

3.1 Čo je dualita

V predchádzajúcich častiach sme sa snažili ukázať, že lineárne programovanie je silný a flexibilný nástroj pre riešenie optimalizačných problémov. Teraz sa zoznámime s ďalšou jeho podstatnou vlastnosťou, ktorá sa bude dať tiež mnohostranne využiť pri návrhu algoritmov. Tou vlastnosťou je tzv. *dualita*. Predpokladajme, že Róbert chce nájsť minimum funkcie

$$f(x_1, x_2, x_3) := 10x_1 + 3x_2 + 5x_3$$

spomedzi takých hodnôt x_1, x_2, x_3 , ktoré spĺňajú

$$6x_1 + x_2 - x_3 \geq 2 \quad (39)$$

$$2x_1 + 2x_2 + 6x_3 \geq 8 \quad (40)$$

$$6x_1 + 3x_2 + 5x_3 = 30$$

$$x_1, x_2, x_3 \geq 0$$

Angela vidí, že je to lineárny program a ľahko nájde minimum. Otázka je, ako má o tom presvedčiť Róberta, ktorý lineárne programovanie nepozná. O hornom odhade ho presvedčí ľahko: ukáže mu ako *svedka* nejaké konkrétne riešenie. Angela môže napríklad vybrať vektor $x_1 = 2, x_2 = 6, x_3 = 0$ a Róbert z toho ľahko zistí, že hľadané minimum je nanajvýš 38. Ako ale nájsť *svedka* pre odhad z opačnej strany? Angela môže napríklad skúsiť takýto argument: "Pozrime sa na riadok (42). Akékoľvek nezáporné čísla x_1, x_2, x_3 zvolíme, vždy bude platiť $6x_1 \leq 10x_1, x_2 \leq 3x_2$ a $-x_3 \leq 5x_3$. Preto $6x_1 + x_2 - x_3 \leq f(x_1, x_2, x_3)$. Lenže podľa (42) je $6x_1 + x_2 - x_3 \geq 2$, takže každá prípustná hodnota (a teda aj minimum) funkcie f je aspoň 2". Angela môže pokračovať napríklad tak, že sčíta riadky (42) a (43) a použije rovnaký argument. Môže aj každý riadok prenásobiť nejakým číslom (v riadkoch (42) a (43) to musí byť nezáporné číslo) a sčítať ich dokopy; ak platí, že výsledná lineárna kombinácia ľavých strán obmedzení je po zložkách menšia ako funkcia f , lineárna kombinácia pravých strán tvorí dolný odhad minima. Aby poskytla čo najlepší argument, Angela vlastne rieši lineárny program: nájsť maximum funkcie

$$g(y_1, y_2, y_3) := 2y_1 + 8y_2 + 30y_3$$

spomedzi takých hodnôt y_1, y_2, y_3 , ktoré spĺňajú

$$6y_1 + 2y_2 + 6y_3 \leq 10$$

$$y_1 + 2y_2 + 3y_3 \leq 3$$

$$-y_1 + 6y_2 + 5y_3 \leq 5$$

$$y_1, y_2 \geq 0$$

V tomto prípade je na jednej strane *svedok* $x_1 = 0, x_2 = 5, x_3 = 3$, že hľadané minimum f je najviac 30, na druhej strane *svedok* $y_1 = y_2 = 0, y_3 = 1$ ukazuje, že 30 je skutočná minimálna hodnota f . Je to náhoda, že sa podarilo touto metódou nájsť tesný odhad?

Pozrime sa všeobecnejšie na to, čo sa dialo. Majme nasledovný lineárny program, ktorý nazveme *primárny*

minimalizovať	c_1x_1	$+c_2x_2$	$+\cdots$	$+c_nx_n$	
pri obmedzeniach	$a_{1,1}x_1$	$+a_{1,2}x_2$	$+\cdots$	$+a_{1,n}x_n$	$\geq b_1$
	$a_{2,1}x_1$	$+a_{2,2}x_2$	$+\cdots$	$+a_{2,n}x_n$	$\geq b_2$
	\vdots	\vdots	\ddots	\vdots	\vdots
	$a_{m,1}x_1$	$+a_{m,2}x_2$	$+\cdots$	$+a_{m,n}x_n$	$\geq b_m$
				x_1, \dots, x_n	≥ 0

alebo skrátené

$$(P) : \min_{\mathbf{x} \in \mathbb{R}^n} \{ \mathbf{c}^T \mathbf{x} \mid A\mathbf{x} \geq \mathbf{b}, \mathbf{x} \geq \mathbf{0} \}.$$

K nemu sme zostrojili *duálny* lineárny program

$$\begin{array}{llllll} \text{maximalizovať} & b_1 y_1 & + b_2 y_2 & + \cdots & + b_m y_m & \\ \text{pri obmedzeniach} & a_{1,1} y_1 & + a_{2,1} y_2 & + \cdots & + a_{m,1} y_m & \leq c_1 \\ & a_{1,2} y_1 & + a_{2,2} y_2 & + \cdots & + a_{m,2} y_m & \leq c_2 \\ & \vdots & \vdots & \ddots & \vdots & \vdots \\ & a_{1,n} y_1 & + a_{2,n} y_2 & + \cdots & + a_{m,n} y_n & \leq c_n \\ & & & & y_1, \dots, y_m & \geq 0 \end{array}$$

alebo skrátené

$$(D) : \max_{\mathbf{y} \in \mathbb{R}^m} \{ \mathbf{b}^T \mathbf{y} \mid A^T \mathbf{y} \leq \mathbf{c}, \mathbf{y} \geq \mathbf{0} \}.$$

Z úvah, ktoré sme robili, by mala byť zrejmá nasledovná veta, ktorá sa zvykne volať aj *slabá veta o dualite*:

Veta 3.1. Ak \mathbf{x} je ľubovoľné prípustné riešenie primárnej úlohy (P) a \mathbf{y} je ľubovoľné prípustné riešenie duálnej úlohy (D), potom platí

$$\mathbf{c}^T \mathbf{x} \geq \mathbf{b}^T \mathbf{y}$$

Dôkaz:.

$$\mathbf{c}^T \mathbf{x} \geq (\mathbf{A}^T \mathbf{y})^T \mathbf{x} = \mathbf{y}^T A\mathbf{x} \geq \mathbf{y}^T \mathbf{b} \quad (41)$$

Prvá nerovnosť vyplýva z podmienok duálnej úlohy, pretože $\mathbf{c} \geq A^T \mathbf{y}$ a druhá nerovnosť z podmienok primárnej úlohy, pretože $A\mathbf{x} \geq \mathbf{b}$. \square

Veta 3.1 hovorí len to, ako sme duálny program konštruovali: všetky prípustné riešenia (D) sú dolné odhady minima (P). V našom príklade však navyše platilo, že optimálne riešenia primárnej a duálnej úlohy sa rovnali. Naším cieľom bude ukázať, že to nebola náhoda. Ešte predtým ale niekoľko poznámok o konštrukcii duálnej úlohy. V našom príklade sme vychádzali z minimalizačnej úlohy a hľadali sme najväčší dolný odhad minima, preto duálna úloha bola maximalizačná. Všetky naše úvahy ale boli z tohto hľadiska symetrické: keby sme vychádzali z maximalizačnej úlohy, rovnakým spôsobom by sme hľadali najmenší horný odhad maxima a dostali by sme duálnu minimalizačnú úlohu. Čitateľ sa ľahko presvedčí, že v tomto všeobecnejšom chápaní duality je pojem primárnej a duálnej úlohy symetrický: ak by sme za primárnu úlohu zobrali program (D) a konštruovali duálnu úlohu, dostali by sme (P). Vo všeobecnosti teda platí, že duálna úloha k duálnej úlohe je primárna úloha.

Pozrime sa ešte na tvar obmedzení. V motivačnom príklade sme mali v primárnej úlohe obmedzenia v tvare nerovnosti (\geq) aj rovnosti. V prvom prípade museli byť príslušné multiplikátory y_i nezáporné, v prípade rovnosti mohli byť ľubovoľné. Zároveň všetky premenné x_i v primárnej úlohe boli nezáporné, a preto sme potrebovali, aby každé obmedzenie duálnej úlohy bolo v tvare nerovnosti. Čo by sa stalo, ak by niektorá premenná, napríklad x_1 mohla byť aj záporná? Podobne, ako keď sme odvodzovali normálny tvar lineárneho programu, mohli by sme nahradiť $x_1 = z_1 - z_2$, $z_1, z_2 \geq 0$ a dostali by sme primárnu úlohu minimalizovať

$$f(z_1, z_2, x_2, x_3) := 10z_1 - 10z_2 + 3x_2 + 5x_3$$

spomedzi takých hodnôt z_1, z_2, x_2, x_3 , ktoré spĺňajú

$$6z_1 - 6z_2 + x_2 - x_3 \geq 2 \quad (42)$$

$$2z_1 - 2z_2 + 2x_2 + 6x_3 \geq 8 \quad (43)$$

$$6z_1 - 6z_2 + 3x_2 + 5x_3 = 30$$

$$z_1, z_2, x_2, x_3 \geq 0$$

Príslušná duálna úloha by bola nájsť maximum funkcie

$$g(y_1, y_2, y_3) := 2y_1 + 8y_2 + 30y_3$$

spomedzi takých hodnôt y_1, y_2, y_3 , ktoré spĺňajú

$$\begin{aligned} 6y_1 + 2y_2 + 6y_3 &\leq 10 \\ -6y_1 - 2y_2 - 6y_3 &\leq -10 \\ y_1 + 2y_2 + 3y_3 &\leq 3 \\ -y_1 + 6y_2 + 5y_3 &\leq 5 \\ y_1, y_2 &\geq 0 \end{aligned}$$

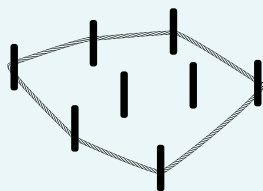
pričom prvé dve nerovnosti môžeme zlúčiť do jednej rovnosti $6y_1 + 2y_2 + 6y_3 = 10$. Keď zhrnieme naše doterajšie úvahy, konštrukciu duálnej úlohy môžeme opísať nasledovným receptom:

primárna úloha		duálna úloha	
minimalizovať	$\mathbf{c}^T \mathbf{x}$	maximalizovať	$\mathbf{b}^T \mathbf{y}$
i -te obmedzenie tvaru	$\sum_{j=1}^n a_{ij} x_j = b_i$	i -ta premenná	$y_i \in \mathbb{R}$
i -te obmedzenie tvaru	$\sum_{j=1}^n a_{ij} x_j \geq b_i$	i -ta premenná	$y_i \geq 0$
j -ta premenná	$x_j \in \mathbb{R}$	j -te obmedzenie tvaru	$\sum_{i=1}^m a_{ij} y_i = c_j$
j -ta premenná	$x_j \geq 0$	j -te obmedzenie tvaru	$\sum_{i=1}^m a_{ij} y_i \leq c_j$

Skôr, ako pokročíme v našich úvahách, urobme malú odbočku:

Malá odbočka ku konvexným obalom

Čitateľ sa už možno stretol s konvexným obalom v dvoch rozmeroch: pre dané body v rovine je ich konvexný obal najmenší konvexný mnohoúhelník, ktorý ich všetky obsahuje.



Ak si body predstavíme ako kolíky, ktoré omotáme špagátom, tak kolíky, na ktorých sa špagát zachytí, tvoria vrcholy konvexného obalu. Takáto predstava ešte celkom dobre zafunguje v troch rozmeroch, kde ako keby sme balili body do baliaceho papiera. V n rozmeroch a s $n - 1$ -rozmerným papierom je to ale komplikovanejšie; nie všetky vlastnosti, na ktoré sme z 2 rozmerov zvyknutí, platia aj v n rozmeroch, a preto si treba v úvahách dávať obzvlášť pozor na argumenty typu “je jasné, že...”.

