Diskrétní matematika

Petr Kovář
petr.kovar@vsb.cz

Vysoká škola báňská – Technická univerzita Ostrava

zimní semestr 2022/2023
DiM 470-2301/01, 470-2301/03*, 470-2301/05

O tomto souboru

Tento soubor je zamýšlen především jako pomůcka pro přednášejícího. Řadu důležitých informací v souboru nenajdete, protože přednášející je říká, ukazuje, případně maluje na tabuli. Přednášky jsou na webu k dispozici, aby studenti mohli snadno dohledat probíraná témata z přednášek, které zameškali.

Pro samostatné studium doporučuji skripta:

- M. Kubesa: Základy diskrétní matematiky, výukový text
- P. Kovář: Úvod do teorie grafů, výukový text

Pro přípravu ke zkoušce a písemkám doporučuji cvičebnici:

- P. Kovář: Cvičení z diskrétní matematiky, sbírka příkladů

Vše na http://homel.vsb.cz/~kov16/predmety_dm.php

Přehled přednášky

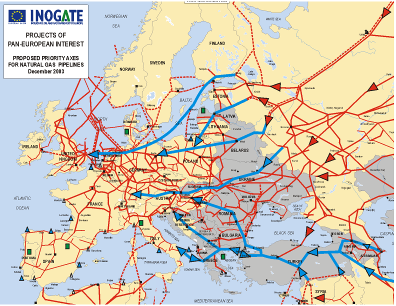
Kapitola Toky v sítích

- motivace
- definice sítě
- hledání největšího toku
- zobecnění sítí
- další aplikace

Kapitola Toky v sítích

Teorie grafů hraje klíčovou roli při řešení řady síťových úloh. Máme dánu síť (počítačovou síť, produktovod, …), kde hrany reprezentují spojení a vrcholy jsou například křižovatky nebo switche.

Je přirozené, že hrany a vrcholy mají omezenou kapacitu (kapacita = číslo + fyzikální jednotka).



Otázka

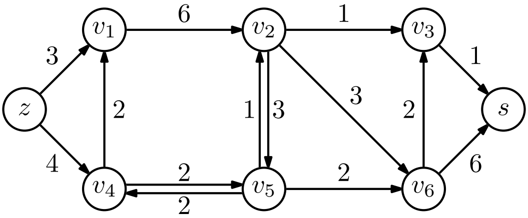
Jaký největší objem látky nebo dat můžeme po síti (s danými omezeními) dopravit z vrcholu z (zdroj) do vrcholu s (stok).

Síť je graf, pro který stanovíme kapacity hran a zvolíme zdroj a stok.

Definice

Síť je čtveřice $S = (G, z, s, w)$, kde

- G je orientovaný graf,
- vrcholy $z \in V(G)$, $s \in V(G)$ nazýváme zdroj a stok v síti S,
- $w : E(G) \to \mathbb{R}^+$ je kladné ohodnocení hran, tzv. kapacita hran.



Otázka

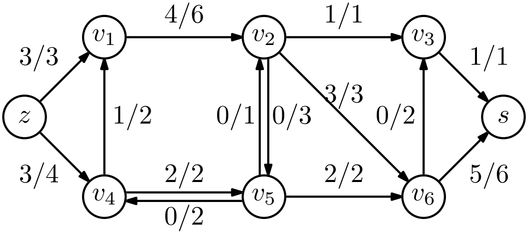
Jaké je největší objem, který můžeme v G přepravit ze zdroje z ke spotřebiteli ve stoku s, jestliže dodržíme stanovené *maximální kapacity* hran?

Složitější síť může mít

- kapacity vrcholů
- více zdrojů a stoků
- více produktů

Někdy můžeme úlohu převézt na základní úlohu. Později ukážeme jak.

Všimněte si, že při největším možném toku nemusí být využita kapacita některých hran. Proto zavedeme pojem toku (tok/kapacita).



Poznámka

Kapacita hrany nemusí nutně odpovídat fyzikální kapacitě (šířka vozovky, maximální počet automobilů za minutu, tloušťka potrubí, odpor vodiče...) Využijeme terminologii z dopravy tekutin: množství, které „teče“ hranou, „přitéká“ do vrcholu a z vrcholu „odtéká“.

