Kombinatorická optimalizace
cvičení č.6
Aplikace toků v sítích

Zdeněk Bäumelt (baumezde@fel.cvut.cz)
Přemysl Šůcha (suchap@fel.cvut.cz)

28. března 2012

1 Toky v sítích

Celá řada praktických problémů týkajících se optimalizace se dá řešit pomocí toku v sítích. Graf, kterým je síť reprezentována, si můžeme představit jako soustavu potrubí, ve kterém proudí nějaká kapalina. Tok v každé hraně (rouře) je ustálený a beze ztrát.

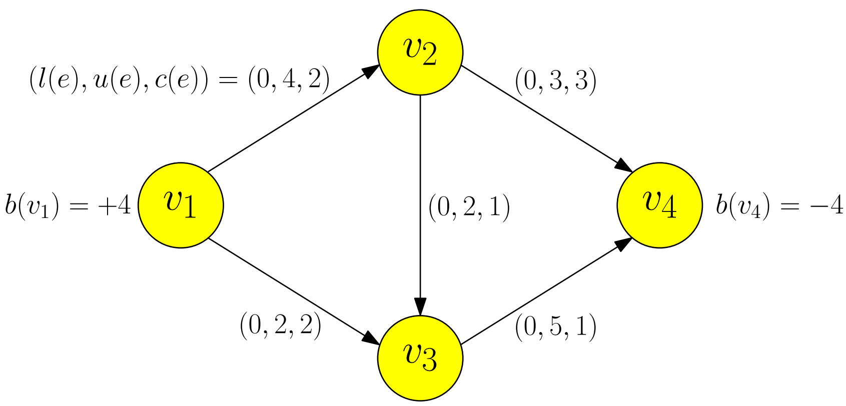
Definice 1.1 ***Tok****. Mějme orientovaný graf G.* ***Tokem v síti*** *nazýváme takové ohodnocení hran reálnými čísly $f : E(G) \to \mathbb{R}$, které pro každý vrchol v splňuje* ***Kirchhoffův zákon*** *[1]*

$$\sum_{e \in E^+(v)} f(e) = \sum_{e \in E^-(v)} f(e); \tag{1}$$

$E^+(v)$ je množina hran s počátečním vrcholem v a $E^-(v)$ je množina hran s koncovým vrcholem v.

Nejčastější úlohou týkající se toků v síti je *úloha hledání nejlevnějšího toku* [2]. Ta je formulována takto: Je dána transportní síť (G, b, l, u, c), kde G je orientovaný graf, b je vektor vrcholů jako zdrojů a spotřebičů, matice l a u jsou dolní a horní meze hran a matice c reprezentuje ceny hrany. Cílem je nalézt v dané síti nejlevnější přípustný tok, tak že pro každý vrchol $v \in V(G)$ platí

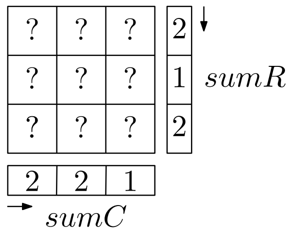
$$\sum_{e \in E^+(v)} f(e) = \sum_{e \in E^-(v)} f(e) + b(v). \tag{2}$$



Obrázek 1: Příklad sítě.

2 Rekonstrukce binárních obrázků pomocí toků v sítích

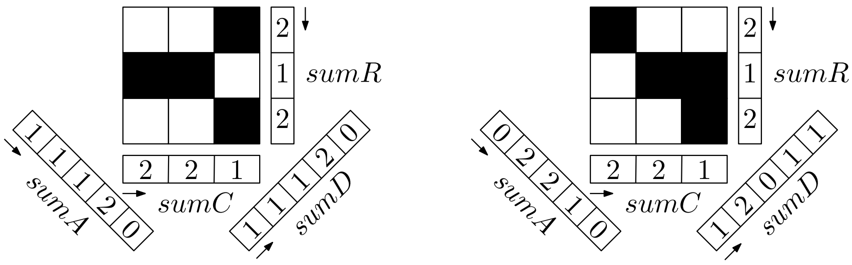
Cílem úlohy je rekonstruovat čtvercový binární obrázek z projekčních dat pomocí toků v sítích [3, 4], tj. určit hodnotu všech pixelů (0 = černý, 1 = bílý). Této metody je využíváno například v lékařství. Metodu popíšeme na jednoduchém binárním obrázku o rozměrech 3 x 3 pixely (viz obr. 2). K dispozici jsou pouze data odpovídající různým projekcím – vektor součtu pixelů po řádcích $sumR$ (projekce $\mathcal{R}$) a vektor součtu pixelů po sloupcích $sumC$ (projekce $\mathcal{C}$). Vektory jsou indexovány dle šipek v obrázku.