Pripomenieme, že konvexné teleso \mathcal{T} je také, že pre každé dva body $\mathbf{x}, \mathbf{y} \in \mathcal{T}$ je celá úsečka medzi nimi v \mathcal{T} , t.j. všetky body tvaru $t\mathbf{x} + (1 - t)\mathbf{y}$ pre $0 \leq t \leq 1$ sú v \mathcal{T} . Definujme konvexný obal takto:

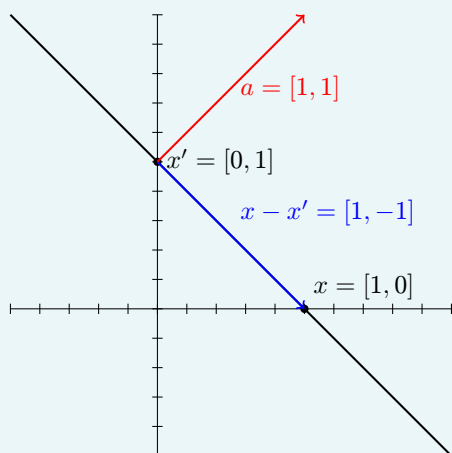
Definícia 3.2. Konvexný obal n bodov $\mathbf{a}_1, \dots, \mathbf{a}_n$ je prienik všetkých konvexných telies, ktoré obsahujú body $\mathbf{a}_1, \dots, \mathbf{a}_n$.

Konvexných telies, ktoré obsahujú dané body je síce nespočítateľne veľa, ale ich prienik je dobre definovaný, takže nás to netrápi. Navyše, prienik ľubovoľných konvexných telies je zjavne opäť konvexné teleso, takže naša definícia je dobrá v tom, že konvexný obal je, tak ako sa patrí, konvexný. Bude sa nám hodiť aj nasledovná charakterizácia:

Lema 3.3. *Konvexný obal n bodov $\mathbf{a}_1, \dots, \mathbf{a}_n$ je tvorený bodmi konvexnými kombináciami bodov $\mathbf{a}_1, \dots, \mathbf{a}_n$, t.j. bodmi $z_1\mathbf{a}_1 + \dots + z_n\mathbf{a}_n$, kde $z_1, \dots, z_n \in \mathbb{R}^+$ a $z_1 + \dots + z_n = 1$.*

Dôkaz:. Označme K množinu všetkých konvexných kombinácií bodov $\mathbf{a}_1, \dots, \mathbf{a}_n$. Lahko sa overí, že K je konvexné teleso: pre konvexné kombinácie $\sum z_i \mathbf{a}_i$ a $\sum z'_i \mathbf{a}_i$ je aj $t \sum z_i \mathbf{a}_i + (1-t) \sum z'_i \mathbf{a}_i = \sum (tz_i + (1-t)z'_i) \mathbf{a}_i$ konvexná kombinácia, a teda patrí do K . Pretože K je konvexné teleso a obsahuje body $\mathbf{a}_1, \dots, \mathbf{a}_n$, z definície K je nadmnožina konvexného obalu. Na dôkaz lemy nám stačí ukázať, že každá konvexná kombinácia patrí do konvexného obalu.

Dokážeme to indukciou na počet bodov n . Pre $n = 1$ je to jasné priamo z definície konvexného obalu, pre $n = 2$ sú konvexné kombinácie $z_1\mathbf{a}_1 + (1-z_1)\mathbf{a}_2$ a tieto z definície konvexnosti ležia v každom konvexnom telese, ktoré obsahuje \mathbf{a}_1 a \mathbf{a}_2 (a teda ležia aj v ich prieniku). Nech teraz $n \geq 3$ a majme konvexnú kombináciu $\mathbf{x} = \sum z_i \mathbf{a}_i$. Ak $z_n = 1$, $\mathbf{x} = \mathbf{a}_n$ a z definície \mathbf{x} je v konvexnom obale. Nech teda $z_n < 1$. Označme $z'_i := \frac{z_i}{1-z_n}$ a zoberme bod $\mathbf{x}' = \sum z'_i \mathbf{a}_i$. \mathbf{x}' je konvexná kombinácia bodov $\mathbf{a}_1, \dots, \mathbf{a}_{n-1}$, a preto podľa indukčného predpokladu každé konvexné teleso, ktoré obsahuje body $\mathbf{a}_1, \dots, \mathbf{a}_{n-1}$, obsahuje aj \mathbf{x}' . Zároveň $\mathbf{x} = (1-z_n)\mathbf{x}' + z_n\mathbf{a}_n$, a preto každé konvexné teleso, ktoré obsahuje \mathbf{x}' a \mathbf{a}_n , obsahuje aj \mathbf{x} . \square



Ešte pripomeňme, že ak sme v m -rozmernom priestore, nadrovina \mathcal{H} je množina bodov \mathbf{x} spĺňajúcich rovnosť $a_1x_1 + \dots + a_mx_m = 1$ (jednotka na pravej strane je takmer všeobecná: ľubovoľná nadrovina, ktorá neprechádza bodom $[0, \dots, 0]$ sa dá normovaním upraviť na tento tvar). Ak označíme $\langle \cdot, \cdot \rangle$ skalárny súčin, je \mathcal{H} definovaná ako $\langle \mathbf{x}, \mathbf{a} \rangle = 1$. Body, pre ktoré $\langle \mathbf{x}, \mathbf{a} \rangle < 1$ sú pod \mathcal{H} , a body $\langle \mathbf{x}, \mathbf{a} \rangle > 1$ nad \mathcal{H} . Vektor \mathbf{a} je *normálový vektor* \mathcal{H} a pre ľubovoľné dva body \mathbf{x}, \mathbf{x}' z \mathcal{H} platí,

$$\langle \mathbf{x} - \mathbf{x}', \mathbf{a} \rangle = \sum_{i=1}^m (x_i - x'_i) a_i = \langle \mathbf{x}, \mathbf{a} \rangle - \langle \mathbf{x}', \mathbf{a} \rangle = 0.$$

V dvoch rozmeroch je nadrovinou priamka, tá na obrázku vľavo je daná rovnicou $x_1 + x_2 = 1$.

Naše úvahy o konvexných obaloch zavíšieme nasledovným pozorovaním, ktoré je pre dva a tri rozmery úplne zjavné:

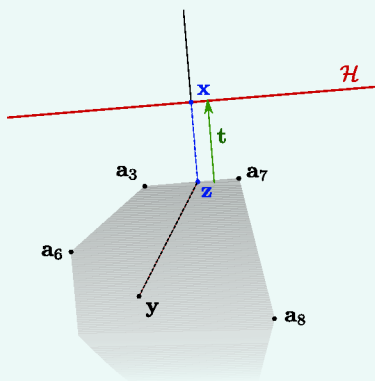
Lema 3.4. *Majme body $\mathbf{x}, \mathbf{a}_1, \dots, \mathbf{a}_n$ v m -rozmernom priestore. Nech K je konvexný obal $\mathbf{a}_1, \dots, \mathbf{a}_n$. Potom platí:*

- Ak \mathbf{x} je vo vnútri¹ K , potom neexistuje nadrovina \mathcal{H} , ktorá prechádza cez \mathbf{x} a všetky body z K ležia na jednej strane \mathcal{H} .
- Ak $\mathbf{x} \notin K$, potom existuje nadrovina \mathcal{H} , ktorá prechádza cez \mathbf{x} a všetky body z K ležia na jednej strane \mathcal{H} .

Dôkaz:. Prvá časť je ľahká: nech \mathbf{x} leží vo vnútri K nech existuje taká nadrovina \mathcal{H} , že všetky body z K sú na jednej strane \mathcal{H} . Pretože \mathbf{x} je vo vnútri K , existuje nejaký bod $\mathbf{x}' \in K$, ktorý leží na opačnej strane \mathcal{H} ako všetky body z K , špeciálne ako body \mathbf{a}_i . Lenže polpriestor ohraničený \mathcal{H} , ktorý obsahuje všetky \mathbf{a}_i je konvexné teleso, a preto z definície konvexného obalu $\mathbf{x}' \notin K$.

Ideme dokázať druhú časť. Majme fixovaný bod \mathbf{x} a pre body $\mathbf{y} \in K$ uvažujme euklidovskú vzdialenosť $f(\mathbf{y}) := \|\mathbf{y} - \mathbf{x}\|$. Pretože K je kompaktná množina a f je spojitá funkcia nadobúdajúca kladné hodnoty, existuje bod $\mathbf{z} \in K$, na ktorom f nadobúda minimum. Označme

$$\mathbf{t} = \mathbf{x} - \mathbf{z} \quad \kappa = \langle \mathbf{t}, \mathbf{x} \rangle$$



Nech \mathcal{H} je nadrovina tvorená bodmi \mathbf{y} , pre ktoré $\langle \mathbf{t}, \mathbf{y} \rangle = \kappa$. Zjavne $\mathbf{x} \in \mathcal{H}$. Platí

$$0 \leq \sum_{i=1}^m (x_i - z_i)^2 = \langle \mathbf{x}, \mathbf{x} \rangle - 2 \langle \mathbf{x}, \mathbf{z} \rangle + \langle \mathbf{z}, \mathbf{z} \rangle$$

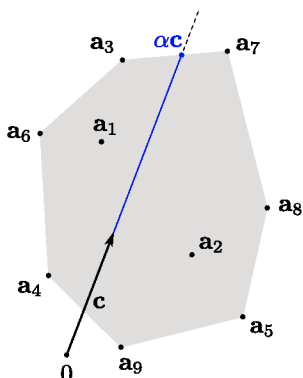
a preto

$$\begin{aligned} \langle \mathbf{t}, \mathbf{z} \rangle &= \langle \mathbf{x} - \mathbf{z}, \mathbf{z} \rangle = \langle \mathbf{x}, \mathbf{z} \rangle - \langle \mathbf{z}, \mathbf{z} \rangle \leq \\ &\leq \langle \mathbf{x}, \mathbf{x} \rangle - \langle \mathbf{x}, \mathbf{z} \rangle = \langle \mathbf{x} - \mathbf{z}, \mathbf{x} \rangle = \langle \mathbf{t}, \mathbf{x} \rangle = \kappa \end{aligned}$$

Teraz stačí ukázať, že pre všetky body $\mathbf{y} \in K$ tiež platí $\langle \mathbf{t}, \mathbf{y} \rangle \leq \kappa$, a teda ležia na rovnakej strane \mathcal{H} ako \mathbf{z} .

Majme teda $\mathbf{z} \in K$. Keďže \mathbf{z} minimalizuje vzdialenosť od \mathbf{x} a K je konvexné teleso, pre všetky $0 \leq \lambda \leq 1$ platí $\|\mathbf{z} - \mathbf{x}\|^2 \leq \|(1 - \lambda)\mathbf{z} + \lambda\mathbf{y} - \mathbf{x}\|^2 = \|(1 - \lambda)(\mathbf{z} - \mathbf{x}) + \lambda(\mathbf{y} - \mathbf{x})\|^2$ a odtiaľ po priamočiarych úpravách $0 \leq \lambda(\lambda - 2)\|\mathbf{z} - \mathbf{x}\|^2 + 2\lambda(1 - \lambda)\langle \mathbf{z} - \mathbf{x}, \mathbf{y} - \mathbf{x} \rangle + \lambda^2\|\mathbf{y} - \mathbf{x}\|^2$. Pretože $\lambda \geq 0$, máme $0 \leq (\lambda - 2)\|\mathbf{z} - \mathbf{x}\|^2 + 2(1 - \lambda)\langle \mathbf{z} - \mathbf{x}, \mathbf{y} - \mathbf{x} \rangle + \lambda\|\mathbf{y} - \mathbf{x}\|^2$ a odtiaľ v limite pre $\lambda \mapsto 0$ máme $0 \leq \langle \mathbf{z} - \mathbf{x}, \mathbf{y} - \mathbf{x} \rangle - \|\mathbf{z} - \mathbf{x}\|^2 = \langle \mathbf{z} - \mathbf{x}, \mathbf{y} - \mathbf{x} \rangle - \langle \mathbf{z} - \mathbf{x}, \mathbf{z} - \mathbf{x} \rangle = \langle \mathbf{z} - \mathbf{x}, \mathbf{y} \rangle - \langle \mathbf{z} - \mathbf{x}, \mathbf{z} \rangle = \langle \mathbf{t}, \mathbf{z} \rangle - \langle \mathbf{t}, \mathbf{y} \rangle$. Preto $\langle \mathbf{t}, \mathbf{y} \rangle \leq \langle \mathbf{t}, \mathbf{z} \rangle \leq \kappa$, čo sme chceli dokázať. \square

Vráťme sa teraz k nášmu cieľu – ukázať, že rovnaká hodnota optimálneho riešenia v primárnej aj duálnej úlohe nie je náhoda. Pozrime sa na nasledovný príklad: Majme m -rozmerný priestor \mathbb{R}^m a v ňom n bodov $\mathbf{a}_1, \dots, \mathbf{a}_n$ (t.j. bod \mathbf{a}_i má súradnice $(a_{i1}, a_{i2}, \dots, a_{im})$) a vektor \mathbf{c} . Cieľom je nájsť čo najväčšie číslo $\alpha \in \mathbb{R}^+$ tak, že bod $\alpha\mathbf{c}$ leží v konvexnom obale K bodov $\mathbf{a}_1, \dots, \mathbf{a}_n$.



