# 4. Goldbergův algoritmus *(zapsali R. Tupec, J. Volec, J. Záloha)*

Představíme si nový algoritmus pro hledání maximálního toku v síti, který se ukáže stejně dobrý jako *Dinicův algoritmus* ($\mathcal{O}(MN^2)$) a po několika vylepšeních bude i lepší.

Tento algoritmus narozdíl od Dinicova algoritmu začíná s přebytky v sousedních vrcholech zdroje a snaží se jich zbavit pomocí převádění. Pokud bychom toto převádění dělali „tupým způsobem“, mohl by se algoritmus zacyklit. Proto pro každý vrchol budeme definovat výšku, a jak uvidíme, s její pomocí se vyhneme zacyklení.

**Definice:** Funkce $f : E \to \mathbb{R}_0^+$ je *vlna* v síti $(V, E, z, s, c)$ tehdy, když $\forall uv \in E$ : $f(uv) \leq c(uv)$, kde $c(uv)$ je kapacita hrany $uv$, a $\forall v \neq z, s : f^\Delta(v) \geq 0$. Funkcí $f^\Delta(v)$ pro libovolný vrchol $v$ rozumíme *přebytek* v tomto vrcholu, což je součet všeho, co do vrcholu $v$ přiteče, minus součet všeho, co z $v$ odteče. To můžeme zapsat jako:

$$f^\Delta(v) := \sum_{uv \in E} f(uv) - \sum_{vu \in E} f(vu).$$

Každý tok je vlna, kde $\forall v \neq z, s : f^\Delta(v) = 0$.

Algoritmus používá sít rezerv, kterou jsme nadefinovali již v předchozí kapitole věnované Dinicovi.

Dále budeme provádět následující dvě operace na vrcholech sítě. K tomu budeme potřebovat přiřadit všem vrcholům výšku pomocí funkce $h : V \to \mathbb{N}$.

**Operace:** Pro hranu $uv \in E$ definujme *převedení přebytku*:

Pokud platí, že:

1. ve vrcholu $u$ je nenulový přebytek, tj. $f^\Delta(u) > 0$,
2. vrchol $u$ je výš než vrchol $v$, tj. $h(u) > h(v)$, a
3. hrana $uv$ má nenulovou rezervu, tj. $r(uv) > 0$,

převedeme tok o velikosti $\delta := \min(f^\Delta(u), r(uv))$ z $u$ do $v$ tímto způsobem:

1. $f^\Delta(u) \leftarrow f^\Delta(u) - \delta$ a $f^\Delta(v) \leftarrow f^\Delta(v) + \delta$,
2. $r(uv) \leftarrow r(uv) - \delta$ a $r(vu) \leftarrow r(vu) + \delta$.

**Definice:** Řekneme, že převedení je *nasycené*, pokud je po převodu rezerva na hraně $uv$ nulová, tedy $r(uv) = 0$. Naopak převedení je *nenasycené*, pokud po převodu $f^\Delta(u) = 0$. Pokud $r(uv) = 0$ a $f^\Delta(u) = 0$, budeme převedení považovat za *nasycené*.

**Operace:** Pro vrchol $u \in V$ definujme *zvednutí vrcholu*: Pokud během výpočtu narazíme ve vrcholu $u$ na přebytek, který nelze nikam převést, zvětšíme výšku vrcholu $u$ o jedničku, tj. $h(u) \leftarrow h(u) + 1$.

**Algoritmus (hledání maximálního toku v síti, Goldberg)**

1. $\forall v \in V : h(v) \leftarrow 0$ (všem vrcholům nastavíme počáteční výšku nula).
2. $h(z) \leftarrow N$ (zdroj zvedneme do výšky $N$).
3. $\forall e \in E : f(e) \leftarrow 0$ (po hranách na počátku nenecháme protékat nic).