Symbol $e \to v$ označuje množinu všech hran *přicházejících* do v a symbol $e \leftarrow v$ označuje množinu všech hran *vycházejících* z v.

Definice

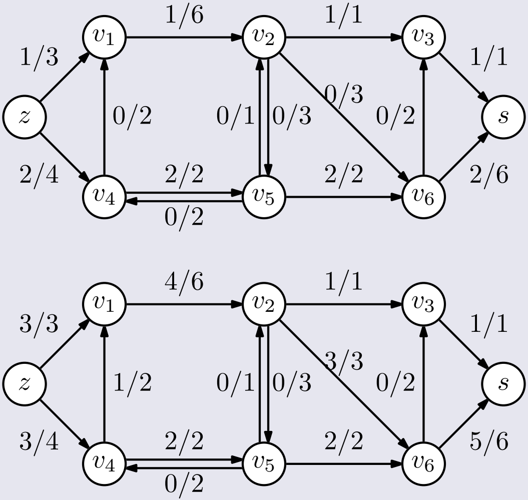
Tok v síti $S = (G, z, s, w)$ je funkce $f : E(G) \to \mathbb{R}_0^+$, kde

- nepřekročíme kapacity hran: $\forall e \in E(G) : 0 \leq f(e) \leq w(e)$,
- platí zákon kontinuity: $\forall v \in V(G), v \neq z, s : \sum_{e \to v} f(e) = \sum_{e \leftarrow v} f(e)$.

Velikost toku f je

$$\|f\| = \sum_{e \leftarrow z} f(e) - \sum_{e \to z} f(e).$$

Příklad



Tok a největší tok v síti (G, z, s, w).

Zdroj a stok jsou významné vrcholy v síti. Mají jiný význam než „křižovatky“ produktovodů. Neplatí pro ně zákony kontinuity!

- Ze zdroje „teče“ více než do zdroje,
- do stoku „teče“ více než ze stoku.

Rozdíl musí být pro oba vrcholy stejný.

Lemma

Označíme f_z rozdíl součtu toků na odchozích hranách a na příchozích hranách do zdroje. Dále označíme f_s rozdíl součtu toků na odchozích hranách a na příchozích hranách do stoku. Platí $f_z = -f_s$.

Důkaz

$$0 = \sum_e (f(e) - f(e)) = \sum_v \sum_{e \leftarrow v} f(e) - \sum_v \sum_{e \rightarrow v} f(e) = \sum_{v \in \{z,s\}} \left(\sum_{e \leftarrow v} f(e) - \sum_{e \rightarrow v} f(e) \right) .$$

Dvojité sumy nabývají stejných hodnot pro všechny vrcholy s výjimkou z, s.

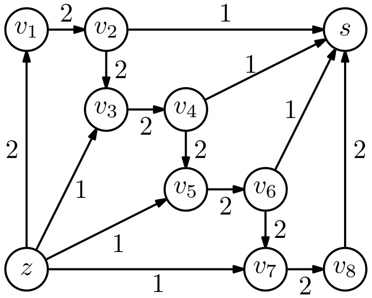
$$\left(\sum_{e \leftarrow z} f(e) - \sum_{e \rightarrow z} f(e) \right) = - \left(\sum_{e \leftarrow s} f(e) - \sum_{e \rightarrow s} f(e) \right) . \qquad \square$$

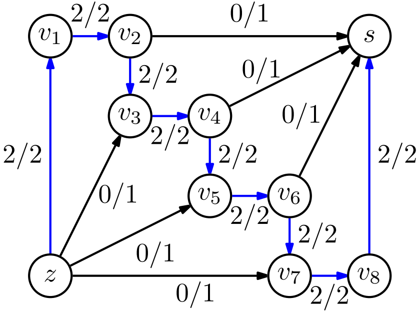
Poznámka

Hledání největšího toku

Úkolem je najít co největší možný tok z vrcholu z do vrcholu s v dané síti (G) s ohodnocením w.

Hladový algoritmus nevede k nejlepšímu řešení! (viz obrázek).





Užitím hladového algoritmu najdeme tok o velikosti 2.

Tok nemůžeme zvýšit přidáním nějaké cesty s nenulovým tokem, avšak to není největší tok, neboť existuje tok velikosti 5.