Predpokladajme pre jednoduchosť, že úloha má riešenie, t.j. polpriamka generovaná vektorom \mathbf{c} pretína K . Z konvexity vyplýva, že prienik K a polpriamky je úsečka. Optimálne riešenie je preto bod $\alpha\mathbf{c}$, v ktorom polpriamka opúšťa K . Podľa Lemy 3.3 sa body konvexného obalu sa dajú vyjadriť ako konvexná kombinácia bodov $\mathbf{a}_1, \dots, \mathbf{a}_n$, t.j. náš hľadaný bod $\alpha\mathbf{c}$ sa musí dať napísať v tvare $z_1\mathbf{a}_1 + \dots + z_n\mathbf{a}_n$ pre nejaké $z_1, \dots, z_n \geq 0$, kde $\sum_{i=1}^n z_i = 1$. Pre ľubovoľný prípustný bod $\alpha\mathbf{c}$ si označme $y_j = \frac{z_j}{\alpha}$, potom platí

$$y_j \geq 0 \quad \sum_{j=1}^n y_j = \frac{1}{\alpha} \quad \alpha\mathbf{c} = \alpha \sum_{j=1}^n \mathbf{a}_j y_j.$$

Nájsť bod, pre ktorý je α maximálne, znamená nájsť bod, pre ktorý je $1/\alpha$ minimálne. Snažíme sa teda minimalizovať hodnotu $\sum_{j=1}^n y_j$ spomedzi takých $y_1, \dots, y_n \geq 0$,

pre ktoré $\sum_{j=1}^n \mathbf{a}_j y_j = \mathbf{c}$; stačí si uvedomiť, že z daných hodnôt \mathbf{y} vieme zrekonštruovať \mathbf{z} a α . Našu úlohu teda vieme zapísať ako úlohu lineárneho programovania:

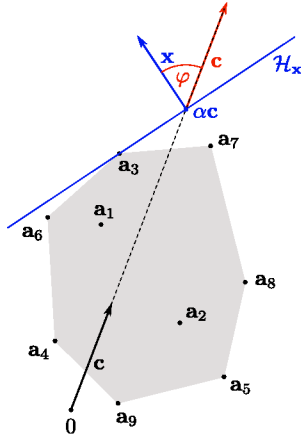
$$\begin{array}{llllll} \text{minimalizovať} & y_1 & + y_2 & + \cdots & + y_n & \\ \text{pri obmedzeniach} & a_{1,1}y_1 & + a_{2,1}y_2 & + \cdots & + a_{n,1}y_n & = c_1 \\ & a_{1,2}y_1 & + a_{2,2}y_2 & + \cdots & + a_{n,2}y_n & = c_2 \\ & \vdots & \vdots & \ddots & \vdots & \vdots \\ & a_{1,m}y_1 & + a_{2,m}y_2 & + \cdots & + a_{n,m}y_n & = c_m \\ & & & & y_1, \dots, y_n & \geq 0 \end{array} \quad (44)$$

alebo skrátené

$$(P) \min_{\mathbf{y} \in \mathbb{R}^n} \{ \mathbf{1}^T \mathbf{y} \mid A^T \mathbf{y} = \mathbf{c}, \mathbf{y} \geq \mathbf{0} \},$$

kde A je matica rozmerov $n \times m$, ktorej riadky sú tvorené súradnicami bodov $\mathbf{a}_1, \dots, \mathbf{a}_n$. Na program (44) použijeme náš dualizačný recept a dostaneme duálny program

$$(D) \max_{\mathbf{x} \in \mathbb{R}^m} \{ \mathbf{c}^T \mathbf{x} \mid A\mathbf{x} \leq \mathbf{1} \} \quad (45)$$



Ako môžeme program (45) interpretovať? Zoberme si ľubovoľné prípustné riešenie \mathbf{x} . Nech $\mathcal{H}_{\mathbf{x}}$ je nadrovina daná bodmi \mathbf{y} , pre ktoré $\langle \mathbf{y}, \mathbf{x} \rangle = 1$. Obmedzenia programu (45) hovoria, že pre každý bod \mathbf{a}_i je $\langle \mathbf{a}_i, \mathbf{x} \rangle \leq 1$, a z konvexity je preto celý konvexný obal K pod $\mathcal{H}_{\mathbf{x}}$. Ďalej ak označíme $\alpha = \frac{1}{\mathbf{c}^T \mathbf{x}}$, tak bod $\alpha \mathbf{c}$ leží v nadrovine $\mathcal{H}_{\mathbf{x}}$, lebo $\langle \alpha \mathbf{c}, \mathbf{x} \rangle = 1$. Každému prípustnému riešeniu (45) teda zodpovedá bod $\alpha \mathbf{c}$ na polpriamke generovanej vektorom \mathbf{c} , ktorý (podľa Lemy 3.4) neleží vo vnútri K . Naviac je zrejmé, že optimálne riešenie je nezáporné, a preto môžeme bez ujmy na všeobecnosti predpokladať, že $\langle \mathbf{x}, \mathbf{c} \rangle = \|\mathbf{x}\| \cdot \|\mathbf{c}\| \cos \varphi \geq 0$, kde φ je uhol, ktorý zvierajú vektory \mathbf{x} a \mathbf{c} . Preto nás zaujímajú iba tie prípustné riešenia, ktorým zodpovedajú body $\alpha \mathbf{c}$ ležiace na polpriamke generovanej \mathbf{c} za K .

Platí to aj naopak. Zoberme si hocikáký bod $\alpha \mathbf{c}$ ležiaci za K . Podľa Lemy 3.4 existuje nadrovina \mathcal{H} taká, že všetky body z K ležia pod ňou. Nech \mathcal{H} je tvorená bodmi \mathbf{y} spĺňajúcimi $\langle \mathbf{x}, \mathbf{y} \rangle = 1$ pre nejaký vektor \mathbf{x} , potom platí $A\mathbf{x} \leq \mathbf{1}$. Zároveň, pretože \mathcal{H} prechádza cez $\alpha \mathbf{c}$, platí $\langle \mathbf{x}, \alpha \mathbf{c} \rangle = 1 = \alpha \langle \mathbf{x}, \mathbf{c} \rangle$, a teda $\alpha = \frac{1}{\mathbf{c}^T \mathbf{x}}$. Pre maximálnu hodnotu $\mathbf{c}^T \mathbf{x}$ je príslušná α minimálna, a preto program (45) vyžaduje najšť minimálne α tak, aby bol $\alpha \mathbf{c}$ ležal za K na polpriamke generovanej vektorom \mathbf{c} . To je ale zjavne bod, v ktorom polpriamka opúšťa K , a teda programy (44) a (45) majú rovnaké optimálne riešenie. Dokázali sme teda tvrdenie

Lema 3.5. Ak primárny program $\min_{\mathbf{y} \in \mathbb{R}^n} \{ \mathbf{1}^T \mathbf{y} \mid A^T \mathbf{y} = \mathbf{c}, \mathbf{y} \geq \mathbf{0} \}$ má prípustné riešenie, potom aj duálny program $\max_{\mathbf{x} \in \mathbb{R}^m} \{ \mathbf{c}^T \mathbf{x} \mid A\mathbf{x} \leq \mathbf{1} \}$ má prípustné riešenie a optimálne hodnoty oboch programov sa rovnajú.

Keby sa nám podarilo Lemu 3.5 zovšeobecniť na programy tvaru $\min_{\mathbf{y} \in \mathbb{R}^n} \{ \mathbf{b}^T \mathbf{y} \mid A^T \mathbf{y} = \mathbf{c}, \mathbf{y} \geq \mathbf{0} \}$ pre ľubovoľný vektor \mathbf{b} , boli by sme spokojní, lebo každý lineárny program sa dá napísať v takomto tvare. Majme teda primárny program

$$(P) \min_{\mathbf{y} \in \mathbb{R}^n} \{ \mathbf{b}^T \mathbf{y} \mid A^T \mathbf{y} = \mathbf{c}, \mathbf{y} \geq \mathbf{0} \} \quad (46)$$

a k nemu duálny program

$$(D) \max_{\mathbf{x} \in \mathbb{R}^m} \{ \mathbf{c}^T \mathbf{x} \mid A\mathbf{x} \leq \mathbf{b} \} \quad (47)$$

Uvažujme najprv vektory \mathbf{b} také, že všetky zložky $b_i > 0$. Označme

$$\mathbf{a}'_i = \frac{1}{b_i} \mathbf{a}_i \quad y'_j = b_j y_j \quad (48)$$

Platí $\mathbf{b}^\top \mathbf{y} = \sum_{j=1}^n b_j y_j = \mathbf{1}^\top \mathbf{y}'$ a pre každé $i \in \{1, \dots, m\}$ $\sum_{j=1}^n a_{ji} y_j = \sum_{j=1}^n a'_{ji} b_i \frac{y'_j}{b_j}$. Pretože $b_j > 0$, program (46) je ekvivalentný² s programom

$$\min_{\mathbf{y}' \in \mathbb{R}^n} \{ \mathbf{1}^\top \mathbf{y}' \mid A' \mathbf{y}' = \mathbf{c}, \mathbf{y}' \geq \mathbf{0} \} \quad (49)$$

kde stĺpce matice A' sú vektory \mathbf{a}'_i . Podľa Lemy 3.5, ak má program (49) prípustné riešenie, tak jeho optimum je rovnaké ako optimum programu

$$\max_{\mathbf{x}' \in \mathbb{R}^m} \{ \mathbf{c}^\top \mathbf{x}' \mid A'^\top \mathbf{x}' \leq \mathbf{1} \} \quad (50)$$

Obmedzenia programu (50) sú v tvare $\sum_{j=1}^m a'_{ji} x'_j \leq 1$ a tak s použitím označenia (48) dostávame, že programy (50) a (47) sú ekvivalentné, preto ak (46) má prípustné riešenie, má ho aj (47) a hodnoty optima sú rovnaké.

Cvičenie. Upravte predchádzajúci postup tak, aby platil pre $b_i \geq 0$.

Nech teraz \mathbf{b} nadobúda ľubovoľné hodnoty a nech $\tilde{\mathbf{x}}$ je prípustné riešenie (47). Označme

$$\mathbf{x}' = \mathbf{x} - \tilde{\mathbf{x}} \quad b'_i = b_i - \mathbf{a}_i^\top \tilde{\mathbf{x}} \quad (51)$$

Ako sa v tomto označení zmenia programy (46) a (47)? Pre prípustné riešenia \mathbf{x}, \mathbf{y} platí

$$\begin{aligned} \mathbf{c}^\top \mathbf{x} &= \mathbf{c}^\top \mathbf{x}' + \mathbf{c}^\top \tilde{\mathbf{x}} \\ \mathbf{b}^\top \mathbf{y} &= \sum_{i=1}^n b_i y_i = \sum_{i=1}^n b'_i y_i + \sum_{i=1}^n y_i (\mathbf{a}_i^\top \tilde{\mathbf{x}}) = \mathbf{b}'^\top \mathbf{y} + \sum_{i=1}^n y_i \sum_{j=1}^m a_{i,j} \tilde{x}_j = \\ &= \mathbf{b}'^\top \mathbf{y} + \sum_{j=1}^m \tilde{x}_j \left(\sum_{i=1}^n y_i a_{i,j} \right) = \mathbf{b}'^\top \mathbf{y} + \mathbf{c}^\top \tilde{\mathbf{x}} \end{aligned}$$

kde posledná rovnosť vyplýva z toho, že $A^\top \mathbf{y} = \mathbf{c}$. Pretože $\mathbf{c}^\top \tilde{\mathbf{x}}$ je konštanta, programy (46) a (47) vieme ekvivalentne zapísať ako

$$(P') \quad \min_{\mathbf{y} \in \mathbb{R}^n} \{ \mathbf{b}'^\top \mathbf{y} \mid A^\top \mathbf{y} = \mathbf{c}, \mathbf{y} \geq \mathbf{0} \} \quad (D') \quad \max_{\mathbf{x}' \in \mathbb{R}^m} \{ \mathbf{c}^\top \mathbf{x}' \mid A \mathbf{x}' \leq \mathbf{b}' \}$$

Navyše z prípustnosti $\tilde{\mathbf{x}}$ pre (47) vyplýva, že $\mathbf{b}' \geq \mathbf{0}$ a preto môžeme aplikovať predchádzajúci prípad.