4. $\forall zu \in E : f(zu) \leftarrow c(zu)$ (ze zdroje pustíme maximální možnou vlnu).
5. Dokud $\exists u \in V \setminus \{z, s\}, f^{\Delta}(u) > 0$:
6. Pokud $\exists uv \in E, r(uv) > 0$ a $h(u) > h(v)$: převedeme přebytek po hraně $uv$.
7. V opačném případě zvedneme $u$.
8. Vrátíme tok $f$ jako výsledek.

Nyní bude následovat několik lemmat a invariantů, jimiž dokážeme správnost a časovou složitost výše popsaného algoritmu.

**Invariant A:** Funkce $f : E \to \mathbb{R}$ je v každém kroku algoritmu vlna, $h(v)$ nikdy neklesá, $h(z) = N$ a $h(s) = 0$.

*Důkaz:* Pro první část invariantu si stačí rozmyslet, že v žádném kroku algoritmu nepřekročíme kapacity hran a nevytvoříme záporný přebytek. Pro $v \in V \setminus \{z, s\}$ skutečně výšku pouze zvyšujeme a z podmínky v pátém kroku algoritmu vyplývá, že nás přebytky v $z$ a $s$ v podstatě nezajímají, tudíž se $h(z)$ a $h(s)$ nemění. ♡

**Invariant S (o Spádu):** Neexistuje hrana se spádem větším než jedna a nenulovou rezervou, neboli $\forall uv \in E, r(uv) > 0 : h(u) \leq h(v) + 1$.

*Důkaz:* Podívejme se, kdy by mohla vzniknout nenasycená hrana se spádem větším než jedna. Během inicializace k tomu evidentně nedojde, protože všechny hrany jsou nenasycené nebo mají kapacitu nula, proto je můžeme vypustit. Během práce algoritmu k tomu však také nedojde, jak uvidíme z rozebrání následujících případů. Pokud již existuje vrchol $v$ s kladným přebytkem, dále existuje nenasycená hrana $vu$ a $h(v) = h(u) + 1$, vrchol $v$ algoritmus nezvedne, ale přebytek pošle po této hraně. Uvažme tedy ještě druhý případ, kdy existuje nasycená hrana $uv$ se spádem větším než jedna a tuto hranu se pokusíme odsytit. Jenže pokud bychom chtěli něco poslat v protisměru, snažili bychom se o přelití proti směru funkce $h$. ♡

**Lemma K (o Korektnosti):** Když se algoritmus zastaví, je $f$ maximální tok.

*Důkaz:* Nejprve ukážeme, že $f$ je tok, a pak jeho maximalitu. Vyjděme z toho, že $f$ je vlna a algoritmus se může zastavit, jen pokud nastanou oba následující případy současně:

- Ve vrcholech grafu nejsou žádné přebytky (mimo $z$ a $s$), protože jinak by se algoritmus nezastavil a pokračoval dále ve výpočtu. Tudíž $f$ je tok.
- Neexistuje nenasycená cesta ze zdroje do stoku, čímž z *Ford-Fulkersonovy věty* okamžitě vyplývá, že $f$ je tok maximální. A jak tuto neexistenci nahlédneme? Pro spor předpokládejme, že nějaká nenasycená cesta $P$ ze $z$ do $s$ existuje. Tato cesta může mít maximálně $N - 1$ hran. O nich víme, že všechny mají kladnou rezervu, a dále víme, že po celou dobu výpočtu je výška zdroje $N$ a výška stoku 0. Takže celkový spád cesty $P$ je $N$, což ale znamená, že na cestě $P$ existuje hrana s kladnou rezervou, která má spád alespoň 2. To je však v rozporu s invariantem S. ♡

**Invariant C (Cesta domů, do zdroje):** Je-li $v \in V \setminus \{z, s\}$ a $f^{\Delta}(v) > 0$, pak existuje nenasycená cesta z $v$ do $z$.

*Důkaz:* Mějme nějaký vrchol $v \in V$ takový, že $f^\Delta(v) > 0$. Potom definujme množinu $A := \{u \in V : \exists$ nenasycená cesta z $v$ do $u\}$. Mějme vrcholy $a \in A$ a $b \in V \setminus A$ takové, že $ba \in E$. O nich víme, že $f(ba) = 0$, protože pokud by tomu tak nebylo, muselo by platit $r(ab) > 0$, a tudíž by $b$ patřilo do množiny $A$.

