KMI/ALGO3 — Algoritmy 3

8. Toky v sítích jako návrhový nástroj

Jan Konečný

Katedra informatiky, PřF UP Olomouc

Kde jsme: sylabus KMI/ALGO3

- **Co znamená navrhnout algoritmus** — specifikace, invariant, složitost
- **Hladové algoritmy** — kdy je lokální volba bezpečná
- **Hladové algoritmy na grafech** — kostry, nejkratší cesty
- **Rozděl a panuj** — rekurence, Master theorem
- **Rozděl a panuj II** — chytré rozklady
- **Dynamické programování I** — hledání stavů
- **Dynamické programování II** — vícerozměrné stavy
- **Toky v sítích** — max-flow/min-cut jako návrhový nástroj
- **Systematické prohledávání** — hrubá síla a backtracking
- **Branch-and-bound** — meze přesného řešení, P a NP
- **Adversariální návrh** — minimax, alfa-beta
- **Když přesné optimum není realistické** — aproximace, heuristiky
- **Randomizace a syntéza** — jak si vybrat metodu

Jiný způsob, jak navrhovat

Dosud jsme navrhovali algoritmus *přímo* pro zadaný problém. Tokové sítě nabízejí jinou cestu: problém převedeme na přesun abstraktního materiálu sítí s omezenými kapacitami.

Jediný algoritmus pak řeší dopravu, párování, disjunktní cesty i výběr vzájemně závislých projektů.

Budeme se ptát, jak poznat optimální tok a jak jeho hodnotu doložit krátkým certifikátem. Novinkou oproti předchozím metodám je možnost dřívější rozhodnutí rušit: zpětná reziduální hrana dovolí část toku odvolat a přesměrovat.

Tok a řez

- Tok je *přípustné řešení*: ukazuje, jak požadované množství skutečně přepravit.
- Řez je *horní mez*: ukazuje úzké hrdlo, přes které žádný tok nemůže poslat více.

Mají-li stejnou hodnotu, dostáváme současně řešení i důkaz jeho optimálnosti.

Dvojici řešení a nezávisle ověřitelné meze použijeme i v dalších přednáškách (branch-and-bound, aproximace).

Síť a přípustný tok

Toková síť: orientovaný graf $G = (V, E)$ se zdrojem s, stokem t a nezápornými celočíselnými kapacitami $c : E \to \mathbb{Z}_{\geq 0}$.

Definice 1 (Tok)

Funkce $f : E \to \mathbb{R}_{\geq 0}$ splňující

1. **kapacitní omezení:** $0 \leq f(u, v) \leq c(u, v)$;
2. **zachování toku:** pro každý $v \in V - \{s, t\}$

$$\sum_{(u,v)\in E} f(u, v) = \sum_{(v,w)\in E} f(v, w).$$

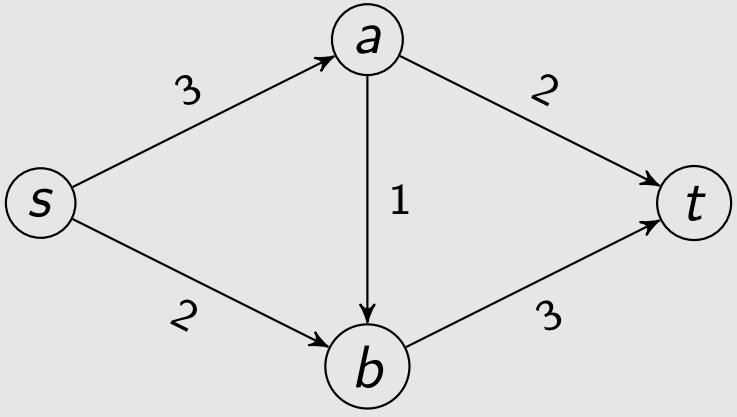
Hodnota toku je *čistý odtok ze zdroje*

$$|f| = \sum_{(s,v)\in E} f(s, v) - \sum_{(v,s)\in E} f(v, s).$$

Racionální kapacity převedeme na celé pronásobením jmenovatelem.
U iracionálních se základní algoritmus nemusí ani zastavit.

Malá síť

Příklad 2



Toky $f(s,a) = 3$, $f(s,b) = 2$, $f(a,b) = 1$, $f(a,t) = 2$, $f(b,t) = 3$ splňují kapacity i zachování. Hodnota je 5.

Zdá se, že stačí hledat cesty ze s do t a posílat po nich tok. Volba nevhodné cesty ale může později překážet — potřebujeme umět část staršího toku vrátit.

Reziduální síť

Definice 3 (Reziduální síť)

G_f má stejné vrcholy jako G; pro každou původní hranu $e = (u, v)$ obsahuje

- dopřednou hranu $u \to v$ s kapacitou $c_f(u, v) = c(u, v) - f(u, v)$;
- zpětnou hranu $v \to u$ s kapacitou $c_f(v, u) = f(u, v)$.