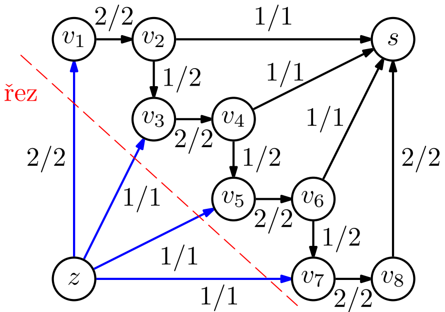
Definice

Řez C v síti $S = (G, z, s, w)$ je taková podmnožina hran $C \subseteq E(G)$, že v grafu $G - C$ nezůstane žádná orientovaná cesta ze zdroje z do stoku s. Kapacitu (nebo velikostí) řezu C značíme $\|C\|$ a definujeme ji jako součet kapacit všech hran řezu C, tj. $\|C\| = \sum_{e \in C} w(e)$.

Věta

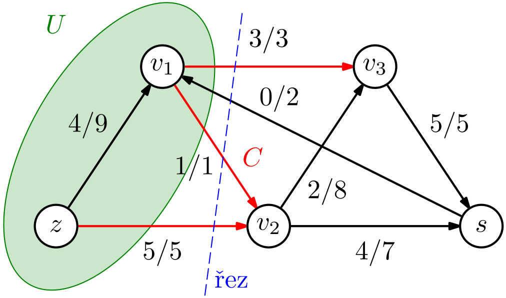
Velikost největšího toku v síti je rovna kapacitě nejmenšího řezu.

Důkaz později...



Tok velikosti 5 *a řez s kapacitou* 5.

Pozor, řez je nutno chápat správně!



Řez obsahuje jen odchozí hrany z množiny U.

Věta je dobrou charakterizací největšího toku:

Tok o velikosti x je největší, jestliže najdeme řez o velikost x.

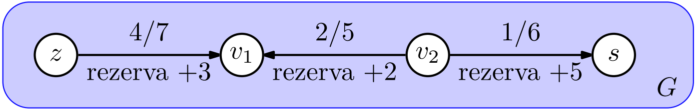
Definice

Máme dánu síť $S = (G, z, s, w)$ a v ní tok f. Nenasycená cesta v síti S je neorientovaná cesta $e_1, e_2, \ldots, e_m$ v G z vrcholu u do vrcholu v (obvykle ze zdroje z do stoku s), kde

- tok $f(e_i) < w(e_i)$ pro hrany e_i „ve směru" z u do v,
- tok $f(e_i) > 0$ (je nenulový) pro hrany e_i v opačném směru.

Hodnotě $w(e_i) - f(e_i)$ pro hranu e_i ve směru z vrcholu u do vrcholu v říkáme rezerva kapacity hrany e_i a stejně říkáme hodnotě $f(e_i)$ pro hrany e_i v opačném směru.

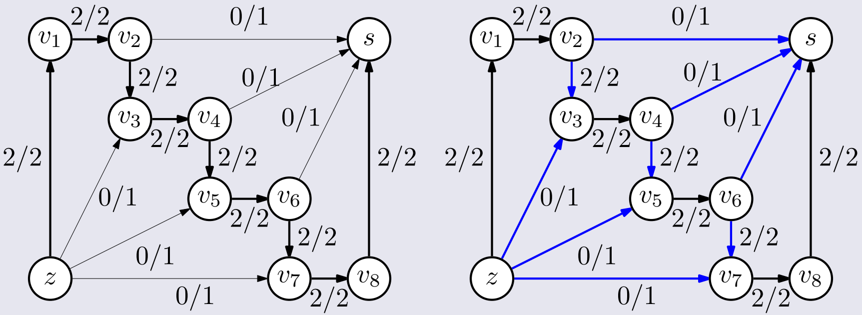
Nenasycená cesta je cesta s kladnými rezervami δ kapacit všech hran.



Cesta s rezervou kapacit $\delta = 2$.

Příklad

Najděte nenasycené cesty v síti na obrázku.



Příklad tří nenasycených cest v síti.