Sečtěme přebytky ve všech vrcholech množiny $A$. Protože přebytek každého vrcholu se spočítá jako součet toků do něj vstupujících minus součet toků z něj vystupujících a všechny hrany, jejichž oba vrcholy leží v $A$, se jednou přičtou a jednou odečtou, platí:

$$\sum_{u\in A} f^\Delta(u) = \sum_{\substack{ab\in E\cap\mathbb{A} \\ \mathbb{A}=\bar{A}\times A}} f(a,b) - \sum_{\substack{ba\in E\cap\mathbb{A} \\ \mathbb{A}=A\times\bar{A}}} f(b,a).$$

My však víme, že do $A$ nic neteče, a proto $\sum_{v\in A} f^\Delta(v) \le 0$. Zároveň však v $A$ je vrchol s kladným přebytkem, totiž $v$, proto v $A$ musí být také vrchol se záporným přebytkem a jediný takový je $z$. ♡

**Invariant V (Výška):** $\forall v \in V$ platí $h(v) \le 2N$.

*Důkaz:* Víme, že počet hran na cestě ze $z$ do $\forall v \in V$ je maximálně $N - 1$. Pokud by existoval vrchol $v$ s výškou $h(v) > 2N$, museli jsme tento vrchol zvednout alespoň $(2N + 1)$-krát. Snadno si uvědomíme, že $z$ nikdy nezvedáme, a tudíž by na cestě ze $z$ do $v$ musela být hrana se spádem větším než jedna, což je spor s invariantem S. ♡

**Lemma Z (počet Zvednutí):** Počet všech zvednutí je maximálně $2N^2$.

*Důkaz:* Stačí si uvědomit, že každý vrchol můžeme zvednout maximálně $2N$-krát a vrcholů je $N$. ♡

**Lemma S (naSycená převedení):** Počet všech nasycených převedení je nejvýš $NM$.

*Důkaz:* Mějme hranu $uv \in E$, kterou jsme právě nasytili. Tedy platí $h(v) < h(u)$ a zároveň $r(uv) = 0$. Aby se rezerva této hrany změnila, musel by ji někdo odsytit. Pro odsycení hrany se musí otočit nerovnost mezi výškami koncových vrcholů. Tedy $h(v) > h(u)$. Proto, abychom tuto hranu opět nasytili, musíme opět změnit nerovnost výšek na $h(v) < h(u)$. Mezi dvěma nasyceními hranami $uv$ proto proběhla minimálně dvě zvednutí vrcholu $u$. Algoritmus nikdy výšku vrcholu nesnižuje, a tedy počet všech nasycených převedení je skutečně $NM$. ♡

**Lemma N (Nenasycená převedení):** Počet všech nenasycených převedení je $\mathcal{O}(N^2M)$.

*Důkaz:* Důkaz provedeme pomocí potenciálu – nadefinujme si následující funkci jako potenciál:

$$\psi := \sum_{\substack{v: f^\Delta(v)>0 \\ v\neq z,s}} h(v).$$

Nyní se podívejme, jak se náš potenciál během algoritmu vyvíjí a jaké má vlastnosti:

- Během celého algoritmu je $\psi \ge 0$, neboť je součtem nezáporných členů.
- Na počátku je $\psi = 0$.

- Zvednutí vrcholu zvýší $\psi$ o jedničku. Již víme, že za celý průběh algoritmu je všech zvednutí maximálně $2N^2$, proto zvedáním vrcholů zvýšíme potenciál dohromady nejvýše o $2N^2$.
- Nasycené převedení zvýší $\psi$ nejvýše o $2N$, protože buď po převodu hranou $uv$ v $u$ zůstal nějaký přebytek, takže se mohl potenciál zvýšit až o $2N$, nebo je přebytek v $u$ po převodu nulový a potenciál se dokonce o jedna snížil. Za celý průběh tak dojde k maximálně $NM$ takovýmto převedením, díky nimž se potenciál zvýší maximálně o $2N^2M$.
- Konečně když převádíme po hraně $uv$ nenasyceně, tak od potenciálu určitě odečteme výšku vrcholu $u$ a možná přičteme výšku vrcholu $v$. Jenže $h(v) = h(u)-1$, a proto nenasycené převedení potenciál vždy sníží alespoň o jedna.

