1. Toky, řezy a Fordův-Fulkersonův algoritmus

V této kapitole nadefinujeme toky v sítích, odvodíme základní věty o nich a také Fordův-Fulkersonův algoritmus pro hledání maximálního toku. Také ukážeme, jak na hledání maximálního toku převést problémy týkající se řezů, separátorů a párování v bipartitních grafech. Další tokové algoritmy budou následovat v příštích kapitolách.

Toky v sítích

Intuitivní pohled: síť je systém propojených potrubí, který přepravuje tekutinu ze zdroje s (source) do spotřebiče t (target), přičemž tekutina se nikde mimo tato dvě místa neztrácí ani nevzniká.

Definice:

- *Síť* je uspořádaná pětice (V, E, s, t, c), kde:
  - (V, E) je orientovaný graf,
  - $s \in V$ *zdroj,*
  - $t \in V$ *spotřebič* neboli *stok* a
  - $c : E \to \mathbb{R}$ funkce udávající nezáporné *kapacity* hran.
- *Ohodnocení* hran je libovolná funkce $f : E \to \mathbb{R}$. Pro každé ohodnocení f můžeme definovat:

  $$f^+(v) = \sum_{e=(\cdot,v)} f(e), \quad f^-(v) = \sum_{e=(v,\cdot)} f(e), \quad f^\Delta(v) = f^+(v) - f^-(v)$$

  [co do vrcholu přiteče, co odteče a jaký je v něm přebytek].
- *Tok* je ohodnocení $f : E \to \mathbb{R}$, pro které platí:
  - $\forall e : 0 \le f(e) \le c(e)$, *(dodržuje kapacity)*
  - $\forall v \ne s, t : f^\Delta(v) = 0$. *(Kirchhoffův zákon)*
- *Velikost toku:* $|f| = -f^\Delta(s)$.
- *Řez (tokový):* množina vrcholů $C \subset V$ taková, že $s \in C$, $t \notin C$. Řez můžeme také ztotožnit s množinami hran $C^- = E \cap (C \times \overline{C})$ [těm budeme říkat hrany zleva doprava] a $C^+ = E \cap (\overline{C} \times C)$ [hrany zprava doleva].
- *Kapacita řezu:* $|C| = \sum_{e \in C^-} c(e)$ (bereme v úvahu jen hrany zleva doprava).
- *Tok přes řez:* $f^+(C) = \sum_{e \in C^+} f(e)$, $f^-(C) = \sum_{e \in C^-} f(e)$, $f^\Delta(C) = f^+(C) - f^-(C)$.
- *Cirkulace* je tok nulové velikosti, čili f takové, že $f^\Delta(v) = 0$ pro všechna v.

Základním problémem, kterým se budeme zabývat, je hledání *maximálního toku* (tedy toku největší možné velikosti) a *minimálního řezu* (řezu nejmenší možné kapacity).

Větička: V každé síti existuje maximální tok a minimální řez.

Důkaz: Existence minimálního řezu je triviální, protože řezů v každé síti je konečně mnoho; pro toky v sítích s reálnými kapacitami to ovšem není samozřejmost a je k tomu potřeba trocha matematické analýzy (v prostoru všech ohodnocení hran tvoří toky kompaktní množinu, velikost toku je lineární funkce, a tedy i spojitá, pročež nabývá maxima). Pro racionální kapacity dostaneme tuto větičku jako důsledek analýzy Fordova-Fulkersonova algoritmu. ♡

Pozorování: Kdybychom velikost toku definovali podle spotřebiče, vyšlo by totéž. Platí totiž:

$$f^\Delta(s) + f^\Delta(t) = \sum_v f^\Delta(v) = \sum_e f(e) - f(e) = 0$$

(první rovnost plyne z Kirchhoffova zákona – všechna ostatní $f^\Delta(v)$ jsou nulová; druhá pak z toho, že každá hrana přispěje k jednomu $f^+(v)$ a k jednomu $f^-(v)$). Proto je $|f| = -f^\Delta(s) = f^\Delta(t)$.

Stejně tak můžeme velikost toku změřit na libovolném řezu:

Lemma: Pro každý řez C platí, že $|f| = -f^\Delta(C) \le |C|$.