Algoritmus Ford–Fulkersonův algoritmus

```
vstup < síť S = (G,z,s,w);
výchozí tok f je pro každou hranu nulový;

do {
   prohledáním grafu najdeme množinu U vrcholů G,
   dosažitelných ze z po nenasycených cestách;
   if (s náleží U) {
      P = (výše nalezená) nenasycená cesta v S ze z do s;
      zvětšíme tok f o minimální rezervu kapacit hran P;
} while (s náleží U);
výstup:  vypíšeme největší tok f;
výstup:  nejm. řez je množina hran vedoucích z U do V(G)-U.
```

Důkaz Důkaz správnosti algoritmu provedeme přímo.

Je zřejmé, že musí platit $\|f\| \leq \|C\|$.
Jestliže po skončení algoritmu budeme mít tok f a najdeme v síti S řez o stejné velikosti $\|C\| = \|f\|$, je zřejmé, že jsme našli největší možný tok v síti S. Zároveň tím dokážeme platnost Věty o největším toku a nejmenším řezu.
Stačí dokázat, že po skončení algoritmu dostaneme rovnost $\|f\| = \|C\|$, kde C je řez mezi U a zbytkem grafu G.

Předpokládejme, že máme tok f v S bez nenasycené cesty ze z do s. To znamená, že množina U z algoritmu neobsahuje vrchol s (je dosažitelný jen po nenasycené cestě).

Protože z U nevedou žádné nenasycené cesty (ani hrany), má každá hrana $e \leftarrow U$ (odcházející z U) plný tok $f(e) = w(e)$ a každá hrana $e \rightarrow U$ (přicházející do U) tok $f(e) = 0$. Velikost toku f ze z do s je

$$\|f\| = \sum_{e \leftarrow U} f(e) - \sum_{e \rightarrow U} f(e) = \sum_{e \leftarrow U} f(e) - 0 = \sum_{e \in C} w(e) = \|C\| \,.$$

To je dokazované tvrzení. □

Uvedený algoritmus najde největší tok s velikostí $\|f\|$. Navíc po skončení běhu algoritmu snadno sestavíme nejmenší řez s kapacitou $\|C\|$.
Uvedeme jedno pěkné pozorování:

Důsledek

Jsou-li kapacity hran sítě S *přirozená* (resp. nezáporná celá) čísla, tak největší tok v síti bude také přirozené (resp. nezáporné celé) číslo.

Největší tok v síti vždy plně využije kapacity hran nějakého řezu a proto je velikost toku součtem ohodnocení hran řezu. Tj. jsou-li všechny kapacity celá čísla, bude i velikost toku celočíselná.

Poznámka

Pozor, vynecháme-li požadavek, aby kapacity hran byla nezáporná celá čísla, tak je možno sestavit příklady jednoduchých sítí s iracionálními kapacitami, pro které Ford-Fulkersonův algoritmus neskončí po *konečném* počtu kroků.

Dokonce postupné iterace ani nemusí konvergovat k největšímu toku.

Zobecnění sítí

Uvedenou úlohu hledání největšího toku můžeme zobecnit a

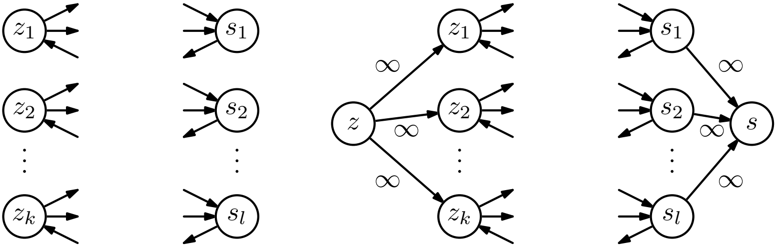
1. více zdrojů a stoků v síti,
2. dovolíme neorientované hrany v síti,
3. přidat kapacity vrcholů,
4. přepravovat jednou sítí více produktů současně,
5. stanovit minimální kapacity hran (něco musí téct).

Místo abychom hledali nové algoritmy nebo modifikovali uvedený algoritmus, tak ukážeme, jak modifikovat graf sítě.

Složitější úlohu tak převedeme na základní úlohu řešenou pomocí Ford-Fulkersonova algoritmu.

1) Více zdrojů a stoků

Vyskytuje-li se v síti více zdrojů nebo stoků, můžeme úlohu jednoduše převést na úlohu s jediným zdrojem a jediným stokem.

