := f(e) + \epsilon$. Kapacity nepřekročíme (ϵ je největší možná hodnota, abychom tok zvětšili, ale nepřekročili kapacitu ani jedné z hran cesty P) a Kirchhoffovy zákony zůstanou neporušeny, neboť zdroj a stok nezahrnují a každému jinému vrcholu na cestě P se přítok $f^+(v)$ i odtok $f^-(v)$ zvětší přesně o ϵ.

Otázka: Najdeme takto ovšem opravdu maximální tok?

Odpověď: Nemusíme. Např. na obrázku je vidět, že když najdeme nejdříve cestu přes hranu s kapacitou 1 (na obrázku tučně) a už hodnotu toku na této hraně nesnížíme, tak dosáhneme velikost toku nejvýše 19. Ale maximální tok této sítě má velikost 20.



Čísla představují kapacity jednotlivých hran.

Zde by ovšem situaci zachránilo, kdybychom poslali tok velikosti 1 proti směru prostřední hrany – to můžeme udělat třeba odečtením jedničky od toku po směru hrany.

Někdy je tedy potřeba poslat něco i v protisměru. Definujme si *rezervu hrany*. Ta nám říká, kolik můžeme daným směrem ještě poslat. Využijeme zde, že síť je symetrická.

Definice: *Rezerva hrany uv* je $r(uv) := c(uv) - f(uv) + f(vu)$.

Ukažme si nyní algoritmus, který rezervy využívá, a dokažme, že je konečný a že najde maximální tok každé racionální sítě.

Algoritmus (Fordův-Fulkersonův)

1. $f \leftarrow$ libovolný tok, např. všude nulový ($\forall e \in E : f(e) \leftarrow 0$).
2. Dokud $\exists P$ cesta ze z do s taková, že $\forall e \in P : r(e) > 0$, opakujeme:
3. $\quad \varepsilon \leftarrow \min\{r(e) \mid e \in P\}$.
4. $\quad$ Pro všechny hrany $uv \in P$:
5. $\quad\quad \delta \leftarrow \min\{f(vu), \varepsilon\}$
6. $\quad\quad f(vu) \leftarrow f(vu) - \delta$
7. $\quad\quad f(uv) \leftarrow f(uv) + \varepsilon - \delta$
8. Prohlásíme f za maximální tok.

Problém: Zastaví se Fordův-Fulkersonův algoritmus?

- Pro celočíselné kapacity se v každém kroku zvětší velikost toku alespoň o 1. Algoritmus se tedy zastaví po nejvíce tolika krocích, jako je nějaká horní závora pro velikost maximálního toku – např. součet kapacit všech hran vedoucích do stoku
$$\sum_{u:us \in E} c(us).$$
- Pro racionální kapacity využijeme jednoduchý trik – kapacity vynásobíme společným jmenovatelem a převedeme na původní případ. Uvědomme si, že algoritmus nikde kapacity hran ale ani toky na hranách nedělí, takže už zůstanou celočíselné. A tak jsme převedli racionální kapacity na celočíselné, pro které už víme, že se algoritmus zastaví.
- Na síti s iracionálními kapacitami se algoritmus chová mnohdy divoce, nemusí se zastavit ale dokonce ani konvergovat ke správnému výsledku. **K zamyšlení:** Zkuste vymyslet příklad takové sítě.

Otázka: Vydá algoritmus maximální tok?

Odpověď: Vydá. Abychom si to dokázali, zaveďme si řezy a použijme je jako certifikát maximality nalezeného toku.

Definice: *Řez* je uspořádaná dvojice množin vrcholů (A, B) taková, že A a B jsou disjunktní, pokrývají všechny vrcholy, A obsahuje zdroj a B obsahuje stok. Neboli $A \cap B = \emptyset$, $A \cup B = V$, $z \in A$, $s \in B$.

Definice: *Hrany řezu* $E(A, B) := E \cap A \times B$.

Poznámka: Řezy se dají definovat více způsoby, jedna z definic je, že řez je množina hran grafu takových, že po jejich odebrání se graf rozpadne na více komponent. Tuto definici splňuje i ta naše, ale ne naopak.

Definice: *Kapacita řezu* je

$$c(A, B) := \sum_{e \in E(A,B)} c(e).$$

Definice: *Tok přes řez* je

$$f(A,B) := \sum_{e\in E(A,B)} f(e) - \sum_{e\in E(B,A)} f(e).$$

Pozorování: Pro každý tok f a každý řez (A,B) platí, že $f(A,B) \le c(A,B)$.