Důkaz: První část indukcí: každý řez můžeme získat postupným přidáváním vrcholů do triviálního řezu $\{s\}$ [čili přesouváním vrcholů zprava doleva], a to, jak ukáže jednoduchý rozbor případů, nezmění f^Δ. Druhá část: $-f^\Delta(C) = f^-(C) - f^+(C) \le f^-(C) \le |C|$. ♡

Víme tedy, že velikost každého toku lze omezit kapacitou libovolného řezu. Kdybychom našli tok a řez stejné velikosti, můžeme si proto být jisti, že tok je maximální a řez minimální. To se nám opravdu povede, platí totiž následující věta:

Věta (Ford, Fulkerson): V každé síti je velikost maximálního toku rovna velikosti minimálního řezu.

Důkaz: Jednu nerovnost jsme dokázali v předchozím lemmatu, druhá plyne z duality lineárního programování [max. tok a min. řez jsou navzájem duální úlohy], ale k pěknému kombinatorickému důkazu půjde opět použít Fordův-Fulkersonův algoritmus. ♡

Fordův-Fulkersonův algoritmus

Nejpřímočařejší způsob, jak bychom mohli hledat toky v sítích, je začít s nějakým tokem (nulový je po ruce vždy) a postupně ho zlepšovat tak, že nalezneme nějakou nenasycenou cestu a pošleme po ní „co půjde“. To bohužel nefunguje, ale můžeme tento postup trochu zobecnit a být ochotni používat nejen hrany, pro které je $f(e) < c(e)$, ale také hrany, po kterých něco teče v protisměru a my můžeme tok v našem směru simulovat odečtením od toku v protisměru. Trochu formálněji:

Definice:

- *Rezerva* hrany $e = (v, w)$ při toku f se definuje jako $r(e) = (c(e) - f(e)) + f(e')$, kde $e' = (w, v)$. Pro účely tohoto algoritmu budeme předpokládat, že ke každé hraně existuje hrana opačná; pokud ne, dodefinujeme si ji a dáme jí nulovou kapacitu.

- *Zlepšující cesta* je orientovaná cesta, jejíž všechny hrany mají nenulovou rezervu.

Algoritmus:

1. $f \leftarrow$ *nulový tok.*
2. Dokud existuje zlepšující cesta P z s do t:
3. $\quad\varepsilon \leftarrow \min_{e \in P} r(e)$.
4. $\quad$Zvětšíme tok f podél P o ε (každé hraně $e \in P$ zvětšíme $f(e)$, případně zmenšíme $f(e')$, podle toho, co jde).

Analýza: Nejdříve si rozmysleme, že pro celočíselné kapacity algoritmus vždy doběhne: v každém kroku stoupne velikost toku o $\varepsilon \geq 1$, což může nastat pouze konečněkrát. Podobně pro racionální kapacity: přenásobíme-li všechny kapacity jejich společným jmenovatelem, dostaneme síť s celočíselnými kapacitami, na které se bude algoritmus chovat identicky a jak již víme, skončí. Pro iracionální kapacity obecně doběhnout nemusí, zkuste vymyslet protipříklad.

Uvažme nyní situaci po zastavení algoritmu. Funkce f je určitě tok, protože jím byla po celou dobu běhu algoritmu. Prozkoumejme teď množinu C vrcholů, do nichž po zastavení algoritmu vede zlepšující cesta ze zdroje. Jistě $s \in C$, $t \notin C$, takže tato množina je řez. Navíc pro každou hranu $e \in C^-$ musí být $f(e) = c(e)$ a pro každou $e' \in C^+$ je $f(e') = 0$, protože jinak by rezerva hrany e nebyla nulová. Takže $f^-(C) = |C|$ a $f^+(C) = 0$, čili $|f| = |C|$.

Našli jsme tedy k toku, který algoritmus vydal, řez stejné velikosti, a proto, jak už víme, je tok maximální a řez minimální. Tím jsme také dokázali Fordovu-Fulkersonovu větu⟨1⟩ a existenci maximálního toku. Navíc algoritmus nikdy nevytváří z celých čísel necelá, čímž získáme:

Důsledek: Síť s celočíselnými kapacitami má maximální tok, který je celočíselný.