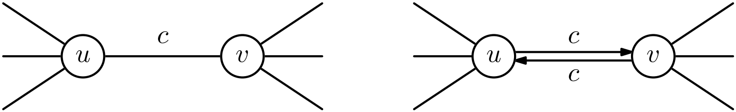


Můžeme dokonce stanovit kapacity jednotlivých zdrojů a stoků. Stačí stanovit jiné kapacity než ∞.

2) Neorientované hrany

U některých praktických aplikací sítí je orientace hran dána (například kanalizační potrubí, jízdní pruhy v silniční síti, ...) U jiných však orientace může být libovolná (informační nebo počítačové sítě).
Ford-Fulkersonův algoritmus je navržen pro orientované grafy.

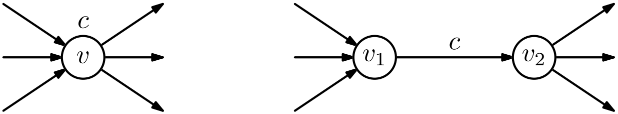
Neorientovanou hranu v síti můžeme snadno nahradit dvojicí opačně orientovaných hran se stejnou kapacitou.



Pozor, po skončení algoritmu je dán rozdílem hodnot toků na obou orientovaných hranách.

3) Kapacity vrcholů

Je přirozené, že nejen hrany, ale také místa křížení mají omezenou kapacitu.
Síť s kapacitami hran i vrcholů můžeme snadno převést na síť, ve které jsou ohodnocené pouze hrany.



Každý vrchol se stanovenou kapacitou c nahradíme dvojicí vrcholů spojenou hranou s kapacitou c (vrchol zdvojíme).

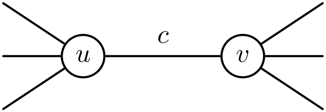
Hrany příchozí do v budou končit ve vrcholu v_1 a hrany odchozí z vrcholu v budou začínat ve vrcholu v_2.

4) Víceproduktové toky

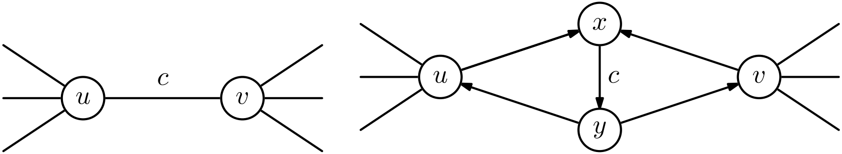
Pro víceproduktové toky je úloha složitější. Algoritmus pro hledání největšího víceproduktového (vícekomoditního) toku je nad rámec kurzu.

Ukážeme alespoň, jak převedeme neorientovaný graf na orientovaný s předepsanou kapacitou hrany.

Pozor: nestačí dvě opačně orientované hrany.



Nesmí se stát, aby jeden produkt putoval jedním směrem a jiný produkt opačným směrem a v součtu by tok překročil kapacitu hrany!



V uvedeném nahrazení zajistíme dodržení kapacit i pro různé komodity.

5) Minimální kapacity hran

Pokud kromě maximálních kapacit hran stanovíme i minimální kapacity, tj. požadavek, aby tok hranou měl nějaký nenulový tok, nemusí mít úloha řešení.

Tato úloha přesahuje rámec kurzu.

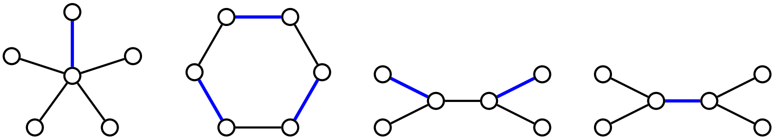
Pro zájemce: J. Demel, Grafy, SNTL, Praha, (1989).

Párování v bipartitních grafech

Rozmanitost aplikací algoritmu pro hledání největšího toku je překvapivá. Ukážeme, jak převedeme úlohu hledání největšího párování na úlohu hledání největšího toku.

Definice

Párování v (bipartitním) grafu G je podmnožina nezávislých hran $M \subseteq E(G)$ (žádné dvě hrany z M nemají společný koncový vrchol).