Načrtli sme³ dôkaz fundamentálnej vety v teórii lineárneho programovania:

Veta 3.6 (Silná veta o dualite). *Pre dvojicu duálnych lineárnych programov*

$$(P) \quad \min_{\mathbf{y} \in \mathbb{R}^n} \{ \mathbf{b}^\top \mathbf{y} \mid A^\top \mathbf{y} = \mathbf{c}, \mathbf{y} \geq \mathbf{0} \} \quad (D) \quad \max_{\mathbf{x} \in \mathbb{R}^m} \{ \mathbf{c}^\top \mathbf{x} \mid A \mathbf{x} \leq \mathbf{b} \}$$

Platí práve jedna zo štyroch možností:

1. (P) ani (D) nemajú žiadne prípustné riešenie
2. (P) je neohraničený a (D) nemá prípustné riešenie
3. (D) je neohraničený a (P) nemá prípustné riešenie
4. (P) aj (D) majú prípustné riešenie. V tom prípade sa optimálne hodnoty (P) a (D) rovnajú.

²v tom zmysle, že každému riešeniu programu (46) prislúcha nejaké riešenie programu (49) s rovnakou hodnotou a naopak

³pre kompletný dôkaz treba ošetriť ešte niekoľko špeciálnych prípadov, ktoré prenechávame na čitateľa

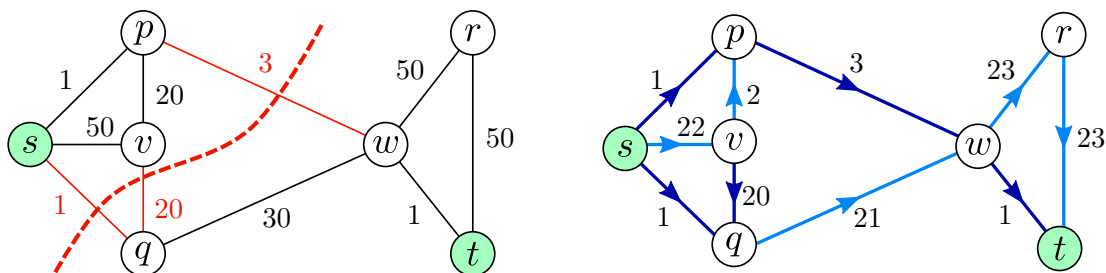
3.2 MAX-FLOW–MIN-CUT očami duality

Dualita lineárneho programovania, ktorú sme predstavili v predchádzajúcej časti, často pomáha lepšie pochopiť min–max charakterizácie výpočtových problémov. V tejto časti si z pohľadu duality priblížime známe tvrdenie, že veľkosť maximálneho toku sa rovná kapacite minimálneho rezu.

Začneme tým, že predstavíme problém maximálneho toku. Keďže predpokladáme, že čitateľ sa s ním už stretol, odpustíme si tentokrát motivačné rozprávky a zopakujeme len definíciu. Majme daný (neorientovaný) graf $G = (V, E)$ s nezápornými váhami hrán, t.j. funkciou $c : E \mapsto \mathbb{R}^+$; namiesto $c(\{u, v\})$ budeme skrátene písať $c_{uv} = c_{vu}$. V grafe sú dva význačné vrcholy s a t . V probléme MAX-FLOW je cieľom nájsť čo najväčší tok z s do t . Tok je funkcia $f : V^2 \mapsto \mathbb{R}^+$, ktorej interpretácia je taká, že $f(u, v)$ je množstvo kvapaliny, ktorá tečie z u do v . Funkcia f musí spĺňať tieto vlastnosti:

1. ak $f(u, v) \neq 0$, potom $(u, v) \in E$, t.j. tieť musí po hranách grafu
2. $f(u, v) = -f(v, u)$, t.j. znamienko udáva, ktorým smerom tok ide
3. pre každé $v \notin \{s, t\}$ platí $\sum_{u \in V} f(u, v) = 0$, t.j. čo do vrchola vteká, to z neho aj vytečie

Veľkosť toku je $\sum_{v \in V} f(s, v)$. Rez v grafe G je množina hrán, ktorej odobratie oddelí vrcholy s a t . Cena rezu je súčet váh prerezaných hrán. Problém MIN-CUT je nájsť rez minimálnej ceny. Na nasledovnom obrázku je graf s kapacitami hrán a minimálnym rezom (vľavo) a maximálnym tokom (vpravo; tmavo sú označené hrany, ktorých kapacita je naplnená tokom) s hodnotou 24.



Problém MAX-FLOW sa dá prirodzene formulovať ako úloha lineárneho programovania, a to hneď niekoľkými spôsobmi. Z dôvodov, ktoré, ako dúfame, budú jasné na konci tejto časti, si zvolíme takúto formuláciu: pre každú hranu $(u, v) \in E$ budeme mať dve nezáporné premenné x_{uv} a x_{vu} , ktoré budú udávať množstvo toku z u do v (všimnite si, že sme tým povolili situáciu, že po tej istej hrane tečie tok oboma smermi: x_{uv} v smere z u do v a x_{vu} v smere z v do u ; ničmenej, tieto dve hodnoty stačí odčítať a dostaneme riešenie v zmysle našej definície). Ďalej budeme mať jednu premennú f , ktorá bude udávať veľkosť toku (a tú sa budeme snažiť maximalizovať):

$$f = \sum_{u:(s,u) \in E} x_{su} - \sum_{u:(s,u) \in E} x_{us}.$$
 Zákony zachovania toku aj obmedzenia kapacít ľahko zapíšeme pomocou lineárnych nerovností a dostaneme program

maximalizovať

f

$$\begin{aligned}
\text{pri obmedzeniach} \quad & \sum_{u:(s,u) \in E} x_{su} - \sum_{u:(s,u) \in E} x_{us} - f = 0 \\
& \sum_{u:(t,u) \in E} x_{ut} - \sum_{u:(s,u) \in E} x_{tu} + f = 0 \\
& \sum_{u:(u,v) \in E} x_{vu} - \sum_{u:(u,v) \in E} x_{uv} = 0 \quad \forall v \in V - \{s, t\} \\
& x_{uv} \leq c_{uv} \quad \forall (u, v) \in E \\
& x_{vu} \leq c_{uv} \quad \forall (u, v) \in E \\
& x_{uv} \geq 0 \quad \forall (u, v) \in E
\end{aligned} \tag{52}$$

Prvé a druhé obmedzenie definujú veľkosť toku f ako to, čo vyteká z s , resp. čo vteká do t . Vo všetkých ostatných vrchoch platí zákon zachovania toku.

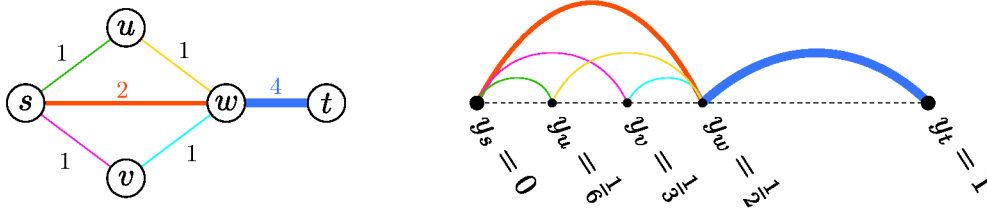
Použijme teraz náš dualizačný recept a zostrojme k programu (52) duálny minimalizačný program. Každému obmedzeniu v primárnom programe zodpovedá premenná v duálnom programe. V programe (52) máme dva druhy obmedzení: obmedzenia v tvare $\dots = 0$ pre každý vrchol a obmedzenia $x_{uv} \leq c_{uv}$. Zavedieme si preto duálne premenné $y_v \in \mathbb{R}$ pre každý vrchol $v \in V$ (y_s zodpovedá prvému obmedzeniu, y_t druhému a ostatné y_v v poradí ďalším) a premenné z_{uv} a z_{vu} pre každú hranu $(u, v) \in E$, pričom $z_{uv}, z_{vu} \geq 0$. Duálny program sa vyrába tak, že i -ta duálna premenná je multiplikátor, ktorým prenášobíme i -te obmedzenie a výsledky sčítame; súčet pravých strán je príslušný odhad a vyžadujeme, aby súčet ľavých strán bol v každej zložke väčší ako maximalizovaná funkcia. Súčet pravých strán je v našom prípade $\sum_{(u,v) \in E} (z_{uv} + z_{vu})c_{uv}$. Maximalizovaná funkcia

má pri všetkých premenných okrem f hodnotu 0. Premenná f vystupuje iba v prvom a druhom obmedzení (52), preto dostávame v duálnom programe obmedzenie $y_t - y_s = 1$ (je jasné, že tok je nezáporný, ale formálne sme v (52) nevyžadovali, aby $f \geq 0$). Každá premenná x_{uv} sa vyskytuje v dvoch obmedzeniach: v obmedzení zodpovedajúcom vrcholu u s kladným znamienkom, v obmedzení zodpovedajúcom vrcholu v so záporným znamienkom a v obmedzení $x_{uv} \leq c_{uv}$. Dostávame tak duálny program

$$\begin{aligned}
\text{minimalizovať} \quad & \sum_{(u,v) \in E} (z_{uv} + z_{vu}) c_{uv} \\
\text{pri obmedzeniach} \quad & y_t - y_s = 1 \\
& y_v - y_u + z_{uv} \geq 0 \quad \forall (u, v) \in E \\
& y_u - y_v + z_{vu} \geq 0 \quad \forall (u, v) \in E \\
& z_{uv}, z_{vu} \geq 0 \quad \forall (u, v) \in E
\end{aligned} \tag{53}$$

Z vety o dualite vieme, že programy (52) a (53) majú rovnakú hodnotu optima. Ako môžeme program (53) interpretovať? Bez ujmy na všeobecnosti môžeme predpokladať, že aspoň jedna z dvojice premenných z_{uv}, z_{vu} je nulová: jediné obmedzenia na z_{uv} a z_{vu} sú $z_{uv} \geq y_u - y_v$ a $z_{vu} \geq y_v - y_u$. Zjavne aspoň jedna z hodnôt $y_u - y_v$ a $y_v - y_u$ nie je kladná, a teda ak máme ľubovoľné prípustné riešenie, tak aspoň jednu z premenných z_{uv}, z_{vu} môžeme nastaviť na 0 a nezväčšíme hodnotu minimalizovanej funkcie. Môžeme si preto označiť $\bar{z}_{uv} = \max\{z_{uv}, z_{vu}\}$; obmedzenia potom hovoria, že $\bar{z}_{uv} \geq |y_u - y_v|$ a z rovnakých dôvodov ako pred chvíľou môžeme bez ujmy na všeobecnosti predpokladať, že $\bar{z}_{uv} = |y_u - y_v|$. Môžeme si predstaviť, že vrcholy grafu ukladáme na priamku: vrchol v je uložený v bode y_v , pričom, opäť bez ujmy na všeobecnosti, je s uložený v bode 0 a t v bode 1. \bar{z}_{uv} je dĺžka hrany (u, v) v našom uložení. Keď to zhrnieme, optimálne riešenie programu

(53) nám dá také uloženie vrcholov grafu na úsečku dĺžky 1, že s a t sú na krajoch a minimalizuje sa váhovaná dĺžka hrán.