Časová složitost F-F algoritmu může být pro obecné sítě a nešikovnou volbu zlepšujících cest obludná, ale jak dokázali Edmonds s Karpem, pokud budeme hledat cesty prohledáváním do šířky (což je asi nejpřímočařejší implementace), poběží v čase $\mathcal{O}(m^2 n)$. Pokud budou všechny kapacity jednotkové, snadno nahlédneme, že stačí $\mathcal{O}(nm)$. Edmondsův a Karpův odhad nebudeme dokazovat, místo toho si v příští kapitole předvedeme efektivnější algoritmus.

Řezy, separátory a k-souvislost

Teorie toků nám rovněž poslouží ke zkoumání násobné souvislosti grafů.

Definice: Pro každý neorientovaný graf G a libovolné jeho vrcholy s, t zavedeme:

- *st-řez* je množina hran F taková, že v grafu $G - F$ jsou vrcholy s, t v různých komponentách souvislosti.

⟨1⟩ Dokonce jsme ji dokázali i pro reálné kapacity, protože můžeme algoritmus spustit rovnou na maximální tok místo nulového a on se ihned zastaví a vydá certifikující řez.

- *st-separátor* je množina vrcholů W taková, že $s,t \notin W$ a v grafu $G - W$ jsou vrcholy s,t v různých komponentách souvislosti.
- *Řez* je množina hran, která je xy-řezem pro nějakou dvojici vrcholů x,y.
- *Separátor* je množina vrcholů, která je xy-separátorem pro nějakou dvojici vrcholů x,y.
- G je *hranově k-souvislý,* pokud $|V| > k$ a všechny řezy v G mají alespoň k hran.
- G je *vrcholově k-souvislý,* pokud $|V| > k$ a všechny separátory v G mají alespoň k vrcholů.

Všimněte si, že nesouvislý graf má řez i separátor nulové velikosti, takže vrcholová i hranová 1-souvislost splývají s obyčejnou souvislostí pro všechny grafy o alespoň dvou vrcholech. Hranově 2-souvislé jsou právě (netriviální) grafy bez *mostů,* vrcholově 2-souvislé jsou ty bez *artikulací.*

Pro orientované grafy můžeme st-řezy a st-separátory definovat analogicky (totiž, že po odstranění příslušné množiny hran či vrcholů neexistuje orientovaná cesta z s do t), globální řezy a separátory ani vícenásobná souvislost se obvykle nedefinují.

Poznámka: Minimální orientované st-řezy podle této definice a minimální tokové řezy podle definice ze začátku kapitoly splývají: každý tokový řez C odpovídá st-řezu stejné velikosti tvořenému hranami v C^-; naopak pro minimální st-řez musí být množina vrcholů dosažitelných z s po odebrání řezu z grafu tokovým řezem, opět stejné velikosti. [Velikost měříme součtem kapacit hran.] Dává tedy rozumný smysl říkat obojímu stejně. Podobně se chovají i neorientované grafy, pokud do „tokového“ řezu počítáme hrany v obou směrech.

Analogií toků je pak existence nějakého počtu disjunktních cest (vrcholově nebo hranově) mezi vrcholy s a t. Analogií F-F věty pak budou známé Mengerovy věty:

Věta (Mengerova, lokální hranová orientovaná): Buď G orientovaný graf a s,t nějaké jeho vrcholy. Pak je velikost minimálního st-řezu rovna maximálnímu počtu hranově disjunktních st-cest.⟨2⟩

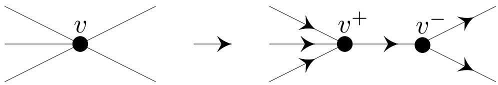
Důkaz: Z grafu sestrojíme síť tak, že s bude zdroj, t spotřebič a všem hranám nastavíme kapacitu na jednotku. Řezy v této síti odpovídají řezům v původním grafu. Podobně každý systém hranově disjunktních st-cest odpovídá toku stejné velikosti a naopak ke každému celočíselnému toku dovedeme najít systém disjunktních cest (hladově tok rozkládáme na cesty a průběžně odstraňujeme cirkulace, které objevíme). Pak použijeme F-F větu. ♡

Věta (Mengerova, lokální vrcholová orientovaná): Buď G orientovaný graf a s,t nějaké jeho vrcholy takové, že $st \notin E$. Pak je velikost minimálního st-separátoru rovna maximálnímu počtu vrcholově disjunktních st-cest.⟨3⟩

⟨2⟩ orientovaných cest z s do t

⟨3⟩ Tím myslíme cesty disjunktní až na krajní vrcholy.