Zatímco v prvním grafu (hvězdě) párování má jedinou hranu, pro druhý graf existuje párování pokrývající všechny vrcholy.

Metoda Nalezení největšího bipartitního párování

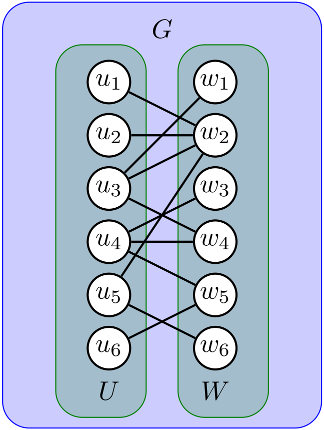
Mějme bipartitní graf G s množnou vrcholů rozdělenou do partit U, W.

1. Sestrojíme síť S: do bipartitního grafu přidáme zdroj z a stok s, všechny vrcholy z U spojíme se zdrojem z a všechny vrcholy z W spojíme s stokem s, dále všechny hrany sítě S orientujeme od zdroje do stoku a přiřadíme jim kapacity 1,
2. najdeme (celočíselný) největší tok Ford-Fulkersonovým algoritmem,
3. největší párování M v grafu G obsahuje všechny ty hrany původního grafu G, které mají nenulový tok v síti S.

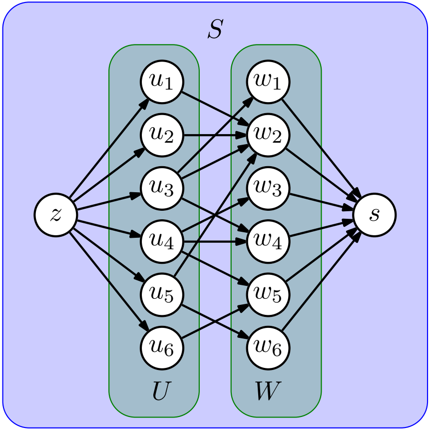
Důkaz Podle uvedeného důsledku bude největší tok celočíselný, tj. každou hranou „poteče“ tok 0 nebo 1.

Do každého vrcholu v U vede pouze jedna hrana s kapacitou 1, proto bude tento vrchol pouze v jedné hraně párování. Podobně pro vrcholy množiny W. Tím budou vybrány hrany párování a podle zadaných kapacit nebudou sdílet koncové vrcholy.

Párování je největší, protože z každého párování lze získat příslušný tok a větší než nalezený tok v S neexistuje. □

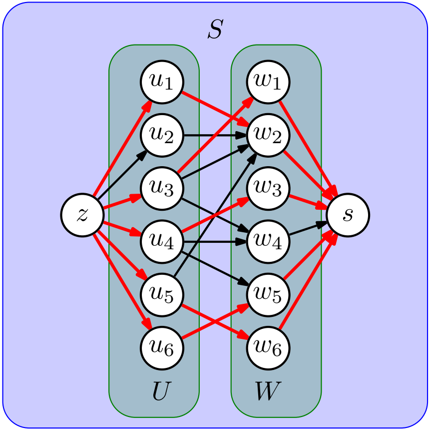


Máme dán bipartitní graf G.

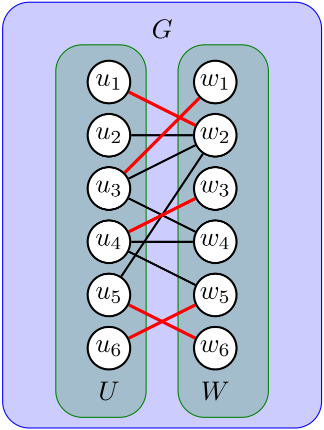


Sestrojíme příslušnou síť S:

- do grafu přidáme zdroj z a stok s,
- všechny vrcholy z U spojíme se z a všechny vrcholy z W spojíme s s,
- všechny hrany sítě S orientujeme od zdroje do stoku; přiřadíme jim kapacity 1.



Najdeme největší tok v síti S pomocí Ford-Fulkersonova algoritmu. Tok bude podle uvedeného důsledku celočíselný.



Hledané párování obsahuje jen ty hrany grafu G, které mají nenulový tok.