Hrany s nulovou reziduální kapacitou vynecháváme.

Definice 4 (Zvětšující cesta)

Orientovaná cesta ze s do t v G_f. Její rezerva (úzké hrdlo) je $\Delta = \min_{e \in P} c_f(e)$.

Po cestě pošleme Δ jednotek: na dopředné hraně tok zvýšíme, na zpětné o Δ *snížíme*.

Co zpětná hrana znamená

Příklad 5

Má-li původní hrana (u, v) kapacitu 7 a tok 5:

- dopředná rezerva 2 — tolik lze ještě přidat ve směru $u \to v$;
- zpětná rezerva 5 — tolik dříve poslaného toku smíme *zrušit*.

Zpětná hrana neznamená, že síť umí přepravovat materiál opačně; je to jen možnost opravit starší rozhodnutí.

Zvětšení zachová přípustnost

Lemma 6 (Invariant zvětšení)

Zvětšení přípustného toku podél zvětšující cesty o její rezervu vytvoří opět přípustný tok a zvýší jeho hodnotu o Δ.

Důkaz.

BÚNO je cesta jednoduchá (opakuje-li vrchol, úsek mezi návštěvami vypustíme). Rezerva nepřekročí žádnou dopřednou volnou kapacitu ani množství toku rušeného na zpětné hraně, takže *kapacitní omezení* platí.

Zachování toku: každý vnitřní vrchol cesty leží na jedné vstupní a jedné výstupní reziduální hraně.

vstupní dopředná	přítok do v $+\Delta$
vstupní zpětná	odtok z v $-\Delta$
výstupní dopředná	odtok z v $+\Delta$
výstupní zpětná	přítok do v $-\Delta$

Vstupní hrana tedy rozdíl „přítok − odtok“ zvýší o Δ, výstupní jej o Δ sníží — dohromady beze změny.

Fordův–Fulkersonův algoritmus

Začneme nulovým tokem. Dokud reziduální síť obsahuje cestu P ze s do t:

1. najdeme rezervu Δ cesty;
2. zvýšíme tok na dopředných hranách P o Δ;
3. snížíme tok na původních hranách, které P používá *zpětně*, o Δ.

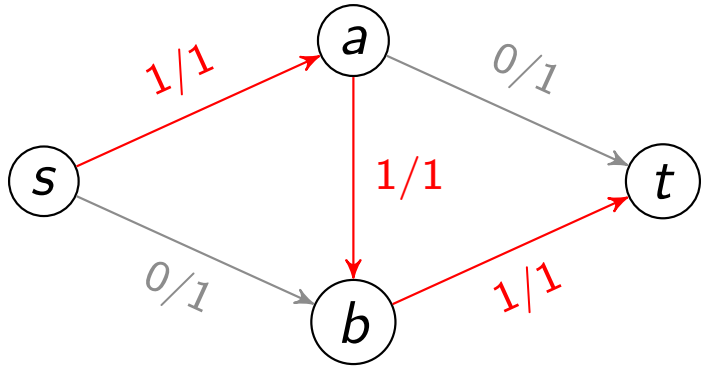
Když žádná zvětšující cesta neexistuje, vrátíme aktuální tok.

Lemma o zvětšení je invariantem algoritmu: tok je po každé iteraci přípustný a jeho hodnota roste.

Zbývá ukázat, že *zastavení znamená optimálnost*. K tomu potřebujeme řezy.

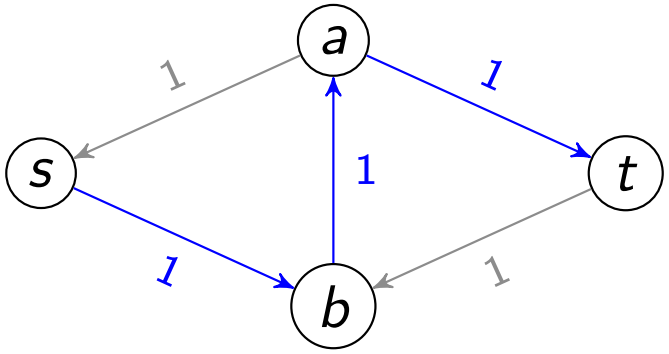
Zpětná hrana v akci: diamant se zkratkou

Vrcholy s, a, b, t; hrany $(s, a), (s, b), (a, t), (b, t)$ a zkratka (a, b). Všechny kapacity 1. Schválně nešikovně zvolíme první cestu $s \to a \to b \to t$:

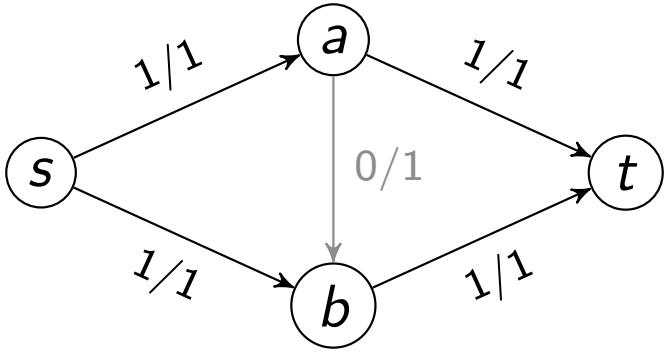


Hodnota toku je 1. Bez zpětných hran bychom tu uvízli.

Reziduální síť a oprava



Jediná zvětšující cesta je $s \to b \to a \to t$ — prostřední krok $b \to a$ je *zpětná* hrana. Pošleme $\Delta = 1$ a tok na zkratce (a, b) o jedna *snížíme*:



Hodnota 2, zkratka je opět prázdná.

Celý běh na naší malé síti

Kapacity: (s,a):3, (s,b):2, (a,b):1, (a,t):2, (b,t):3. Začínáme nulovým tokem.

kolo	zvětšující cesta	Δ	tok po kroku (sa, sb, ab, at, bt)
0	—	—	$(0,0,0,0,0)$, $\lvert f\rvert = 0$
1	$s \to a \to t$	$\min\{3,2\} = 2$	$(2,0,0,2,0)$, $\lvert f\rvert = 2$
2	$s \to b \to t$	$\min\{2,3\} = 2$	$(2,2,0,2,2)$, $\lvert f\rvert = 4$
3	$s \to a \to b \to t$	$\min\{1,1,1\} = 1$	$(3,2,1,2,3)$, $\lvert f\rvert = 5$

Celý běh na naší malé síti

Kapacity: (s,a):3, (s,b):2, (a,b):1, (a,t):2, (b,t):3. Začínáme nulovým tokem.

kolo	zvětšující cesta	Δ	tok po kroku (sa, sb, ab, at, bt)
0	—	—	$(0,0,0,0,0)$, $\lvert f\rvert = 0$
1	$s \to a \to t$	$\min\{3,2\} = 2$	$(2,0,0,2,0)$, $\lvert f\rvert = 2$
2	$s \to b \to t$	$\min\{2,3\} = 2$	$(2,2,0,2,2)$, $\lvert f\rvert = 4$
3	$s \to a \to b \to t$	$\min\{1,1,1\} = 1$	$(3,2,1,2,3)$, $\lvert f\rvert = 5$

Po třetím kole už reziduální síť neobsahuje cestu ze s do t: hrany (s,a) i (s,b) jsou nasycené, takže ze zdroje nevede *žádná* dopředná rezerva.

Ze zdroje přitom nemůže odtéct víc než

$$c(s,a) + c(s,b) = 3 + 2 = 5 = |f|, \qquad \checkmark$$

takže tok je maximální. (Jde o řez $S = \{s\}$; řezy zavedeme za chvíli.)

Odkud se úloha vzala

Ford a Fulkerson formulovali problém maximálního toku v RAND Corporation roku 1955. Zadání bylo konkrétní a dlouho utajené:

jakou kapacitu má sovětská železniční síť
zásobující satelitní státy ve východní Evropě?

Studie „Fundamentals of a Method for Evaluating Rail Net Capacities“ (Harris & Ross, 1955) obsahuje mapu sítě s ručně vyznačeným minimálním řezem, označeným jako „*the bottleneck*“.

Šlo tedy o to, kde síť přerušit. Táž věta odpovídá i na obrácenou otázku: kolik lze sítí nejvýše přepravit.

Zpráva byla odtajněna až v roce 1999.

Řezy jako horní meze

Definice 7 (Řez tokové sítě)

Rozdělení (S, T) vrcholů s $s \in S$, $t \in T$. Kapacita

$$c(S,T) = \sum_{\substack{(u,v)\in E \\ u\in S,\ v\in T}} c(u,v)$$

— počítáme *pouze* původní hrany ze strany zdroje na stranu stoku.

Lemma 8 (Čistý tok přes řez)

Pro každý přípustný tok a každý řez

$$|f| = \sum_{u\in S,\ v\in T} f(u,v) - \sum_{u\in T,\ v\in S} f(u,v).$$

Hodnota toku = tok dopředu přes řez − tok zpět přes řez.

Slabá dualita

Čistý tok přes řez.

Sečteme $\mathrm{out}(v) - \mathrm{in}(v)$ přes $v \in S$; podle zachování toku je součet roven $|f|$. Táž suma po hranách: hrany uvnitř S se odečtou, hrany $S \to T$ přispějí $+f$, hrany $T \to S$ $-f$, hrany uvnitř T nic.