Důkaz: Podobně jako v důkazu předchozí věty zkonstruujeme vhodnou síť. Tentokrát ovšem rozdělíme každý vrchol na vrcholy v^+ a v^-, všechny hrany, které původně vedly do v, přepojíme do v^+, hrany vedoucí z v povedou z v^- a přidáme novou hranu z v^+ do v^-. Všechny hrany budou mít jednotkové kapacity. Toky nyní odpovídají vrcholově disjunktním cestám, řezy v síti separátorům. ♡



Rozdělení vrcholu

Podobně dostaneme neorientované lokální věty (neorientovanou hranu nahradíme orientovanými v obou směrech) a z nich pak i globální varianty popisující k-souvislost grafů:

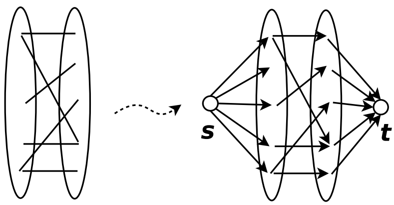
Věta (Mengerova, globální hranová neorientovaná): Neorientovaný graf G je hranově k-souvislý právě tehdy, když mezi každými dvěma vrcholy existuje alespoň k hranově disjunktních cest.

Věta (Mengerova, globální vrcholová neorientovaná): Neorientovaný graf G je vrcholově k-souvislý právě tehdy, když mezi každými dvěma vrcholy existuje alespoň k vrcholově disjunktních cest.

Maximální párování v bipartitním grafu

Jiným problémem, který lze snadno převést na hledání maximálního toku, je nalezení *maximálního párování* v bipartitním grafu, tedy množiny hran takové, že žádné dvě hrany nemají společný vrchol. Maximálním míníme vzhledem k počtu hran, nikoliv vzhledem k inkluzi.

Z bipartitního grafu $(A \cup B, E)$ sestrojíme síť obsahující všechny původní vrcholy a navíc dva nové vrcholy s a t, dále pak všechny původní hrany orientované z A do B, nové hrany z s do všech vrcholů partity A a ze všech vrcholů partity B do t. Kapacity všech hran nastavíme na jedničky:



Nyní si všimneme, že párování v původním grafu odpovídají celočíselným tokům v této síti a že velikost toku je rovna velikosti párování. Stačí tedy nalézt maximální celočíselný tok (třeba F-F algoritmem) a do párování umístit ty hrany, po kterých něco teče.

Podobně můžeme najít souvislost mezi řezy v této síti a *vrcholovými pokrytími* zadaného grafu – to jsou množiny vrcholů takové, že se dotýkají každé hrany. Tak z F-F věty získáme jinou standardní větu:

Věta (Königova): V každém bipartitním grafu je velikost maximálního párování rovna velikosti minimálního vrcholového pokrytí.

Důkaz: Pokud je W vrcholové pokrytí, musí hrany vedoucí mezi vrcholy této množiny a zdrojem a spotřebičem tvořit stejně velký řez, protože každá st-cesta obsahuje alespoň jednu hranu bipartitního grafu a ta je pokryta. Analogicky vezmeme-li libovolný st-řez (ne nutně tokový, stačí hranový), můžeme ho bez zvětšení upravit na st-řez používající pouze hrany ze s a do t, kterému přímočaře odpovídá vrcholové pokrytí stejné velikosti. ♡

Některé algoritmy na hledání maximálního párování využívají také volné střídavé cesty:

Definice: *(Volná) střídavá cesta* v grafu G s párováním M je cesta, která začíná i končí nespárovaným vrcholem a střídají se na ní hrany ležící v M s hranami mimo párování.

Všimněte si, že pro bipartitní grafy odpovídají zlepšující cesty v příslušné síti právě volným střídavým cestám a zlepšení toku podél cesty odpovídá přexorováním párování volnou střídavou cestou. Fordův-Fulkersonův algoritmus tedy lze velice snadno formulovat i v řeči střídavých cest.