PB165 – Grafy a sítě

Toky v síti

Obsah přednášky

Řez v grafu

Neformálně:

- „Rozříznutí" grafu napříč hranami (nikoliv skrz vrcholy) na dvě poloviny.
- Rozdělení vrcholů na dvě části.

Definice

Řezem v grafu $G = (V, E)$ nazýváme rozklad množiny V na 2 podmnožiny $P, \overline{P}$.

Jelikož se jedná o rozklad, platí:

$$P \cap \overline{P} = \emptyset, P \cup \overline{P} = V$$

Řezy v grafu

V každém grafu existuje $2^{|V|-1}$ řezů.

Obrázek: Příklady řezů v grafu jsou vyznačeny čárkovaně.

Hrany křižující řezy

Definice

Nechť řez C dělí vrcholy na množiny $P, \overline{P}$. O hranách (u, v), jejichž jeden vrchol leží v P a druhý nikoliv, říkáme že křižují řez C.

Obrázek: Hrany křižující řez C jsou vyznačeny modře.

Váha řezu

Definice

Vahou řezu v hranově neohodnoceném grafu označujeme počet hran, které tento řez křižují.
V hranově ohodnoceném grafu se vahou rozumí součet ohodnocení všech hran křižujících tento řez.

Obrázek:

Minimální a maximální řez

Definice

Minimálním rozumíme takový řez v grafu, jehož váha je minimální. Maximální řez je naopak ten s maximální vahou.

Minimální řez v grafu může být nalezen v čase polynomiálním vůči velikosti grafu. Naopak, problém maximálního řezu je NP-úplný.
Obrazek

Síť

Definice

Sítí nazýváme orientovaný, hranově ohodnocený graf $G = (V, E)$*. Hodnotící funkce* $c : E \to \mathbb{R}$ *definuje tzv. kapacitu hrany. V síti jsou definovány dva význačné vrcholy – zdroj a stok (výlevka, sink).*

Sítě jsou vhodnými reprezentacemi reálných sítí – např. železniční, silniční, vodovodní – nebo počítačové. Kapacita hrany potom reprezentuje např. propustnost linky. Zdroj a výlevka mohou odpovídat přijímajícím a odesílajícím uzlům.

Toky v síti

Na hraně může být definován také *tok*, reprezentující např. množství dat aktuálně přenášených konkrétní linkou počítačové sítě.

Definice

Tok je hodnotící funkce tvaru $f : E \rightarrow \mathbb{R}$.

Tok musí splňovat následující vlastnosti:

- Pro všechny hrany e platí $f(e) < c(e)$.
- Pro všechny vrcholy platí (vyjma zdroje a stoku) tzv. Kirchhoffův zákon – součet všech toků vstupujících do vrcholu je roven součtu všech toků z vrcholu vystupujících.
- Ze zdroje vychází větší součet toků než do něj vchází a stejný rozdíl je mezi součtem toků do stoku vcházejících a z něj vycházejících.

Hodnota $c(v) - f(v)$ definuje *reziduální* kapacitu hrany. Reziduální kapacity hran tvoří reziduální síť.

Toky v síti – příklad

Obrazek

Velikost toku

Velikost toku značíme $F(f)$. Velikost toku od zdroje ke stoku definujeme jako množství toku, které vzniká ve zdroji s.

$$F(f) = \sum_{e \in E^+(s)} f(e) - \sum_{e \in E^-(s)} f(e)$$

E^+, E^- označují v tomto případě (a dále) součet toků vstupujících do vrcholu, resp. vystupujících z něj.
Obrazek

Velikost toku přes řez

Nechť řez C dělí vrcholy grafu na množiny $P, \overline{P}$. Označíme jej C_P. Dále nechť zdroj toku náleží do množiny P a stok nikoliv. Potom má smysl definovat velikost F_P toku přes řez C_P jako rozdíl mezi velikostí toku na hranách vedoucích z množiny P a velikostí toku na hranách vedoucích do této množiny. Říkáme, že řez C_P *odděluje* zdroj a stok.

$$F_P(f) = \sum_{e \in W^-(P)} f(e) - \sum_{e \in W^+(P)} f(e)$$

W^-, W^+ značí hrany vycházející z množiny W, resp. do ní vstupující.

Shodnost toků přes řezy

Věta

Nechť C_P je libovolný řez, který odděluje zdroj a stok. Potom pro velikost F_P toku přes C_P platí

$$F_P(f) = F(f)$$

Přes všechny řezy oddělující zdroj a stok tedy protéká stejný tok.
Obrazek

Shodnost toků přes řezy – důkaz

Důkaz.

Důkaz povedeme indukcí:
Základ indukce: Položme $P = \{s\}$, kde s je zdroj toku. Tvrzení platí z definice.
Indukční krok: Do množiny P přidáme libovolný vrchol grafu, různý od stoku. Jelikož pro tento vrchol musí platit Kirchhoffův zákon, nezmění se nikterak rozdíl mezi velikostmi toků z P vytékajících a do P vtékajících. Platnost tvrzení se tedy nezmění. Jelikož postupným přidáváním vrcholů lze získat libovolný řez, který odděluje zdroj od stoku, platí tvrzení pro všechny řezy zdroj od stoku oddělující. $\square$

Kapacita řezu

Kapacita řezu oddělujícího zdroj a stok specifikuje, jaký maximální tok může tímto řezem protéct.

$$C(C_P) = \sum_{e \in W^-(P)} f(e)$$

Obrazek

Kapacita řezu má význam pro nalezení tzv. maximálního toku v grafu.