Obrázek 2: Příklad rekonstruovaného binárního obrázku.

S využitím projekcí $\mathcal{R}$ a $\mathcal{C}$ však řešení úlohy není jednoznačné (viz obr. 3). V případě, že máme k dispozici data z dalších projekcí, je možné počet řešení omezit. Oba obrázky mají shodné vektory $sumR$ a $sumC$, avšak liší se ve vektorech součtů po diagonále $sumD$ a antidiagonále $sumA$. Tyto čtyři projekce však stále nezaručují věrnou rekonstrukci zcela libovolného obrázku. Obecně, se stoupajícím počtem využitých projekcí pro rekonstrukcí obrázku se zvyšuje pravděpodobnost úspěšnosti rekonstrukce obrázku.

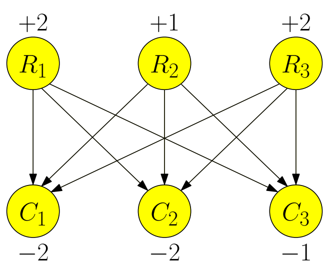
V naší úloze máme k dispozici pouze projekci po řádcích $\mathcal{R}$ a projekci po sloupcích $\mathcal{C}$. Dále však využijeme několika předpokladů o struktuře obrázku, které nám pomohou k jeho rekonstrukci.



Obrázek 3: Nejednoznačnost řešení rekonstruovaného binárního obrázku.

2.1 Reprezentace binárních obrázků pomocí toků v sítích

Jsou dány minimálně dvě různé projekce rekonstruovaného obrázku, v našem případě projekce $\mathcal{R}$ a $\mathcal{C}$. Pak lze rekonstrukci obrázku řešit jako úlohu hledání nejlevnějšího toku sítí (G, b, l, u, c) zobrazenou na obr. 4 odpovídající projekčnímu páru $\mathcal{RC}$. Vrcholy v horní části odpovídají první projekci (v tomto případě $\mathcal{R}$), vrcholy ve spodní části grafu odpovídají druhé projekci (v tomto případě $\mathcal{C}$). Hrany grafu G symbolizují pixely, např. hrana mezi vrcholy R_1 a C_3 odpovídá pixelu $(1, 3)$.



Obrázek 4: Příklad grafu G pro projekční pár $\mathcal{RC}$.

Obrázku o rozměrech $n_1 \times n_2$ pixelů odpovídá graf G, kde projekce $\mathcal{R}$ je zastoupena v grafu vrcholy, jejichž počet je roven délce vektoru $sumR$ (n_1 vrcholů), a podobně projekci $\mathcal{C}$ odpovídá n_2 vrcholů (délka vektoru $sumC$). V horní části sítě jsou vrcholy reprezentující zdroje (s kladným znaménkem), v dolní jsou umístěny spotřebiče (se záporným znaménkem).

2.2 Algoritmus pro rekonstrukci binárních obrázků

V předchozí sekci je popsáno jak reprezentovat obrázek pomocí toků v sítích na základě projekčního páru $\mathcal{RC}$. Algoritmus pro rekonstrukci obrázku pak vypadá následujícím způsobem:

Vstup: Vektory odpovídající projekcím $\mathcal{R}, \mathcal{C}$.
Výstup: Rekonstruovaný obrázek I.