Graf s váhami hrán a jedno z možných optimálnych riešení programu (53): $\bar{z}_{su} = \bar{z}_{vw} = \frac{1}{6}$, $\bar{z}_{sv} = \bar{z}_{uw} = \frac{1}{3}$, $\bar{z}_{sw} = \bar{z}_{wt} = \frac{1}{2}$. Výsledná hodnota je

$$\sum_{(u,v) \in E} c_{uv} \bar{z}_{uv} = \frac{1}{6} + \frac{1}{3} + 2\frac{1}{2} + \frac{1}{3} + \frac{1}{6} + 4\frac{1}{2} = 4.$$

Podme teraz prepísať program (53) do normálneho tvaru $\max\{-\mathbf{c}^T \boldsymbol{\beta} \mid A\boldsymbol{\beta} = 0, \boldsymbol{\beta} \geq 0\}$. Ku každej premennej z_{uv} zavedieme premennú $\hat{z}_{uv} \geq 0$, aby sme získali obmedzenia tvaru $y_v - y_u + z_{uv} - \hat{z}_{uv} = 0$. Ak vektor $\boldsymbol{\beta}$ pozostáva zaradom z hodnôt $y_s, y_t, y_{v_1}, \dots, z_{uv}, z_{vu}, \dots, \hat{z}_{uv}, \hat{z}_{vu}, \dots$, matica A má takúto štruktúru:

$$A = \begin{array}{c|c|c|c} & \overbrace{y_s \ y_t \ \dots \ y_u \ \dots \ y_v \ \dots} & \overbrace{z_{uv} \ z_{vu} \ \dots} & \overbrace{\hat{z}_{uv} \ \hat{z}_{vu} \ \dots} \\ \hline & -1 \ 1 & & \\ \hline & & 1 & -1 \\ \hline & -1 \ 1 & 1 & -1 \\ & 1 \ -1 & 1 & -1 \\ \hline & & & 1 & -1 \end{array} \quad \left. \vphantom{\begin{array}{c|c|c|c} & \overbrace{y_s \ y_t \ \dots \ y_u \ \dots \ y_v \ \dots} & \overbrace{z_{uv} \ z_{vu} \ \dots} & \overbrace{\hat{z}_{uv} \ \hat{z}_{vu} \ \dots} \end{array}} \right\} (u,v) \in E$$

Nasledujúcu priamočiariu povinnú jazdu prenecháme na čitateľa:

Cvičenie. S pomocou viet 2.6 a 2.8 ukážte, že matica A je TUM.

Z Vety 2.5 vyplýva, že existuje celočíselné optimálne riešenie programu (53). To znamená, že všetky vrcholy majú $y_v \in \{0, 1\}$ (t.j. sú uložené na niektorom konci úsečky) a každá hrana má dĺžku 0 alebo 1. Preto každá cesta z s do t musí obsahovať aspoň jednu hranu s dĺžkou 1, a teda ak hrany dĺžky 1 z grafu odstránime, dostaneme rez, ktorý oddeľuje s od t . Môžeme teda povedať

Veta 3.7 (MAX-FLOW-MIN-CUT veta). *Veľkosť maximálneho toku sa rovná kapacite minimálneho rezu.*

Dôkaz: Veľkosť maximálneho toku je optimum programu (52), ktoré je rovnaké, ako optimum programu (53). Existuje celočíselné optimum programu (53), a zároveň je bijekcia medzi $s - t$ rezmí a celočíselnými riešeniami (53): celočíselnému riešeniu (53) prislúcha rez a každému rezu vieme prirodzeným spôsobom priradiť riešenie (53). \square

Je dosť možné, že čitateľ možno pozná oveľa jednoduchší dôkaz tohto tvrdenia. Dôvod, prečo uvádzame tento dôkaz je (okrem toho, že chceme ilustrovať dualitu lineárnych programov) ten, že ukazuje dvojicu MAX-FLOW-MIN-CUT ako špeciálny prípad všeobecnejších problémov; zároveň dáva nový pohľad na otázku, prečo rovnaký výsledok neplatí pre iné podobné problémy (ak

napríklad máme viacero zdrojov a ústí a cieľom je maximalizovať tok rôznych komodít v spoločných potrubíach, máme duálny problém k problému MIN-MULTI-CUT z Definície 2.14, avšak analogický výsledok neplatí) cez unimodularitu príslušných matíc.

3.3 Primárno-duálna metóda

Dúfame, že sme čitateľa presvedčili o tom, že dualita je zaujímavá vlastnosť lineárnych programov. Teraz je načas ukázať, ako sa dá využiť pri návrhu algoritmov. Pozrime sa na dvojicu duálnych programov

$$(P) : \min_{\mathbf{x} \in \mathbb{R}^n} \{ \mathbf{c}^\top \mathbf{x} \mid A\mathbf{x} \geq \mathbf{b}, \mathbf{x} \geq 0 \}$$

$$(D) : \max_{\mathbf{y} \in \mathbb{R}^m} \{ \mathbf{b}^\top \mathbf{y} \mid A^\top \mathbf{y} \leq \mathbf{c}, \mathbf{y} \geq 0 \}$$

Z vety o dualite vieme, že majú rovnakú hodnotu optima, t.j. že existujú vektory $\mathbf{x}^* \geq 0$ a $\mathbf{y}^* \geq 0$, že $A\mathbf{x}^* \geq \mathbf{b}$, $A^\top \mathbf{y}^* \leq \mathbf{c}$ a $\mathbf{c}^\top \mathbf{x}^* = \mathbf{b}^\top \mathbf{y}^*$. Pripomeňme si nerovnosti z dôkazu Vety 3.1, ktoré platia pre všetky dvojice prípustných riešení primárnej a duálnej úlohy, a teda aj pre \mathbf{x}^* , \mathbf{y}^* :

$$\mathbf{c}^\top \mathbf{x}^* \stackrel{(\clubsuit)}{\geq} (A^\top \mathbf{y}^*)^\top \mathbf{x}^* = \mathbf{y}^{*\top} A\mathbf{x}^* \stackrel{(\diamond)}{\geq} \mathbf{y}^{*\top} \mathbf{b} \quad (54)$$

Keďže $\mathbf{c}^\top \mathbf{x}^* = \mathbf{b}^\top \mathbf{y}^*$, (\clubsuit) aj (\diamond) musia byť rovnosti. Pozrime sa na (\clubsuit) a rozpíšme vektorový zápis pomocou sumy:

$$\sum_{j=1}^n c_j x_j^* = \sum_{j=1}^n [A^\top \mathbf{y}^*]_j x_j^*, \quad (55)$$

kde symbol $[\cdot]_j$ označuje j -ty prvok vektora. Z prípustnosti duálneho riešenia vieme, že $\mathbf{c} \geq A^\top \mathbf{y}^*$; nerovnosť vektorov platí v každej zložke, takže pre každé j je $c_j \geq [A^\top \mathbf{y}^*]_j$. Pre každé j je $x_j^* \geq 0$, a teda aj $c_j x_j^* \geq [A^\top \mathbf{y}^*]_j x_j^*$. Z toho vidieť, že ak má platiť rovnosť (55), musí platiť rovnosť v každej zložke.

Rovnaké úvahy sa dajú urobiť pre nerovnosť (\diamond) a dostaneme tak nasledovnú charakterizáciu optimálneho riešenia:

Veta 3.8 (podmienky komplementarity). *Nech \mathbf{x} , \mathbf{y} sú prípustné riešenia primárnej a duálnej úlohy. Potom \mathbf{x} , \mathbf{y} sú obidve optimálne vtedy a len vtedy, ak sú splnené obe nasledujúce podmienky:*

- *primárna podmienka komplementarity:*

$$\forall 1 \leq j \leq n : \text{ buď } x_j = 0 \text{ alebo } \sum_{i=1}^m a_{ij} y_i = c_j$$

- *duálna podmienka komplementarity:*

$$\forall 1 \leq i \leq m : \text{ buď } y_i = 0 \text{ alebo } \sum_{j=1}^n a_{ij} x_j = b_i$$

Edmondsov algoritmus pre MIN-1-FACTOR

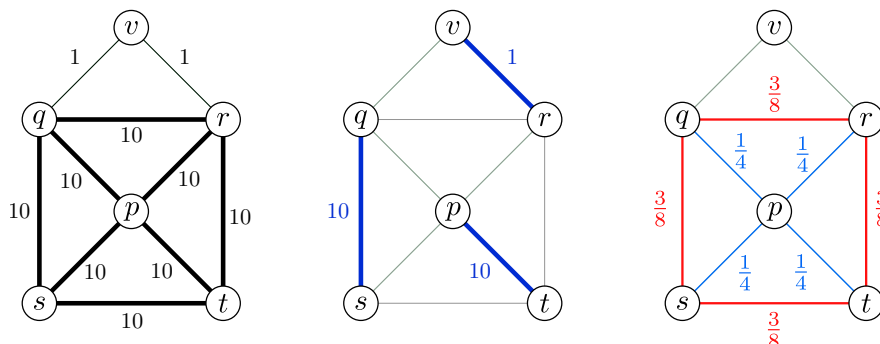
Podmienky komplementarity sú užitočný nástroj pri analýze algoritmov, lebo poskytujú jednoduchý invariant, ktorý charakterizuje optimum. Pripomeňme si definíciu problému MAX-WEIGHTED-BIPARTITE-MATCHING z časti o ILP:

Definícia 2.3. Majme daný bipartitný graf s hranami ohodnotenými nezápornými reálnymi číslami. Cieľom problému MAX-WEIGHTED-BIPARTITE-MATCHING je nájsť množinu hrán s najväčším súčtom váh tak, aby žiadne dve vybrané hrany nezdíeľali vrchol.

Problém sme vtedy formulovali ako celočíselný program

$$\begin{aligned} & \text{maximalizovať} && \sum_{e \in E} \omega_e x_e \\ & \text{pri obmedzeniach} && \sum_{\substack{e \in E \\ e=(v,w)}} x_e \leq 1 && \forall v \in V \\ & && x_e \geq 0 && \forall e \in E \\ & && x_e \in \mathbb{Z} \end{aligned}$$

kde $\omega \in \mathbb{R}^n$ je vektor váh hrán. Ukázali sme, že matica obmedzení je TUM, a preto stačí vyriešiť relaxovaný program a podmienky celočíselnosti máme zadarmo. Celkom prirodzene sa natíska otázka: "A čo keby ten graf nebol bipartitný?" Formulácia ILP bude stále v poriadku, ale už nebude platiť, že musí existovať optimálne celočíselné riešenie.



Vľavo je graf s váhami hrán. Maximálne párovanie má hodnotu 21: hrany váhy 10 sú iba medzi vrcholmi $\{p, q, r, s, t\}$, t.j. môžu byť v riešení najviac dve. Zo zvyšných hrán môže byť najviac 1. Vpravo riešenie relaxovaného programu s hodnotou 25.

Prv, než budeme pokračovať, upravíme trochu formuláciu nášho problému. Párovaniu, ktoré pokrýva všetky hrany, t.j. takej množine hrán $E' \subseteq E$, že každý vrchol susedí s práve jednou hranou z E' , budeme hovoriť *perfektné párovanie*, respektíve *1-faktor* (pochopiteľne, aby graf mal 1-faktor, musí mať párny počet vrcholov). Namiesto hľadania najťažšieho párovania nám stačí vedieť hľadať najťažší 1-faktor: z grafu G vyrobíme G' tak, že ak má G nepárny počet vrcholov, pridáme k nemu jeden izolovaný vrchol a potom všetky vrcholy, ktoré nie sú spojené hranou spojíme hranou váhy 0. Čitateľ sa ľahko presvedčí, že párovania v G zodpovedajú 1-faktorom v G' .

Ďalej nech ω_{\max} je maximálna váha hrany v G' . Ak vyrobíme G'' tak, že nastavíme nové váhy $\omega''_e = \omega_{\max} - \omega_e$, tak vidno, že najľahší 1-faktor v G'' je najťažší 1-faktor v G' . Dostali sme sa tak k nasledovnej definícii:

Definícia 3.10. Majme daný úplný ohodnotený graf $G = (V, E)$ na $n = 2k$ vrchoch, pričom hrana $e = (u, v) \in E$ má váhu $\omega_e \in \mathbb{R}^+$. Problém MIN-1-FACTOR je nájsť 1-faktor v G , pre ktorý je súčet váh hrán minimálny, t.j. $\min_{E'} \sum_{e \in E'} \omega(e)$, kde minimum je brané cez všetky 1-faktory E' .