Důkaz:

$$f(A,B) = \sum_{e\in(A,B)} f(e) - \sum_{e\in E(B,A)} f(e) \le \sum_{e\in E(A,B)} f(e) \le \sum_{e\in E(A,B)} c(e) = c(A,B).$$

♡

Lemmátko: Pro každý tok f a pro každý řez (A,B) platí $f(A,B) = |f|$.

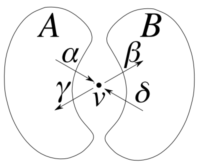
Důkaz: Indukcí a obrázkem.

Začneme s řezem $(V \setminus \{s\}, \{s\})$. Pro tento řez lemma platí z definice velikosti toku. Dále budu postupně přesouvat vrcholy z množiny B do množiny A. Libovolný řez může být takto vytvořen z toho triviálního.

Představme si, že máme již libovolný řez (A,B) a přesouváme vrchol v z A do B. Tedy $A' = A \setminus \{v\}$ a $B' = B \cup \{v\}$.

Uvědomme si, že všechny hrany jednoho typu (např. vedoucí z A do v) se chovají stejně, takže stačí uvažovat hrany pouze 4 typů (+ ostatní hrany (ty, které přesun neovlivní) označíme ε):

- α – hrany vedoucí z A do v
- β – hrany vedoucí z v do B
- γ – hrany vedoucí z v do A
- δ – hrany vedoucí z B do v



Přesun vrcholu v z A do B.

Před přesunem ($v \in A$) se $f(A,B)$ skládá z $\varepsilon + \beta - \delta$. Po přesunu ($v \in B$) se $f(A',B')$ skládá z $\varepsilon + \alpha - \gamma$. Rozdíl těchto hodnot je $\alpha + \delta - \beta - \gamma$.

Nicméně z Kirchhoffova zákonu o vrcholu v (což není ani zdroj ani stok) víme, že $\alpha + \delta - \beta - \gamma = f^{\Delta}(v) = 0$, neboť $\alpha + \delta$ je to, co do v přitéká, a $\beta + \gamma$ je to, co z v vytéká. Tedy tok přes řez před přesunem je stejně velký jako tok přes řez po přesunu. Pokud lemma platilo před přesunem, musí platit i po přesunu. ♡

Důsledek: Pro každý tok f a řez (A,B) platí, že $|f| = f(A,B) \le c(A,B)$.

Pozorování: Pokud najdeme dvojici tok f a řez (A, B) takovou, že platí $|f| = c(A, B)$, pak tok f je maximální a řez (A, B) minimální.

Věta: Pokud se Fordův-Fulkersonův algoritmus zastaví, tak vydá maximální tok.

Důkaz:

Nechť se Fordův-Fulkersonův algoritmus zastaví. Definujme $A = \{v \in V; \exists$ cesta ze z do v jdoucí po hranách s $r > 0\}$ a $B = V \setminus A$.

Uvědomme si, že (A, B) je řez, neboť $z \in A$ (ze z do z existuje cesta délky 0) a $s \in B$ (kdyby $s \notin B$, tak by musela existovat cesta ze z do s s kladnou rezervou, tudíž by algoritmus neskončil, nýbrž tuto cestu vzal a stávající tok vylepšil).

Dále víme, že všechny hrany řezu mají nulovou rezervu, neboli $\forall uv \in E(A, B)$: $r(uv) = 0$ (kdyby měla hrana uv rezervu nenulovou, tedy kladnou, tak by vrchol v patřil do A). Proto po všech hranách řezu vedoucích z A do B teče tolik, kolik jsou kapacity těchto hran, a po hranách vedoucích z B do A neteče nic, tedy $f(uv) = c(uv)$ a $f(vu) = 0$. Máme řez (A, B) takový, že $f(A, B) = c(A, B)$. To znamená, že jsme našli maximální tok a minimální řez. ♡

Zformulujme si, co jsme zjistili a dokázali o algoritmu pánů Forda a Fulkersona.

Věta: Pro síť s racionálními kapacitami se Fordův-Fulkersonův algoritmus zastaví a vydá maximální tok a minimální řez.