Vyšší grafová souvislost Již dříve jsme zavedli pojem hranové a vrcholové souvislosti. Bez důkazu jsme uvedli Mengerovy věty:

Věta (Mengerovy věty)

Netriviální graf G je hranově k-souvislý právě tehdy, když mezi libovolnými dvěma vrcholy existuje alespoň k po dvou hranově disjunktních cest (vrcholy mohou být sdílené).
Netriviální graf G je vrcholově k-souvislý právě tehdy, když mezi libovolnými dvěma vrcholy existuje alespoň k po dvou interně disjunktních cest (koncové vrcholy jsou společné).

Nyní pomocí algoritmu hledání největšího toku větu dokážeme.

Nejprve s využitím definice hranové souvislosti přeformulujeme:

Lemma

Nechť u, v jsou dva různé vrcholy grafu G a $k > 0$ je přirozené číslo. Mezi vrcholy u a v v grafu G existuje alespoň k hranově disjunktních cest právě tehdy, když po odebrání libovolných $k-1$ hran z grafu G zůstanou vrcholy u a v ve společné komponentě souvislosti.

Důkaz

"$\Rightarrow$" Plyne přímo z definice hranové k-souvislosti grafu.

"$\Leftarrow$" Mějme graf G a v něm zvolené dva vrcholy u a v. Označíme vrchol u za zdroj a vrchol v za stok a každé hraně přiřadíme kapacitu 1. Pomocí Ford-Fulkersonova algoritmu najdeme největší tok z u do v.

Velikost toku je alespoň k, jinak by kapacita nejmenšího řezu by byla také menší než k. Odebráním hran řezu dostaneme nesouvislý graf, přičemž odebraných hran je méně než k, což podle předpokladu není možné.

To znamená, že hrany s tokem 1 tvoří různé (hranově disjunktní) cesty z u do v. (První Mengerovu větu dokážeme podobně.) $\square$

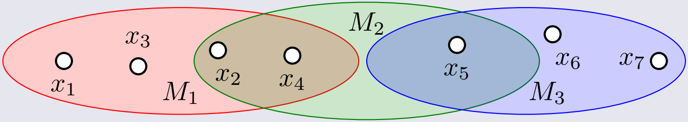
Systém různých reprezentantů

Definice

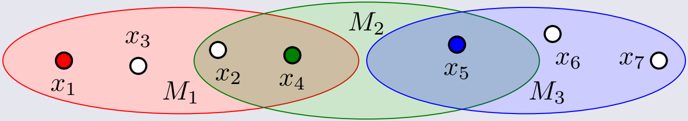
Nechť $M_1, M_2, \ldots, M_k$ jsou neprázdné množiny. Systémem různých reprezentantů množin $M_1, M_2, \ldots, M_k$ rozumíme takovou posloupnost *různých* prvků $(m_1, m_2, \ldots, m_k)$, že $m_i \in M_i$ pro každé $i = 1, 2, \ldots, k$.

Příklad

Najděte systém různých reprezentantů v daném systému množin.



Jedno možné řešení.



Důležitým a dobře známým výsledkem je Hallova věta:

Věta (Hallova věta)

Nechť $M_1, M_2, \ldots, M_k$ pro $k > 0$ jsou neprázdné množiny. Pro tyto množiny existuje systém různých reprezentantů právě tehdy, když platí

$$\forall J \subset \{1, 2, \ldots, k\} : \left|\bigcup_{j \in J} M_j\right| \geq |J|,$$

neboli pokud sjednocení libovolné skupiny j množin má alespoň j prvků (tedy alespoň tolik prvků, kolik množin je sjednoceno).

Hallova věta udává nutnou a dostatečnou podmínku existence různých reprezentantů v daném systému množin.

Praktické nalezení systému reprezentantů je náročné, ale někdy můžeme vhodnou volbou množin existenci vyvrátit.

Poznámka

Hallově větě se v anglické literatuře říká „Marriage Theorem“.