I = nulová matice o rozměrech $n_1 \times n_2$;
c_{prev} = nulová matice o rozměrech $(n_1 + n_2) \times (n_1 + n_2)$;
naplň vektor b a matice l, u pro zvolený projekční pár;
for $ic = 1 : countOfIterations$ **do**
    naplň matici c pro zvolený projekční pár, kde $c = \text{fce}(c_{prev}, I)$;
    $c_{prev} = c$;
    vytvoř prázdný graf g;
    F = řešení úlohy nejlevnějšího toku v síti (g, b, l, u, c);
    transformuj F na obrázek I;
    zobraz obrázek I;
end

Pro projekční pár $\mathcal{RC}$ je vektor b následující:

```
>> b = [sumR -sumC]';
```

Dále je nutné naplnit matice u, l, c, tj. ke každé hraně e přiřadit trojici $\big(l(e), u(e), c(e)\big)$. Pokud je rekonstruovaný obrázek binární, pak platí $\forall e \in E(G) : l(e) = 0, u(e) = 1, c(e) \in \mathbb{R}$. Určením ceny hrany $c(e)$ se detailněji zabývá sekce 2.2.1. Po naplnění matic l, u, c reprezentující vybraný projekční pár pak hledáme přípustný tok s minimální cenou. K řešení je možné použít funkci z TORSCHE s názvem `mincostflow` následujícím způsobem:

```
% create empty graph
>> g = graph;
% generate b,l,u,c
% ...
% solve minimal cost flow problem
>> F = g.mincostflow(c,l,u,b);
```

Výstupem funkce `mincostflow` je matice F odpovídající přípustnému toku v síti (G, b, l, u, c) s minimální cenou. Matici toků F je nutné převést zpět na binární obrázek. V našem případě je transformace jednoduchá, např. pokud existuje nenulový tok v matici F mezi vrcholy R_1 a C_3, pak tento tok odpovídá přímo pixelu $(1, 3)$. Rekonstrukce je prováděna v iteracích dokud nedojde ke konvergenci rekonstruovaného obrázku (tj. obrázek se již dále nemění). K vykreslení obrázku I použijte kostru složenou z následujících funkcí:

```
>> subplot(..., ..., ...);
>> imagesc(logical(I));
>> colormap(gray);
>> axis off;
>> axis square;
```

2.2.1 Určení ceny hrany s ohledem na využítí znalosti struktury obrázku

Jedním z argumentů funkce `mincostflow` je matice cen c. Pro dvojice vrcholů mezi nimiž žádná hrana neexistuje (tj. neodpovídá žádnému pixelu obrázku) je cena vždy nulová. Pro ceny hran přítomných v grafu G platí následující pravidla:

1. Nastav počáteční cenu hrany $c(e) = 0$.
2. Pokud lze vybrat okolí (viz obr. 5) pixelu z obrázku z předchozí iterace o velikosti 3×3, pokračuj dále.
3. Pokud je pixel bílý a v jeho osmiokolí není žádný jiný pixel bílý, pak nastav cenu $c(e) = 1$.
4. Pokud je pixel bílý a v jeho osmiokolí je právě 1 pixel bílý, pak nastav cenu $c(e) =$ 0,2.
5. Pokud je pixel bílý a v jeho osmiokolí jsou právě 2 bílé pixely, pak nastav cenu $c(e) =$ 0,1.
6. Pokud je pixel černý a v jeho čtyřokolí je alespoň jedna dvojice protilehlých pixelů bílá, pak nastav cenu $c(e) =$ -0,1.
7. Výsledná cena hrany je dána součtem $c(e) = c(e) +$ 0,5 $\cdot\, c_{prev}(e)$, kde $c_{prev}(e)$ je cena hrany z předchozí iterace.

Obrázek 5: Určení ceny hrany na základě okolí bílého a černého pixelu.

Úkol: Z projekčních dat (vektory $sumR, sumC$) uložených v souboru `projectionData.mat` zrekonstruujte čtvercový binární obrázek (20×20 pixelů) pomocí nejlevnějšího toku v sítích popsaného v sekci 2. Při správné implemetaci úlohy bude dostatečné pro rekonstrukci obrázku nastavit $countOfIterations$ na hodnotu 40.

Rada: Zamyslete se nad tím, zda nelze graf G z daných dat vytvořit jiným způsobem. Touto jednoduchou modifikací je možné dospět k řešení úlohy již pro $countOfIterations = 23$.

Reference

[1] J. Demel, *Grafy a jejich aplikace*. Academia, second ed., 2002.

[2] B. H. Korte and J. Vygen, *Combinatorial Optimization: Theory and Algorithms*. Springer, fourth ed., 2008.

[3] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall; United States Ed edition, 1993.

[4] K. J. Batenburg, „A network flow algorithm for reconstructing binary images from discrete x-rays,“ *J. Math. Imaging Vis.*, vol. 27, no. 2, pp. 175–191, 2007.