Čitateľ, ktorý s nami vydržal až potiaľto, určite nebude mať ťažkosti zapísať MIN-1-FACTOR ako celočíselný lineárny program: pre každú hranu $e \in E$ zavedieme premennú $x_e \in \{0, 1\}$, ktorá vyjadruje, či je hrana e vybratá do párovania alebo nie. Vybratá množina hrán je 1-faktor práve vtedy, ak s každým vrcholom susedí práve jedna vybratá hrana, čo priamočiaro zapíšeme

$$\begin{aligned} & \text{minimalizovať} && \sum_{e \in E} \omega_e x_e \\ \text{pri obmedzeniach} && \sum_{\substack{e \in E \\ e = (u, v)}} x_e &= 1 && \forall v \in V \\ && x_e &\in \{0, 1\} && \forall e \in E \end{aligned} \quad (56)$$

Ak tento program relaxujeme a namiesto $x_e \in \{0, 1\}$ budeme požadovať iba $x_e \geq 0$ (čitateľ si všimne, že $x_e \leq 1$ vyplýva z minimality), optimum už nemusí byť celočíselné. V nasledujúcich odstavcoch predstavíme primárno-duálny prístup podľa Edmondsa [?]. Najprv ale ešte jedno označenie, ktoré nám zjednoduší zápis:

Definícia 3.11 (hranová hranica množiny). Majme graf $G = (V, E)$ a množinu vrcholov $S \subseteq V$. Hranová hranica množiny S , označovaná $\delta(S)$, je množina hrán s jedným koncom v S a druhým mimo S , t.j.:

$$\delta(S) := \{e \in E \mid e = (u, v), u \in S, v \in V \setminus S\}$$

Ako sa vysporiadať s tým, že relaxácia programu (56) nemá celočíselné riešenie? Trik je v tom, že pridáme (veľa) dodatočných obmedzení, ktoré zabezpečia celočíselnosť optima. Nech \mathcal{S} sú všetky aspoň trojprvkové množiny s nepárnym počtom vrcholov, t.j.

$$\mathcal{S} := \{S \subseteq V \mid |S| > 1, |S| \text{ nepárne} \}$$

Každá hrana má dva konce, a preto vrcholy z $S \in \mathcal{S}$ nemôžu byť popárované medzi sebou, takže v každom 1-faktore musí aspoň 1 hrana odchádzať z S . Pridáme tieto obmedzenia k relaxovanému programu (56), čím dostaneme

$$\begin{aligned} & \text{minimalizovať} && \sum_{e \in E} \omega_e x_e \\ \text{pri obmedzeniach} && \sum_{e \in \delta(\{v\})} x_e &= 1 && \forall v \in V \\ && \sum_{e \in \delta(S)} x_e &\geq 1 && \forall S \in \mathcal{S} \\ && x_e &\geq 0 && \forall e \in E \end{aligned} \quad (57)$$

... a *voilà!* Máme lineárny program, ktorý má celočíselné optimálne riešenie. Dostali sme sa však z dažďa pod odkvap: po prvé potrebujeme dokázať, že naozaj existuje celočíselné optimum programu (57) a podruhé potrebujeme riešiť program, ktorý má exponenciálne veľa obmedzení. Ani jeden z týchto problémov nie je neprekonateľný, ale my to spravíme elegantne: využijeme dualitu a vyhneme sa riešeniu (57); a navyše dôkaz celočíselnosti vypadne zadarmo. Spôsob návrhu algoritmov, ktorý tu ukážeme, sa zvykne nazývať *primárno-duálna metóda*.

Použijeme náš dualizačný recept a zostrojíme duálny program k (57). Duálny program bude maximalizačný a bude mať premennú pre každé obmedzenie. Máme dva typy obmedzení: pre vrcholy a pre množiny, takže si zavedieme duálne premenné $r_v \in \mathbb{R}$ pre $v \in V$ a premenné $w_S \in \mathbb{R}^+$ pre $S \in \mathcal{S}$. Každá primárna premenná x_e má vklad $x_e \omega_e$ do minimalizačnej funkcie a vyskytuje sa v dvoch obmedzeniach pre vrcholy (konkrétne pre dva koncové vrcholy e) a v obmedzeniach pre tie množiny $S \in \mathcal{S}$, kde $e \in \delta(S)$. Dostali sme duálny program (všimnite si, že r_v môže byť aj záporné):

$$\begin{aligned}
& \text{maximalizovať} \quad \sum_{v \in V} r_v + \sum_{S \in \mathcal{S}} w_S \\
& \text{pri obmedzeniach} \quad r_u + r_v + \sum_{\substack{S \in \mathcal{S} \\ e \in \delta(S)}} w_S \leq \omega_e \quad \forall e = (u, v) \in E \\
& \quad \quad \quad w_S \geq 0 \quad \forall S \in \mathcal{S}
\end{aligned} \tag{58}$$

Z didaktických dôvodov, a aj preto, že v našej dvojici programov nie sú všetky premenné nezáporné, prepíšme nerovnosť (54) v našom značení:

$$\sum_{e \in E} x_e \omega_e \stackrel{\clubsuit}{\geq} \sum_{\substack{e \in E \\ e = (u, v)}} x_e (r_u + r_v + \sum_{\substack{S \in \mathcal{S} \\ e \in \delta(S)}} w_S) \stackrel{\heartsuit}{=} \sum_{v \in V} (r_v \sum_{e \in \delta(\{v\})} x_e) + \sum_{S \in \mathcal{S}} w_S \left(\sum_{e \in \delta(S)} x_e \right) \stackrel{\diamondsuit}{\geq} \sum_{v \in V} r_v + \sum_{S \in \mathcal{S}} w_S$$

Rovnosť (\heartsuit) platí preto, lebo na ľavej strane je pre každý vrchol $v \in V$ hodnota r_v zarátaná s koeficientom x_e pre všetky hrany incidentné s v a hodnota w_S je zarátaná s koeficientom x_e pre všetky hrany z hranice S . Z obmedzení programu (57) vyplýva, že modrá suma $\sum_{e \in \delta(\{v\})} x_e = 1$ a červená suma $\sum_{e \in \delta(S)} x_e \geq 1$, preto podmienky komplementarity majú tvar

$$\mathbf{S1}(\clubsuit) \quad \forall e = (u, v) \in E : \quad x_e > 0 \Rightarrow r_u + r_v + \sum_{\substack{S \in \mathcal{S} \\ e \in \delta(S)}} w_S = \omega(e)$$

$$\mathbf{S2}(\diamondsuit) \quad \forall S \in \mathcal{S} : \quad w_S > 0 \Rightarrow \sum_{e \in \delta(S)} x_e = 1$$

Pozrime sa teraz na program (58) a skúsme mu dať nejakú intuitívnu interpretáciu. Predstavme si, že okolo každého vrchola $v \in V$ (resp. okolo každej množiny $S \in \mathcal{S}$) môže byť bublina s nábojom r_v (resp. w_S). Samozrejme, množiny s \mathcal{S} sa môžu rôzne prekrývať, a tak predstava bublín okolo každej z nich nie je úplne intuitívna, ničmenej nakoniec budeme využívať iba systémy do seba zapadajúcich bublín. Program (58) nám hovorí, že chceme maximalizovať celkový náboj. Váhy hrán si teraz vieme predstaviť ako kapacitu a obmedzenia hovoria, že žiadnu hranu e nesmieme "pretrhnúť": celkový náboj na všetkých bublinách, ktoré pretínajú e nesmie prekročiť jej kapacitu. Hranu e , pre ktorú platí $r_u + r_v + \sum_{\substack{S \in \mathcal{S} \\ e \in \delta(S)}} w_S = \omega(e)$, nazveme *plná*. Podmienky **S1** a **S2** spolu s vetou o dualite vieme interpretovať takto:

Lema 3.12. *Ak (ľubovoľným spôsobom) nájdeme 1-faktor M a hodnoty bublín \mathbf{r} a \mathbf{w} tak, že*

(I1) *žiadna hrana nie je preplnená,*

(I2) *hodnoty $w_S \geq 0$,*

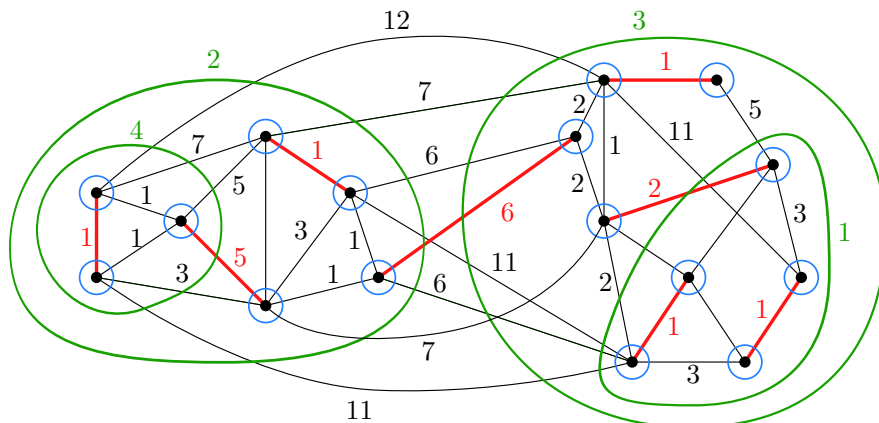
(I3) *všetky hrany z M sú plné a*

(I4) *z každej nenulovej bubliny $S \in \mathcal{S}$ odchádza práve jedna hrana z M ,*

tak máme optimálne riešenie dvojice programov (57) a (58), ktoré má celočíselné hodnoty \mathbf{x} , a teda tvorí minimálny 1-faktor.

Dôkaz:. Podmienky (I1) a (I2) zaručujú prípustnosť duálneho programu (58). Fakt, že M je 1-faktor zaručuje prípustnosť primárneho programu (57). Podmienky (I3) a (I4) sú v poradí podmienky **S1** a **S2**, takže tvrdenie je dôsledok Vety 3.8. \square

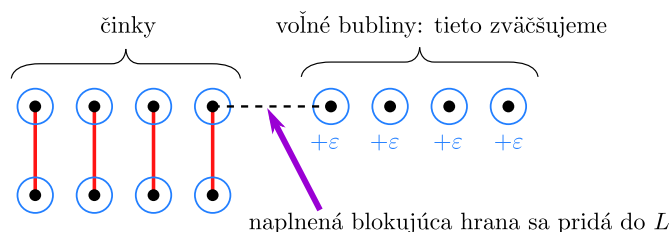
Od tohoto momentu môžeme zabudnúť na celé lineárne programovanie a ideme sa snažiť nájsť algoritmus, ktorý vyrobí hľadané objekty M , \mathbf{r} a \mathbf{w} . Ako sme už naznačili, keďže \mathbf{w} je príliš veľké, budeme si pamätať iba jeho nenulové zložky; zároveň budeme hľadať iba také riešenia, v ktorých sa množiny s nenulovými bublinami nepretínajú. Chceli by sme dostať nejakú takúto štruktúru:



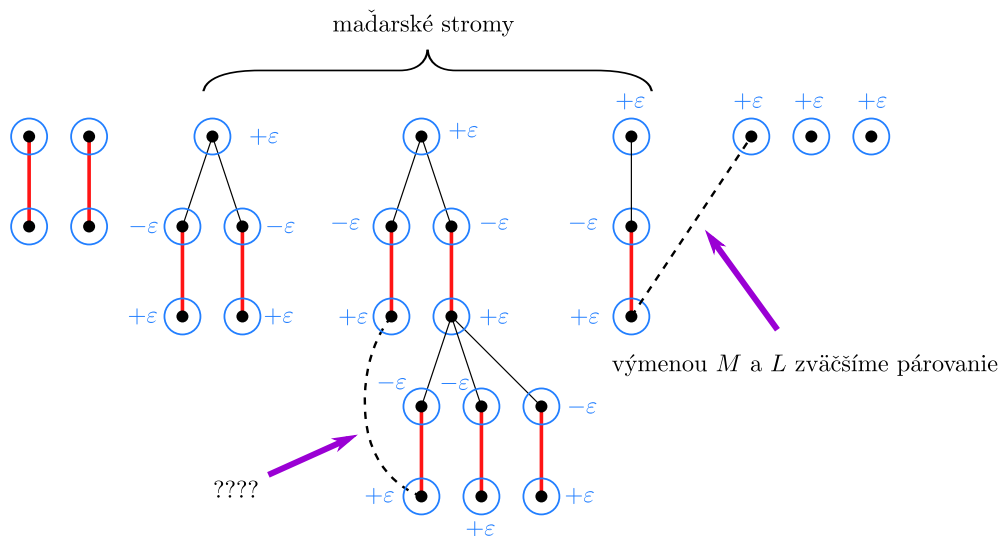
Modré bubliny majú všetky hodnotu $1/2$, hrany, ktoré nie sú zakreslené, majú váhu ∞ . Červené hrany tvoria 1-faktor. Žiadna hrana nie je preplnená, všetky červené hrany sú plné a z každej zelenej bubliny odchádza práve jedna červená hrana; preto vieme, že červený 1-faktor je minimálny.