Z tohoto rozboru chování potenciálu $\psi$ v průběhu algoritmu získáváme, že počet všech nenasycených převedení může být nejvýše $2N^2 + 2N^2M$, což je $\mathcal{O}(N^2M)$. ♡

**Implementace:** Budeme si pamatovat seznam $P$ všech vrcholů $v \neq z, s$ takových, že $f^\Delta(v) > 0$. Když měníme přebytek nějakého vrcholu, můžeme náš seznam v konstantním čase aktualizovat (např. tak, že si každý vrchol pamatuje pozici, na které v seznamu je). A v konstantním čase také umíme odpovědět, zda existuje nějaký vrchol s přebytkem. Dále si $\forall u \in V$ budeme pamatovat $L(u) :=$ seznam $uv \in E$ takových, že $r(uv) > 0$ a $h(v) < h(u)$. Díky tomu můžeme přistupovat k patřičným sousedům $u$ v čase $\mathcal{O}(1)$, stejně jako provádět operace přidání do $L(u)$, resp. smazání v něm. Každé převedení po hraně $uv$ nás stojí konstantní čas na aktualizaci rezerv hran $uv$ a $vu$, stejně tak i na aktualizaci přebytků ve vrcholech $u$ a $v$. V případě, že se jedná o nasycené převedení, musíme ještě odstranit hranu $uv$ z $L(u)$, což také stihneme v čase $\mathcal{O}(1)$. A konečně zvedání vrcholu $v$ nám zabere čas $\mathcal{O}(N)$, protože musíme obejít všechny hrany $uv$, kterých je $\mathcal{O}(N)$, porovnat výšky a případně odebrat $uv$ z seznamu $L(u)$ resp. přidat do $L(v)$. Abychom pro odebrání hrany $uv$ ze seznamu $L(u)$ nemuseli procházet celý seznam, budeme si $\forall v \in V$ pamatovat ještě $L^{-1}(v) :=$ seznam ukazatelů na hrany $uv$ v seznamech $L(u)$.

**Věta:** Goldbergův algoritmus najde maximální tok v čase $\mathcal{O}(N^2M)$.

*Důkaz:* Z lemmatu Z vyplývá, že celkový počet zvednutí je maximálně $2N^2$, přičemž každé zvednutí jsme schopni provést v čase $\mathcal{O}(N)$. Takže dohromady pro zvedání spotřebujeme čas $\mathcal{O}(N^3)$, což je pro souvislé sítě určitě $\mathcal{O}(N^2M)$. Z lemmatu S pro změnu vyplývá, že nasycená převedení nás stojí $\mathcal{O}(NM)$, a na závěr z lemmatu N dostáváme časovou složitost $\mathcal{O}(N^2M)$ pro převedení nenasycená. Proto celková složitost algoritmu je $\mathcal{O}(N^2M)$. ♡

Dokázali jsme, že algoritmus má časovou složitost $\mathcal{O}(N^2M)$ pro libovolnou posloupnost zvedání a převádění. Nabízí se otázka, zda není možné vhodným výběrem těchto operací výpočet zrychlit. Ukážeme, že pokud v 5. kroku algoritmu budeme vždy brát vrchol $u$ takový, že $h(u)$ je maximální, počet nenasycených převedení se sníží.

**Lemma N':** Počet nenasycených převedení v upravené verzi Goldbergova algoritmu



je $\mathcal{O}(N^2\sqrt{M})$, což je maximálně $\mathcal{O}(N^3)$. Díky tomu je i složitost celého algoritmu $\mathcal{O}(N^3)$.

*Důkaz:* Viz příští přednášku.