Lemma 9

Pro každý přípustný tok f a každý řez (S, T) platí $|f| \leq c(S, T)$.

Důkaz.

Druhý součet je nezáporný a odečítá se, takže jeho vynecháním hodnotu nezmenšíme; v prvním součtu je $f(u, v) \leq c(u, v)$.

Každý řez tedy poskytuje certifikát horní meze. V naší malé síti má řez $S = \{s\}$ kapacitu $3 + 2 = 5$; uvedený tok má také hodnotu 5, a musí být proto optimální.

Věta max-flow/min-cut

Věta 10 (Ford, Fulkerson 1956)

V každé tokové síti s celočíselnými kapacitami se maximální hodnota toku rovná minimální kapacitě řezu. Fordův–Fulkersonův algoritmus po zastavení vrátí maximální tok a z jeho reziduální sítě lze přečíst minimální řez.

Důkaz.

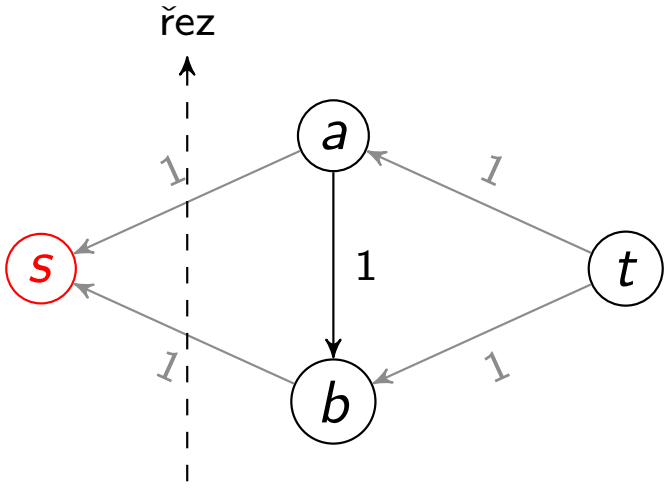
Nechť f je tok, pro který už neexistuje zvětšující cesta. Označme S množinu vrcholů *dosažitelných* ze s v reziduální síti, $T = V - S$; stok leží v T.

Každá hrana (u, v) s $u \in S, v \in T$ je nasycená — jinak by měla kladnou dopřednou rezervu a v by bylo v S. Každá hrana (u, v) s $u \in T, v \in S$ nese nulový tok — jinak by zpětná hrana $v \to u$ zpřístupnila u.

V lemmatu o čistém toku je proto první součet přesně $c(S, T)$ a druhý nulový, tedy $|f| = c(S, T)$.

Certifikát optimálnosti v praxi

Na diamantové síti se algoritmus zastavil s $|f| = 2$:



Ze zdroje nevede *žádná* reziduální hrana (obě hrany jsou nasycené, nabízejí jen zpětné rezervy *do* s). Prohledávání skončí ihned: $S = \{s\}$, $T = \{a, b, t\}$ a

$$c(S, T) = 1 + 1 = 2 = |f|.$$

Obě čísla lze nezávisle přepočítat bez sledování běhu algoritmu.

Zastavení a složitost

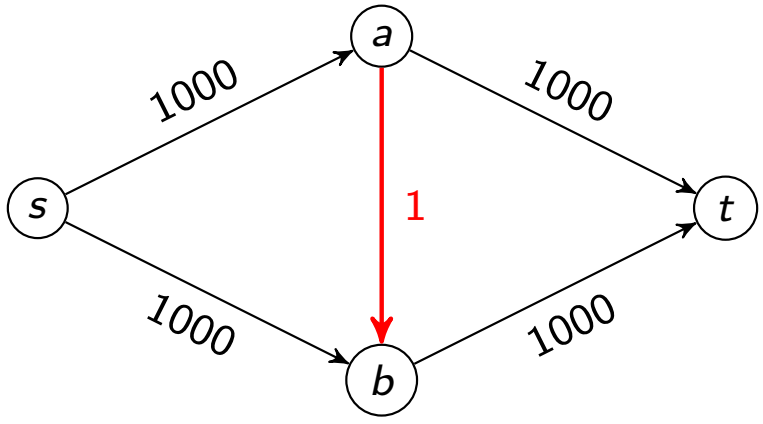
Kapacity jsou celá čísla, každá rezerva Δ je kladné celé číslo $\Rightarrow$ hodnota toku roste v každé iteraci alespoň o jedna. Je-li maximum F, proběhne nejvýše F iterací:

$$O\big((|V| + |E|)\, F\big).$$

Pseudopolynomiální mez — F může být veliké číslo zapsané málo bity.

Tím je doplněn předpoklad věty max-flow/min-cut: algoritmus se vždy zastaví. Následující příklad ukazuje, že mez je těsná.