Samozrejme, nikde zatiaľ nemáme zaručené, že takáto konfigurácia vždy existuje. Ale ak to dokážeme a nájdeme algoritmus, ktorý ju vyrobí, môžeme zajašať a úlohu označiť za splnenú.

Začnime neformálnym opisom činnosti algoritmu po štarte. Počas celého algoritmu budú podmienky **(I1)**, **(I2)** a **(I3)** udržiavané v platnosti. Algoritmus začne s tým, že párovanie M bude prázdne a všetky bubliny budú nulové. Postupne sa bude snažiť pridávať náboj do bublín a hrany do párovania tak, aby nakoniec M bol 1-faktor a boli splnená podmienka **(I4)**; potom M bude minimálny 1-faktor. V prvom kroku začne pridávať náboj na všetky bubliny r_v (v ďalšom budeme tieto bubliny volať *modré*, kým bubliny w_s budú *zelené*). Časom sa stane, že dve modré bubliny r_u a r_v naplnia nejakú hranu $e = (u, v)$. Hrana e sa pridá do M a bublinám r_u a r_v sa nebude v ďalšom kroku pridávať náboj (budú tvoriť *činku*). Raz sa ale naplní aj nejaká hrana e , ktorej jeden vrchol už patrí hrane z M . Táto hrana je plná, a preto jej koncové bubliny sa nemôžu zväčšiť, a zároveň sa nemôže pridať do M . Nazveme ju *blokujúca hrana* a algoritmus si bude udržiavať množinu blokujúcich hrán L .

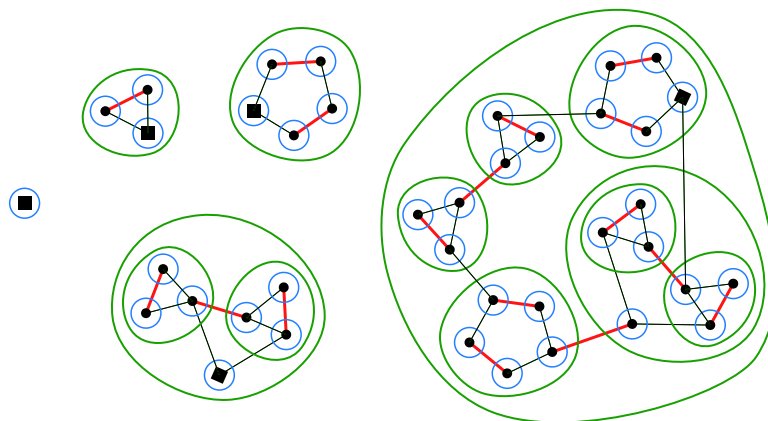


Okrem voľných bublín a čínok vznikol ďalší útvar: cesta dĺžky 3 zložená z hrany z L a hrany z M ; cesty, na ktorých sa striedajú hrany z L a M budeme volať *alternujúce*. Keď sa v ďalšom bude voľným bublinám pridávať náboj $+\varepsilon$, bublinám z alternujúcej cesty sa bude striedavo pridávať $+\varepsilon$ a $-\varepsilon$, aby sa žiadna hrana nepreplnila (pripomíname, že na rozdiel od zelených bublín, modré bubliny môžu mať záporný náboj). V ďalšom sa naplní hrana medzi dvoma bublinami, ktorým sa náboj pridáva, a môžu tak vznikať stromy z alternujúcich ciest (tzv. *maďarské stromy*). Ak naplnenou hranou vznikne alternujúca cesta s nepárnym počtom hrán, je to dobré, lebo sa môže zväčšiť párovanie tak, že na tejto ceste sa vymení príslušnosť hrán medzi M a L a z alternujúcej cesty ostane sada čínok.



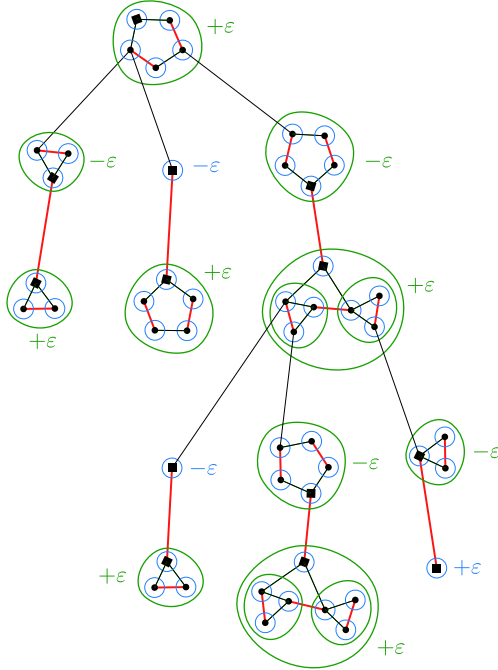
Čo ale máme robiť, ak sa napríklad naplní hrana medzi dvoma bublinami v jednom strome? Toto je moment, v ktorom dozrel čas na zelené bubliny a ich hierarchické štruktúry.

Základnou dátovou štruktúrou algoritmu je *kvet*. Každý kvet má vonkajšiu bublinu a v nej jeden význačný vrchol zvaný *stopka*. Najjednoduchší kvet je jeden vrchol s modrou bublinou okolo neho. Zložitejšie kvety vznikajú rekurzívne: majme nepárny počet kvetov $K_1, K_2, \dots, K_{2r+1}$, $r \geq 1$ (t.j. aspoň tri kvety) tak, že stopky kvetov K_{2i} a K_{2i+1} pre $i = 1, \dots, r$ sú spojené hranou z M , a zároveň pre každú dvojicu kvetov $A := K_{2i-1}$ a $B := K_{2i}$ pre $i = 1, \dots, r$ a $A := K_{2r+1}$ a $B := K_1$ existujú vrcholy $u \in A$ a $v \in B$ také, že hrana $(u, v) \in L$. Potom bublina ohraničujúca K_1, \dots, K_{2r+1} vytvorí nový kvet, ktorého stopka bude stopka kvetu K_1 . Kvety, ktoré nie sú súčasťou iného kvetu, budeme volať *vonkajšie kvety*.



Päť rôznych vonkajších kvetov so stopkami označenými štvorcami. Červené hrany sú z M , čierne z L .

Za povšimnutie stojí, že kvet uzatvára časť grafu, ktorú máme "takmer hotovú": okrem stopky sú všetky vrcholy kvetu pospájané hranami z M a zároveň každú bublinu pretína práve jedna hrana z M , s výnimkou bublín, ktoré ohraničujú stopku (toto je pre činnosť algoritmu kľúčové pozorovanie a čitateľovi odporúčame si ho detailne indukciou dokázať). Ak je medzi stopkami dvoch kvetov plná hrana, jej pridaním do M vznikne činka. Algoritmus skončí, ak sú všetky vrcholy pokryté činkami.

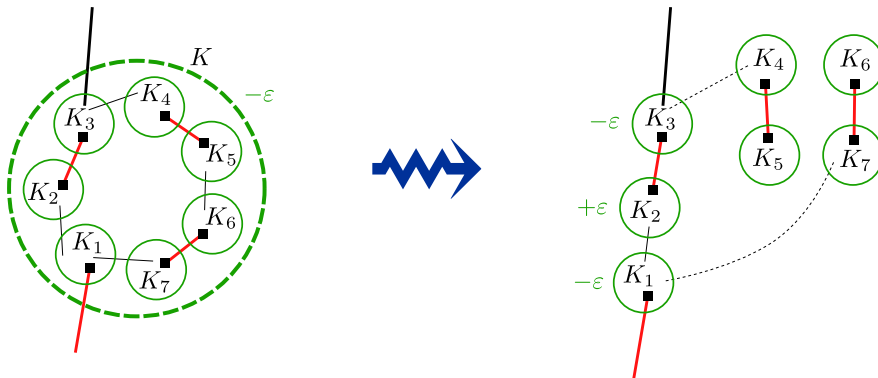


Operácia *presun* na maďarskom strome.

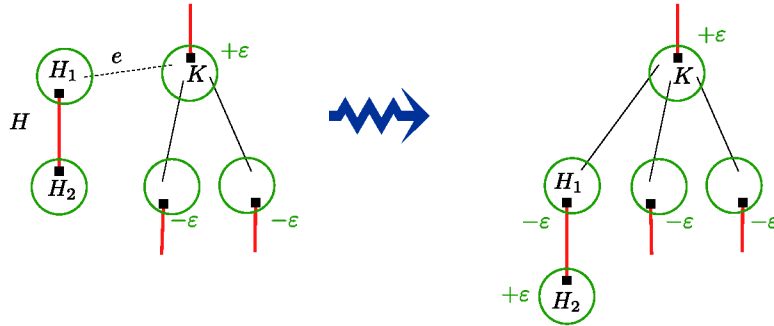
Voľné kvety, ktoré netvorí činky, sú organizované v maďarských stromoch. Kvet na úrovni 0 je koreň, stopka kvetu na úrovni $2h-1$ je spojená hranou z M so stopkou syna na úrovni $2h$. Ak má kvet K na úrovni $2h$ syna H , tak existuje hrana z L medzi nejakým vrcholom z K a nejakým vrcholom z H . Intuitívna predstava je takáto: koreň stromu je kvet K , ktorý by sme chceli zapojiť do činky. Lenže nemôžeme, lebo hrany, ktoré z neho odchádzajú a dali by sa použiť, nie sú plné. Chceli by sme pridať náboj na vonkajšiu bublinu K , ale nemôžeme, lebo nám v tom bránia hrany z L , ktoré vedú do jeho synov.

Algoritmus si v každom momente udržiava sadu maďarských stromov, pričom zvyšok grafu je pokrytý činkami. Algoritmus pracuje v iteráciách, pričom v každej iterácii sa urobí operácia *presun*: všetkým voľným kvetom na párnych úrovniach stromov sa k vonkajšej bubline pripočíta ε a od kvetov na nepárnych úrovniach sa ε odpočíta. Hodnota ε sa zvolí maximálna taká, kým sa nenaruší niektorá z podmienok (I1), (I2). To sa môže stať niekoľkými spôsobmi:

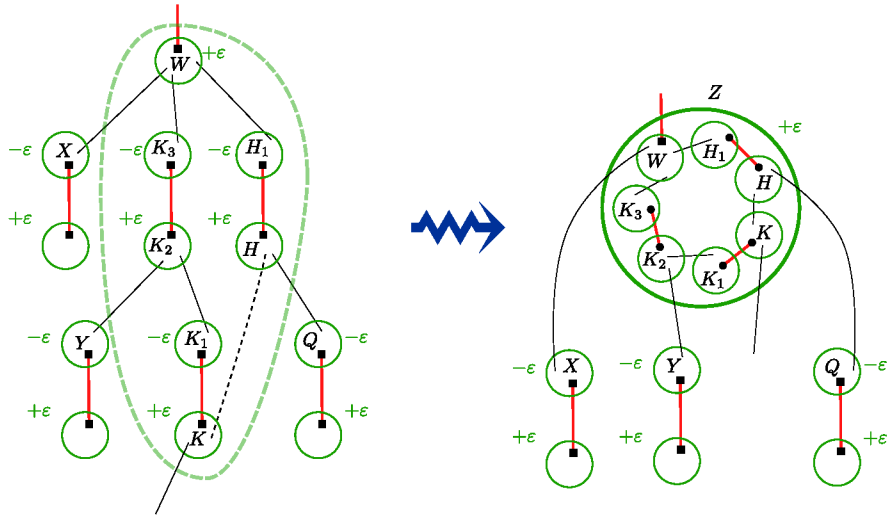
(P1) Zelenej bubline na nepárnej úrovni klesol náboj na 0. Nech K je kvet, ktorému patrí nulová bublina. Z definície kvetu, K obsahuje nepárny počet kvetov K_1, \dots, K_{2r+1} , pričom stopka je vo K_1 . Keďže K je na nepárnej úrovni, má jedného rodiča a jedného syna, ktorý je napojený na stopku. Nech hrana do rodiča ide z kvetu K_t a bez ujmy na všeobecnosti nech t je nepárne. Potom cesta K_1, K_2, \dots, K_t má nepárny počet kvetov a môže nahradiť K v strome. Dvojice vrcholov K_{t+1}, K_{t+2} až K_{2r}, K_{2r+1} tvoria činky. Plné hrany medzi K_t a K_{t+1} a K_1, K_{2r+1} sa dajú odobrať z L , lebo v novom strome bude jedna z nich na nepárnej úrovni a druhá v činke, takže pri najbližšej operácii *posun* prestanú byť plné.