Věta: (Fordova-Fulkersonova)

$$\min_{(A,B)\ \check{r}ez} c(A, B) = \max_{f\ tok} |f|.$$

Důkaz:

Již víme, že $\min_{(A,B)} c(A, B) \geq \max_f |f|$. Stačí tedy dokázat, že vždy existují tok f a řez (A, B) takové, že $c(A, B) = |f|$. Pro racionální kapacity nám Fordův-Fulkersonův algoritmus takový tok (maximální) a řez (minimální) vydá. Jak je to ale s reálnými kapacitami? Využijeme tvrzení, že maximální tok existuje vždy. Pak můžeme spustit Fordův-Fulkersonův algoritmus rovnou na tento maximální tok (místo nulového). Algoritmus se nutně ihned zastaví, neboť neexistuje cesta, která by měla alespoň jednu hranu s kladnou rezervou. A my víme, že pokud se algoritmus zastaví, tak vydá minimální řez. Proto i pro síť s reálnými kapacitami platí, že existuje maximální tok f a minimální řez (A, B) a $c(A, B) = |f|$. ♡

Věta: Síť s celočíselnými kapacitami má aspoň jeden z maximálních toků celočíselný a Fordův-Fulkersonův algoritmus takový tok najde.

Důkaz: Když dostane Fordův-Fulkersonův algoritmus celočíselnou síť, tak najde maximální tok a ten bude zase celočíselný (algoritmus nikde nedělí). ♡

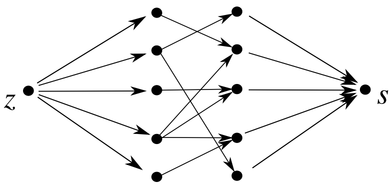
To, že umíme najít celočíselné řešení není úplně samozřejmé. (U jiných problémů takové štěstí mít nebudeme.) Ukažme si rovnou jednu aplikaci, která právě celočíselný tok využije.

Aplikace: Hledání maximálního párování v bipartitních grafech.

Definice: Množina hran $F \subseteq E$ se nazývá *párování*, jestliže žádné dvě hrany této množiny nemají společný ani jeden vrchol. Neboli $\forall e, f \in F : e \cap f = \emptyset$.

Definice: Párování je maximální, pokud obsahuje největší možný počet hran.

Mějme bipartitní graf $G = (V, E)$. V něm hledáme maximální párování. Sestrojme si síť takovou, že vezmeme vrcholy V grafu G a přidáme k nim dva speciální vrcholy z (zdroj) a s (stok) a ze zdroje přidáme hrany do všech vrcholů levé partity a ze všech vrcholů pravé partity povedeme hrany do stoku. Všechny kapacity nastavme na 1. Hrany bipartitního grafu zorientujme z levé partity do pravé. Nyní stačí jen na tuto síť spustit Fordův-Fulkersonův algoritmus (nebo libovolný jiný algoritmus, který najde maximální celočíselný tok) a až doběhne, tak prohlásit hrany s tokem 1 za maximální párování.



Hledání maximálního párování v bipartitním grafu.

Existuje totiž bijekce mezi párováním a celočíselnými toky při zachování velikosti. Z každého toku na výše zmíněném grafu (viz obrázek) lze sestrojit párování o stejné velikosti (velikost toku zde odpovídá počtu hran bipartitního grafu, po kterých poteče 1) a naopak. Důležité je si uvědomit, že definice toku (omezení toku kapacitou a Kirchhoffovy zákony) nám zaručují, že hrany s nenulovým tokem (tedy jedničkovým) budou tvořit párování (nestane se, že by dvě hrany začínaly nebo končily ve stejném vrcholu, neboť by se nutně porušila jedna ze dvou podmínek definice toku). Potom i maximální tok bude odpovídat maximálnímu párování a naopak.

V bipartitním grafu najdeme maximální párování v čase $\mathcal{O}(n \cdot (m+n))$. Fordův-Fulkersonův algoritmus stráví jednou iterací čas $\mathcal{O}(m + n)$ (za prohledání do šířky) a při jednotkových kapacitách bude iterací nejvýše n, protože každou se tok zvětší alespoň o 1 a všechny toky jsou omezené řezem kolem zdroje, který má kapacitu nejvýše n. Výsledná časová složtost hledání maximálního párování bude tedy $\mathcal{O}(n \cdot (m + n))$.