Kdy je Ford–Fulkerson pomalý



Maximální tok je zjevně 2000 (dvě cesty po 1000).

Při nevhodné volbě cest ale algoritmus posílá tok po jedné jednotce:

$$s \to a \to b \to t, \quad s \to b \underset{\text{zpětná}}{\to} a \to t, \quad s \to a \to b \to t, \quad \ldots$$

Každá cesta má rezervu $\Delta = 1$ kvůli prostřední hraně — 2000 iterací na síti s pěti hranami.

Mez $O((|V| + |E|)F)$ je tedy těsná; stačí přepsat kapacitu 1000 na 10^9, aby algoritmus prakticky nedoběhl.

Proč právě BFS: Edmonds–Karp

V předchozím příkladu algoritmus *vybíral zbytečně dlouhé cesty*. Edmonds a Karp (1972): vybírej zvětšující cestu s nejmenším počtem hran, tj. hledej ji pomocí BFS.

- Vzdálenost $\mathrm{dist}_f(s, v)$ v reziduální síti se během běhu nikdy nezmenší.
- Kdykoli je hrana *nasycena* (a zmizí z reziduální sítě), musí se před svým dalším použitím vzdálenost jejího počátku zvětšit alespoň o 2.
- Každá hrana proto může být nasycena nejvýše $O(|V|)$-krát, celkem tedy $O(|V||E|)$ zvětšení.

Celkem tedy $O(|V|\,|E|^2)$, a to nezávisle na kapacitách.

Na diamantu s kapacitami 1000 najde BFS rovnou obě dvouhranové cesty a skončí po *dvou* iteracích.

Celočíselnost a diskrétní úlohy

Věta 11 (Celočíselný tok)

Má-li síť celočíselné kapacity, existuje maximální tok, který je na každé hraně celočíselný.

Důkaz.

Začínáme nulovým celočíselným tokem; reziduální kapacity i rezervy jsou celá čísla, takže zvětšení celočíselnost zachová.

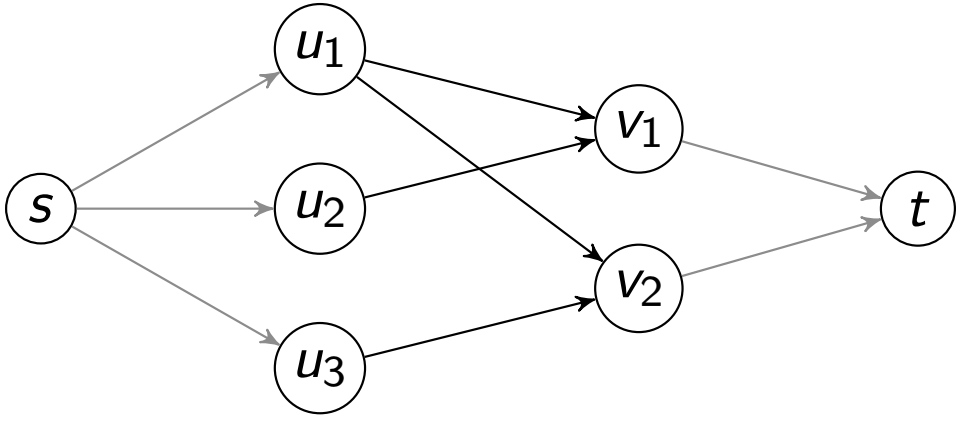
Lemma 12 (Rozklad toku na cesty)

Každý celočíselný tok f lze rozložit na $|f|$ jednotkových cest ze s do t a případně několik jednotkových kružnic tak, že tok na každé hraně je právě počtem cest a kružnic, které ji používají.

Náčrt. Ze zdroje jdeme po hranách s kladným tokem; zachování toku zaručuje, že uvíznout lze jedině ve stoku. Opakovaný vrchol dá kružnici, kterou odečteme; jinak dorazíme do t a odečteme jednotku podél cesty.

Bipartitní párování

$G = (L \cup R, E)$ bipartitní; párování je množina hran bez společného vrcholu. Chceme co nejvíc hran.



Přidáme $s \to u$ pro $u \in L$, orientujeme hrany z L do R, přidáme $v \to t$ pro $v \in R$; všechny kapacity jedna.

Jednotka toku cestou $s \to u \to v \to t$ pak odpovídá hraně párování uv.

Tok a párování — obě strany převodu

Věta 13

Velikost maximálního párování se rovná hodnotě maximálního toku v uvedené síti.

Důkaz.

$\Rightarrow$ Z párování velikosti k pošleme pro každou hranu uv jednotku cestou $s \to u \to v \to t$. Párování nepoužívá vrchol dvakrát, takže jednotkové kapacity vydrží.