Maximální tok v grafu

Problém maximálního toku je hledání největšího toku v grafu od zdroje ke stoku. Následující podmínky jsou ekvivalentní:

1. Tok f je maximální.
2. $|f|$ je kapacita některého řezu oddělujícího zdroj od stoku.
3. V reziduální síti neexistuje cesta ze zdroje do stoku.

Algoritmy pro nalezení maximálního toku vycházejí z těctho ekvivalencí – hledají řez s minimální kapacitou nebo přidávají cesty mezi zdrojem a stokem, dokud nějaké v reziduální síti existují.

Algoritmus – brutální síla

- Nejjednodušší algoritmus.
- Generuje postupně všechny podmnožiny vrcholů, pro každý provede následující kroky:
  - Najde mezi hranami všechny, které křižují řez definovaný touto množinou vrcholů.
  - Sečte kapacity hran křižujících tento řez, směřujících od zdroje ke stoku.
- Výsledkem je řez s minimální vypočtenou kapacitou.

Jelikož všech řezů oddělujících zdroj od stoku je $2^{|V|-2}$, pro každý je potřeba zkontrolovat všech $|E|$ hran, celková časová složitost algoritmu hledání maximálního toku brutální silou je $\mathcal{O}(2^{|V|-2}|E|)$

Ford-Fulkersonův algoritmus

- Využívá ekvivalence mezi maximalitou toku a neexistencí cesty ze zdroje do stoku.
- Hledá *zlepšující* cesty mezi zdrojem a stokem, dokud nějaká taková existuje.
- Pro hledání cest používá i zpětných hran.
- Z hran na cestě se vybere ta, jejíž reziduální kapacita je minimální.
    - V případě zpětné hrany (projité proti směru její orientace) se namísto hodnoty $c(e) - f(e)$ bere tok, který hranou protéká ve směru její orientace – tedy $f(e)$. Toky se takto mohou vzájemně anulovat.
- O tuto minimální kapacitu se zvýší tok po všech dopředných hranách na nalezené cestě.
    - Naopak, na hranách zpětných se hodnota toku sníží o stejnou hodnotu.

Ford-Fulkersonův algoritmus

```
Pro všechny hrany (u,v)
| f(u,v) = 0
Dokud existuje zlepšující cesta p:
| Vyber minimální kapacitu c(p) hrany na této cestě.
| Pro všechny hrany na cestě p:
| | f(u,v) = f(u,v) + c(p)
| | f(v,u) = f(v,u) - c(p)
```

Algoritmus neříká, jakým způsobem se má cesta ze zdroje do stoku hledat. V praxi se používá obvykle průchod do hloubky, nebo průchod do šířky, čímž se z algoritmu stává Edmonds-Karpův (viz dále).

Ford-Fulkersonův algoritmus – složitost

- V obecném případě není možné dokázat, že běh algoritmu skončí.
- V některých případech nemusí hodnota nalezeného toku ani konvergovat k maximu.

V reálných aplikacích jsou kapacity hran obvykle reprezentovány celými čísly. to zaručuje ukončení běhu algoritmu:

- Maximální tok ma v takovém případě také celočíselnou hodnotu.
- Časová složitost nalezení zlepšující cesty je $\mathcal{O}(|E|)$
- S každou nalezenou zlepšující cestou se hodnota nalezeného toku zvýší minimálně o 1.
- Maximálně může tedy proběhnout nejvýše $F(f)$ iterací.

Ford-Fulkersonův algoritmus – příklad

Obrazek

Edmonds-Karpův algoritmus

- Specializace Ford-Fulkersonova algoritmu.
- Pro hledání zlepšujicích cest je použit průchod do šířky.
- Pro potřeby průchodu do šířky jsou délky hran považovány za jednotkové.
- Průchod do šířky zajistí, že každá nalezená zlepšující cesta je nejméně tak dlouhá, jako předchozí nalezená.
- Maximální možná délka zlepšující cesty je $|V|$.
- Složitost algoritmu tak činí $\mathcal{O}(VE^2)$.

Edmonds-Karpův algoritmus – příklad

Obrazek.

Omezení toku vrcholem

Reálné aplikace mohou klást i omezení na velikost toku procházejícího vrcholem – např. sběrné místo kanalizací, aktivní prvek v síti, rychlost zpracování dat na příjemci. Pokud jsou toky nezáporné, lze použít následující transformaci grafu:

Obrazek

Vrchol v s omezenou kapacitou nahradíme vrcholy v_1, v_2, jejichž kapacita nebude omezena. Hrany směřující do v přesměrujeme do v_1, hrany z v vycházející budou vycházet z v_2. Vrcholy v_1, v_2 spojíme hranou, jejíž kapacita bude rovna původní kapacitě vrcholu v. Na takový graf je poté možno použít standardní algoritmy pro hledání maximálního toku.

Několik zdrojů a stoků

Obdobně lze standardní algoritmy použít i v případě, kdy zadání obsahuje více než 1 zdroj nebo stok:
Obrazek
K síti přidáme fiktivní zdroj a stok. Z nově přidaného zdroje povedou hrany (s „neomezenou“ kapacitou) do všech zdrojů, obdobně přidáme hrany ze všech stoků do nově přidaného.

Nejlevnější toky

Ke každé hraně je krom její kapacity definována i cena $a(e)$ jednotkového toku. Cena toku hranou e je potom rovna $a(e)f(e)$. Celková cena toku sítí je potom definována jako

$$\sum_{e \in E} a(e)f(e).$$

Úkolem je potom najít maximální tok sítí takový, že jeho cena bude zároveň minimální.