Důkaz Hallovy věty

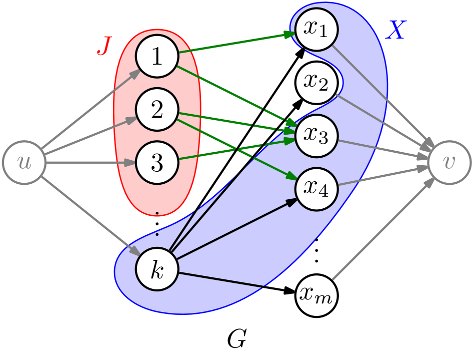
Důkaz implikace „$\Rightarrow$" je snadný. Myšlenka důkazu implikace „$\Leftarrow$":

Označme $x_1, x_2, \dots, x_m$ po řadě všechny prvky ve sjednocení $M_1 \cup M_2 \cup \dots \cup M_k$. Definujeme bipartitní graf G na množině vrcholů $\{1, 2, \dots, k\} \cup \{x_1, x_2, \dots, x_m\} \cup \{u, v\}$. Navíc přidáme hrany $\{u, i\}$ pro $i = 1, 2, \dots, k$, $\{x_j, v\}$ pro $j = 1, 2, \dots, m$ a hrany $\{i, x_j\}$ pro $x_j \in M_i$.

Konstrukce sítě S je obdobná konstrukci sítě v uvedené metodě nalezení největšího párování. Cesta mezi u a v má tvar u, i, x_j, v, ukazuje reprezentanta množiny $x_j \in M_i$. Systém různých reprezentantů odpovídá k disjunktním cestám mezi u a v.

Nechť X je nyní libovolná minimální množina vrcholů v G, po jejímž odebrání z grafu nezůstane žádná cesta mezi u a v. Podle Lemmatu mají naše množiny systém různých reprezentantů právě tehdy, když každá taková oddělující množina X má aspoň k prvků.

Položme $J = \{1, 2, \dots, k\} \setminus X$.



Pak každá hrana z J vede (hrany z u neuvažujeme) do vrcholů množiny $X \cap \{x_1, \ldots, x_m\}$ (aby nevznikla cesta mezi u, v), a proto $\left|\bigcup_{j \in J} M_j\right| = |X \cap \{x_1, \ldots, x_m\}| = |X| - |X \cap \{1, \ldots, k\}| = |X| - k + |J|$.

$$\left|\bigcup_{j \in J} M_j\right| - |J| = |X| - k.$$

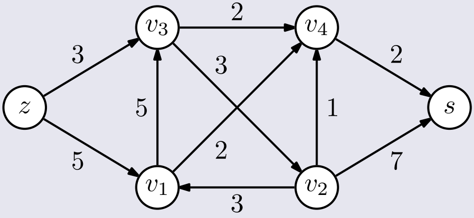
Vidíme, že $|X| \geq k$ pro všechny volby oddělující množiny X právě tehdy, když $\left|\bigcup_{j \in J} M_j\right| \geq |J|$ pro všechny J, což je dokazované tvrzení. $\square$

Příklad

Hledání největšího toku a nejmenšího řezu nestihneme probrat na cvičení. Následují dva ukázkové příklady.

Příklad

Jaký je největší tok v síti (G, z, s, w)?
A kde je nejmenší řez?

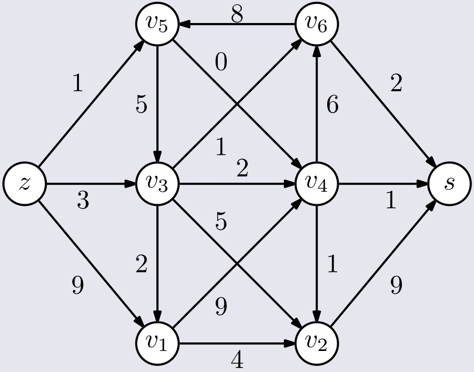


Síť (G, z, s, w).

Poslední příklad

Příklad

Jaký je největší tok v síti (G, z, s, w)?
A kde je nejmenší řez?



Síť (G, z, s, w).

Konec

Děkuji za pozornost

Nezapomeňte, prosím, na (slovní) hodnocení předmětu.

Zkouškové termíny

- středa 20.12. 9:00 – 10:30 EC3 (9:45–11:15 na EC2)
- čtvrtek 4.1. 9:00 – 10:30 EC3 (a případně EC2 od 9:45)
- středa 10.1. 9:00 – 10:30 EC3 (a případně EC2)
- úterý 16.1. 9:00 – 10:30 EC3

Mnoho zdaru u zkoušky!