$\Leftarrow$ Celočíselný tok rozložíme na jednotkové cesty; síť nemá kružnice a každá cesta má tvar $s \to u \to v \to t$. Hrany ze zdroje a do stoku mají kapacitu jedna, takže žádné u ani v neleží na dvou cestách — prostřední hrany tvoří párování velikosti $|f|$.

Co tu dělají zpětné hrany

Zvětšující cesta *střídá* nezvolené a zvolené hrany: nezvolenou přidá, zvolenou na zpětném kroku odebere. Párování vzroste o jedna, přestože jsme část starého rozhodnutí přeuspořádali.

Párování na naší malé instanci

$L = \{u_1, u_2, u_3\}$, $R = \{v_1, v_2\}$, hrany u_1v_1, u_1v_2, u_2v_1, u_3v_2.

Horní mez ihned: do stoku vedou jen *dvě* jednotkové hrany, takže $|f| \leq 2$.
(Je to řez $S = V \setminus \{t\}$.)

Dolní mez konstrukcí: cesty $s \to u_2 \to v_1 \to t$ a $s \to u_3 \to v_2 \to t$ dají tok hodnoty 2.

Maximální párování má tedy velikost 2, například $\{u_2v_1, u_3v_2\}$.

Párování na naší malé instanci

$L = \{u_1, u_2, u_3\}$, $R = \{v_1, v_2\}$, hrany u_1v_1, u_1v_2, u_2v_1, u_3v_2.

Horní mez ihned: do stoku vedou jen *dvě* jednotkové hrany, takže $|f| \leq 2$. (Je to řez $S = V \setminus \{t\}$.)

Dolní mez konstrukcí: cesty $s \to u_2 \to v_1 \to t$ a $s \to u_3 \to v_2 \to t$ dají tok hodnoty 2.

Maximální párování má tedy velikost 2, například $\{u_2v_1, u_3v_2\}$.

Kdyby algoritmus začal nešikovně

První cesta $s \to u_1 \to v_1 \to t$ obsadí v_1, jediného souseda u_2. Algoritmus pak může najít cestu $s \to u_2 \to v_1 \to u_1 \to v_2 \to t$, v níž krok $v_1 \to u_1$ je *zpětná* hrana: u_1 uvolní v_1 pro u_2 a přejde k v_2. Výsledek $\{u_2v_1, u_1v_2\}$ má opět velikost 2.

Běžné modelovací konstrukce

požadavek	**jak jej vyjádřit v síti**
více zdrojů / více stoků	přidej *superzdroj* s^* s hranami do všech zdrojů (kapacita ∞ nebo daná nabídka)
kapacita *vrcholu*	rozděl v na $v_{\text{in}} \to v_{\text{out}}$ s touto kapacitou
neorientovaná hrana	nahraď dvojicí protisměrných hran téže kapacity
hrana musí nést *alespoň* ℓ	tok s dolními mezemi — převádí se na obyčejný tok pomocnou konstrukcí
„každý dostane právě jednu“	jednotkové kapacity u zdroje i stoku
zakázaná kombinace	*nepřidávej* příslušnou hranu

Většinu práce při redukci na tok tedy představuje překlad omezení do kapacit; algoritmus zůstává stejný.

Důsledek: Hallova věta

Co když nás nezajímá velikost párování, ale zda lze spárovat všechny?

Věta 14 (Hall, 1935)

V bipartitním grafu $G = (L \cup R, E)$ existuje párování pokrývající celé L právě tehdy, když pro každou podmnožinu $S \subseteq L$ platí

$$|N(S)| \geq |S|,$$

kde $N(S)$ je množina sousedů vrcholů z S.

Náčrt.

Podmínka je zjevně nutná. Naopak, neexistuje-li úplné párování, je podle max-flow/min-cut minimální řez menší než $|L|$. Označme S vrcholy L na straně zdroje; řez obsahuje hrany $s \to u$ pro $u \in L \setminus S$ a pro každé $v \in N(S)$ hranu $v \to t$ nebo některou hranu $u \to v$ s $u \in S$. Odtud $|L| - |S| + |N(S)| < |L|$, tedy $|N(S)| < |S|$.

Existenční větu jsme tak odvodili z algoritmické: minimální řez dokládá, proč párování neexistuje.

Důsledek: Mengerova věta

Věta 15 (Menger, 1927)

Maximální počet hranově disjunktních cest ze s do t se rovná nejmenšímu počtu hran, jejichž odstranění oddělí s od t.

Důkaz.

Dejme každé hraně kapacitu 1. Maximální tok je pak počet disjunktních cest (rozklad toku), minimální řez je počet odstraněných hran. Věta max-flow/min-cut je ztotožní.

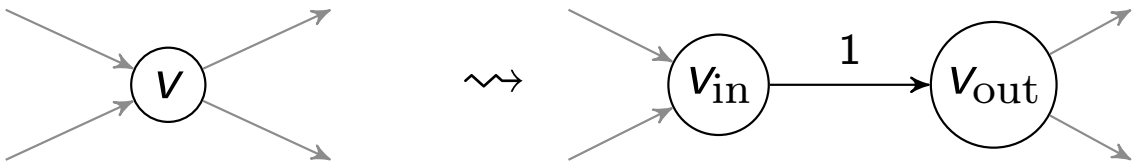
Hallova a Mengerova věta i Königova věta (v bipartitním grafu je největší párování stejně velké jako nejmenší vrcholové pokrytí) jsou tedy specializacemi jediné věty o toku a řezu. Historicky byly ovšem dokázány dřív než ona.

Menger 1927, Hall 1935, König 1931 — Ford a Fulkerson až 1956.

Disjunktní cesty

Hranově disjunktní. Každé původní hraně dáme kapacitu jedna. Celočíselný tok hodnoty k rozložíme na k jednotkových cest (kružnice odstraníme); kapacita jedna zaručí, že žádná hrana neleží ve dvou cestách. Obráceně k disjunktních cest určuje tok hodnoty k.

Vrcholově disjunktní. Rozdělíme vnitřní vrchol:



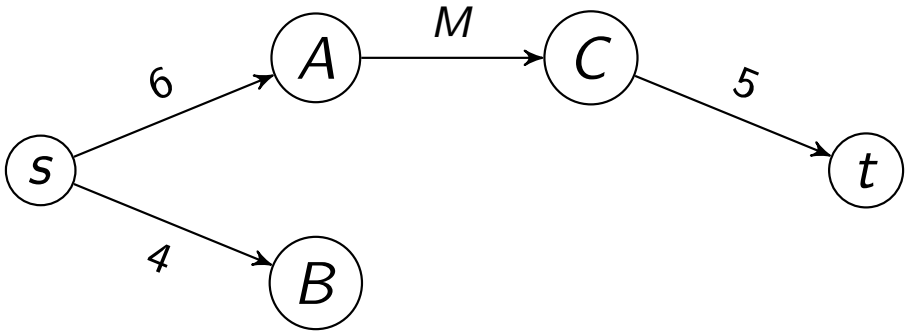
Omezení jako kapacita

Hrana kapacity jedna kóduje „použij nejvýše jednou“; rozdělení vrcholu převádí omezení *na vrcholu* na obyčejné omezení *hrany*.

Výběr projektů se závislostmi

Projekty s přínosy b_i (kladnými i zápornými). Závislost $i \Rightarrow j$: vybereme-li i, musíme vybrat i j. Hledáme *uzavřenou* množinu s největším součtem přínosů.

- $b_i > 0$: hrana $s \to i$ kapacity b_i;
- $b_i < 0$: hrana $i \to t$ kapacity $-b_i$;
- závislost $i \Rightarrow j$: hrana $i \to j$ kapacity M, kde M je větší než součet všech kladných přínosů.



Přínosy $6, 4, -5$; závislost $A \Rightarrow C$.

Proč minimální řez respektuje závislosti

Označme $B^+ = \sum_{b_i>0} b_i$.

Řez $S = \{s\}$ (žádný projekt) přeřezává jen hrany $s \to i$, takže má kapacitu nejvýše B^+ — minimální řez tedy má kapacitu $\leq B^+$.

Kdyby minimální řez přeřízl hranu závislosti, měl by kapacitu alespoň $M > B^+$. Spor.

Minimální řez proto nikdy nenechá zvolený projekt i v S a požadovaný j v T — vybraná množina je uzavřená.

Pro uzavřenou A (a $S = \{s\} \cup A$) překračují řez jen

- hrany $s \to i$ pro kladné $i \notin A$;
- hrany $i \to t$ pro záporné $i \in A$.

Minimalizace řezu = maximalizace přínosu

$$c(S,T) = \sum_{\substack{b_i>0 \\ i\notin A}} b_i + \sum_{\substack{b_i<0 \\ i\in A}} (-b_i)$$

a přínos vybrané množiny

$$\sum_{i\in A} b_i = B^+ - \sum_{\substack{b_i>0 \\ i\notin A}} b_i - \sum_{\substack{b_i<0 \\ i\in A}} (-b_i),$$

tedy

$$\boxed{\text{přínos} = B^+ - c(S,T)}$$

B^+ je pevná konstanta $\Rightarrow$ minimalizace řezu maximalizuje přínos.

Tato redukce tedy nepoužívá tok jako fyzický přesun; větu max-flow/min-cut používá čistě jako optimalizační nástroj.

Projekty A, B, C: výpočet

Přínosy $6, 4, -5$; závislost $A \Rightarrow C$. Součet kladných $B^+ = 6 + 4 = 10$.

výběr A	uzavřený?	$c(S, T)$	přínos $= B^+ - c(S, T)$
$\emptyset$	ano	$6 + 4 = 10$	0
$\{B\}$	ano	6	4
$\{C\}$	ano	$6 + 4 + 5 = 15$	-5
$\{A, C\}$	ano	$4 + 5 = 9$	1
$\{B, C\}$	ano	$6 + 5 = 11$	-1
$\{A, B, C\}$	ano	5	5
$\{A\}$, $\{A, B\}$	ne (A chce C)	—	—

Minimální řez má kapacitu 5 (jediná hrana $C \to t$), takže optimální výběr je $\{A, B, C\}$ s přínosem $6 + 4 - 5 = 5$. ✓

Poslední řádek: výběry $\{A\}$ a $\{A, B\}$ by přeřízly hranu kapacity M, a jejich „řezy“ by tedy stály víc než $B^+ = 10$. Právě proto minimální řez nikdy nevrátí neuzavřenou množinu.

Další aplikace minimálního řezu

- **Segmentace obrazu.** Pixely jsou vrcholy, zdroj = popředí, stok = pozadí. Hrany k s/t nesou, jak moc pixel „vypadá“ jako popředí; hrany mezi sousedy penalizují nesouhlasné označení. Minimální řez = optimální segmentace.
- **Vyřazení v lize** (*baseball elimination*). Může tým ještě vyhrát ligu? Sestaví se síť zbývajících zápasů; tým je vyřazen právě tehdy, když maximální tok nedokáže „rozdělit“ všechna zbývající vítězství.
- **Otevírání dolů (*open-pit mining*).** Blok rudy má hodnotu, ale nelze jej vytěžit bez bloků nad ním; jde přesně o výběr projektů se závislostmi.
- **Plánování směn, přiřazování úkolů, rozvrhy** — vesměs párování s kapacitami.

Rodina algoritmů pro maximální tok

algoritmus	rok	čas
Ford–Fulkerson (libovolná cesta)	1956	$O((\|V\| + \|E\|)\, F)$ — pseudopolynomiální
Edmonds–Karp (BFS)	1972	$O(\|V\|\,\|E\|^2)$
Dinic (vrstvená síť, blokující tok)	1970	$O(\|V\|^2\|E\|)$
na jednotkových kapacitách		$O(\|E\|\sqrt{\|E\|})$
Push–relabel	1988	$O(\|V\|^3)$
Chen a kol. (téměř lineární)	2022	$\|E\|^{1+o(1)}$

- **Dinic** je v praxi nejběžnější volbou; na bipartitním párování s jednotkovými kapacitami se redukuje na Hopcroft–Karpův algoritmus $O(|E|\sqrt{|V|})$.
- Poslední řádek je teoretický výsledek z roku 2022; i u úlohy studované přes šedesát let se tedy stále objevují nové výsledky.

Pro tento kurz stačí umět Ford–Fulkersona a vědět, proč BFS pomáhá.

Jak navrhovat redukci na tok

Kontrolní otázky

1. Co má představovat jedna *jednotka toku*?
2. Které omezení vyjádří kapacita jedna a které větší kapacita?
3. Jak převést libovolné řešení původní úlohy *na* tok?
4. Jak převést celočíselný tok *zpět* na původní řešení?
5. Zachovává převod hodnotu cíle v obou směrech?
6. Lze z minimálního řezu přečíst užitečný certifikát nebo přímo řešení?

Nestačí síť pouze nakreslit. Důkaz redukce má *dvě strany*: každé řešení původní úlohy vytváří tok stejné hodnoty a každý vhodný tok vytváří původní řešení. Teprve obě nerovnosti dokazují rovnost optim.

Shrnutí

- Reziduální síť popisuje nejen volnou kapacitu, ale i možnost *rušit* starý tok.
- Zvětšující cesta zachovává přípustnost a zvyšuje hodnotu toku.
- Každý řez je horní mez; shoda hodnoty toku a kapacity řezu certifikuje optimálnost.
- Celočíselné kapacity zaručují existenci celočíselného maximálního toku — proto lze tok použít pro diskrétní úlohy.
- Kapacity a pomocné vrcholy modelují omezení v párování, disjunktních cestách i projektových závislostech.

Literatura



L. R. Ford Jr., D. R. Fulkerson. Maximal Flow Through a Network. *Canadian J. of Mathematics* 8 (1956), 399–404.



J. Edmonds, R. M. Karp. Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems. *JACM* 19(2) (1972), 248–264.



J. Kleinberg, É. Tardos. *Algorithm Design*. Addison–Wesley (2006), kap. 7.



T. H. Cormen a kol. *Introduction to Algorithms*. MIT Press (2001).