(P2) Naplnila sa hrana e medzi kvetom K na párnej úrovni a činkou H . Nech sa činka H skladá z kvetov H_1 a H_2 tak, že e vedie do nejakého vrchola vo w_1 . Hrana e sa pridá do L a činka H sa pripojí k príslušnému stromu tak, že K (na párnej úrovni) bude mať syna H_1 (na nepárnej úrovni) a ten bude mať jedného syna H_2 (na párnej úrovni).



(P3) Naplnila sa hrana spájajúca kvety K a H v jednom strome. Zjavne K aj H sú na párnej úrovni. Nech W je najbližší spoločný predok K a H . Keďže W má aspoň dvoch synov, musí byť tiež na párnej úrovni. Nech $K, K_1, \dots, K_{2k+1}, W$ a $H, H_1, \dots, H_{2r+1}, W$ sú cesty v strome. Z parity vrcholov vyplýva, že ich môžeme obaliť novou bublinou a dostaneme kvet Z na párnej úrovni, ktorého stopka je stopka W . Synovia Z budú všetci synovia zahrnutých kvetov – títo ostanú na nepárnej úrovni.



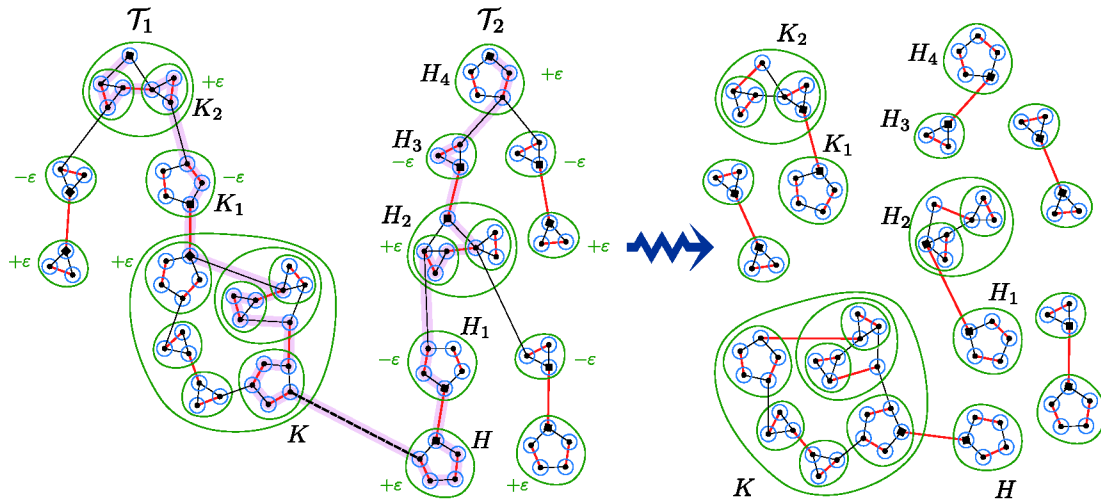
(P4) Naplnila sa hrana e spájajúca kvety K a H v dvoch rôznych stromoch \mathcal{T}_1 a \mathcal{T}_2 . Toto je vlastne jadro celého algoritmu, v ktorom zväčšíme párovanie M . Urobíme to tak, že nájdeme alternujúcu cestu, ktorá spája stopku koreňa stromu \mathcal{T}_1 so stopkou koreňa \mathcal{T}_2 . Keďže sa naplnila hrana, obidva kvety K a H museli byť na párnej úrovni, takže existuje cesta K, K_1, \dots, K_{2r} v strome \mathcal{T}_1 a H, H_1, \dots, H_{2q} v strome \mathcal{T}_2 , kde K_{2r} a H_{2q} sú korene príslušných stromov. Obidve cesty sú tvorené hranami z pôvodného grafu G a striedajú sa v nich hrany z M a L , pričom hrana z M spája stopky susedných kvetov. Aby sme cesty v stromoch mohli doplniť na alternujúce cesty v grafe G , stačí si uvedomiť nasledujúce tvrdenie:

Lema 3.13. *Nech K je kvet, u je jeho stopka a v je jeho ľubovoľný vrchol. Potom existuje alternujúca cesta v G z u do v , ktorá je celá obsiahnutá v K a ak je neprázdna, tak začína hranou z L a končí hranou z M .*

Dôkaz: Indukciou na hĺbku vnorenia kvetu. Pre kvet s jedným vrcholom tvrdenie zrejme platí. Nech teda je kvet K tvorený kvetmi K_1, \dots, K_{2r+1} , pričom stopka K je v K_1 . Bez ujmy na všeobecnosti, nech $v \in K_{2t-1}$ pre nejaké t (keby $v \in K_{2t}$, zmeníme smer číslovania kvetov). Ak $t = 1$, použijeme indukciu na kvet K_1 . Nech teda $t > 1$. Z definície kvetu existuje hrana $(q, w) \in L$,

kde $q \in K_1$ a $w \in K_2$. Z indukcie existuje alternujúca $u - q$ cesta v K_1 , ktorá končí hranou z M . Zároveň existuje alternujúca cesta v K_2 zo stopky do w , ktorá končí hranou z M . Spojením týchto ciest dostaneme alternujúcu cestu z u do stopky K_2 , ktorá končí hranou z L , a teda sa dá predĺžiť do stopky K_3 . Tento postup opakujeme, až kým cestu dostaneme do stopky kvetu K_{2t-1} a odtiaľ opäť z indukčného predpokladu predĺžime cestu do v . \square

S pomocou Lemy 3.13 teraz vieme nájsť alternujúcu cestu medzi stopkou K_{2r} a stopkou H_{2q} . Na tejto ceste vymeníme príslušnosť hrán medzi L a M , čím zvýšime počet hrán v párovaní M . Stromy \mathcal{T}_1 a \mathcal{T}_2 následne "rozoberieme": kvety K_{2i-1} a K_{2i} (a rovnako H_{2j-1} a H_{2j}) budú po výmene hrán na alternujúcej ceste tvoriť činky. Aby sme videli, prečo, stačí si uvedomiť, že cesta z dôkazu Lemy 3.13 pretína každý vnútorný kvet buď ani raz, alebo dvakrát a to jednou hranou z M a jednou z L . Preto po výmene hrán z M a L ostane zachovaná podmienka, že každú bublinu pretína práve jedna hrana z M a stopka stromu sa presunie do vrcholu v z Lemy 3.13. Rovnako vytvoria činku kvety K a H . Zvyšné časti stromov tvoria činky prirodzeným spôsobom.



Celý algoritmus pracuje v cykle, v ktorom kým M nie je 1-faktor, nájde najmenšie ε , ktoré poruší niektorú z podmienok (I1), (I2) Lemy 3.12. Podľa toho, aká situácia nastala, vykoná jednu z akcií (P1), (P2), (P3) alebo (P4) a pokračuje v hlavnom cykle. Podmienka (I3) ostáva splnená stále. Keď algoritmus skončí a M je 1-faktor, žiaden kvet nemôže byť koreň stromu (jeho stopka by bola nespárovaná), preto všetky vrcholy sú pokryté činkami a platí aj podmienka (I4). Preto podľa Lemy 3.12 po skončení algoritmu je M minimálny 1-faktor.

Ostáva nám ukázať, že algoritmus naozaj skončí a, pokiaľ možno aj to, že skončí rýchlo. Kľúčové je pri tom nasledovné pozorovanie:

Lema 3.14. Popísaný algoritmus na riešenie problému MIN-1-FACTOR urobí maximálne $O(n^2)$ iterácií, kde n je počet vrcholov G .

Dôkaz: Počet hrán v M nikdy neklesá a každé vykonanie akcie (P4) ho o jednotku zväčší. Preto sa v celom algoritme vykoná $O(n)$ akcií (P4). Na dôkaz tvrdenia stačí ukázať, že medzi dvoma vykonaniami (P4) sa vykoná najviac $O(n)$ akcií (P1), (P2) a (P3).

Prvá vec, ktorú si všimneme, je, že v ktoromkoľvek okamihu je v celom algoritme $O(n)$ bublín (vrátane vnorených); to vyplýva z toho, že kvet obsahuje aspoň tri vnútorné kvety, a teda ak si nazveme *hlbkou* kvety maximálnu úroveň do seba vnorených bublín, tak kedykoľvek môže byť najviac n kvetov hĺbky 0, $n/3$ kvetov hĺbky 1, $n/3^2$ kvetov hĺbky 2, atď., čo dáva geometrický rad. Uvažujme teraz výpočet algoritmu medzi dvoma akciami (P4). Ďalšie dôležité pozorovanie je, že vonkajšej bubline kvetu, ktorý je na párnej úrovni v nejakom strome, nikdy nebude náboj

ubúdať (buď ostane na párnej úrovni, alebo sa stane súčasťou inej bubliny na párnej úrovni v akcii **(P3)**). Bublinu, ktorá sa niekedy počas výpočtu vyskytovala ako vonkajšia bublina kvetu na párnej úrovni budeme volať *bezpečná*. Na dôkaz tvrdenia stačí ukázať, že v akciách **(P1)**, **(P2)** a **(P3)** pribúdajú bezpečné bubliny. Keďže bezpečné bubliny sa nikdy nerozpadnú a všetkých bublín je lineárne veľa, dostaneme, že medzi dvoma akciami **(P4)** môže byť najviac lineárne veľa iterácií algoritmu.

Akcia **(P1)** rozbiť bublinu B na nepárnej úrovni a aspoň jedna bublina B' z jej vnútra sa dostane na párnú úroveň. B' ale nemohla byť bezpečná, lebo bezpečná bublina sa nikdy nestane súčasťou bubliny na nepárnej úrovni. Akcia **(P2)** pridá novú bezpečnú bublinu z činky a akcia **(P3)** vyrobí novú bezpečnú bublinu. \square

Implementovať jednu iteráciu je možné priamočiaro v čase $O(nm)$, kde n je počet vrcholov a m je počet hrán: pre každú hranu prejdeme všetky bubliny a zistíme, aké veľké ε nám dovoľuje. Vyberieme hranu s najmenším ohraňením a implementujeme príslušnú akciu. Teda môžeme povedať

Veta 3.15. *Problém MIN-1-FACTOR je riešiteľný v čase $O(n^3m)$.*

Pre hlbavého čitateľa poznamenáme, že s rafinovanejšími dátovými štruktúrami sa výsledný čas dá podstatne zlepšiť; nie je to však cieľom tohto textu.

Relaxované podmienky komplementarity

V predchádzajúcej časti sme ukázali primárno-duálnu metódu založenú na charakterizácii optimálnych riešení pomocou podmienok komplementarity. Pre dvojicu duálnych programov

$$(P) : \min_{\mathbf{x} \in \mathbb{R}^n} \{ \mathbf{c}^T \mathbf{x} \mid A\mathbf{x} \geq \mathbf{b}, \mathbf{x} \geq 0 \}$$

$$(D) : \max_{\mathbf{y} \in \mathbb{R}^m} \{ \mathbf{b}^T \mathbf{y} \mid A^T \mathbf{y} \leq \mathbf{c}, \mathbf{y} \geq 0 \}$$

sú vektory \mathbf{x} a \mathbf{y} optimálnym riešením (P) a (D) práve vtedy, keď

$$\forall 1 \leq j \leq n : \text{ buď } x_j = 0 \text{ alebo } \sum_{i=1}^m a_{ij} y_i = c_j$$

$$\forall 1 \leq i \leq m : \text{ buď } y_i = 0 \text{ alebo } \sum_{j=1}^n a_{ij} x_j = b_i$$

Môže nám táto charakterizácia pomôcť aj v situácii, keď máme skromnejší cieľ nájsť riešenie blízke optimu? Ukážeme, že keď sú podmienky komplementarity porušené iba "trochu", máme riešenia, ktoré sú "blízko" optimálnym. Vetu 3.8 modifikujeme takto:

Veta 3.16. *Nech \mathbf{x} a \mathbf{y} sú prípustné riešenia úloh (P) a (D) a nech pre nejaké $\alpha, \beta \geq 1$ platí*

$$\forall 1 \leq j \leq n : \text{ buď } x_j = 0 \text{ alebo } c_j/\alpha \leq \sum_{i=1}^m a_{ij} y_i \leq c_j$$

$$\forall 1 \leq i \leq m : \text{ buď } y_i = 0 \text{ alebo } b_i \leq \sum_{j=1}^n a_{ij} x_j \leq \beta b_i$$

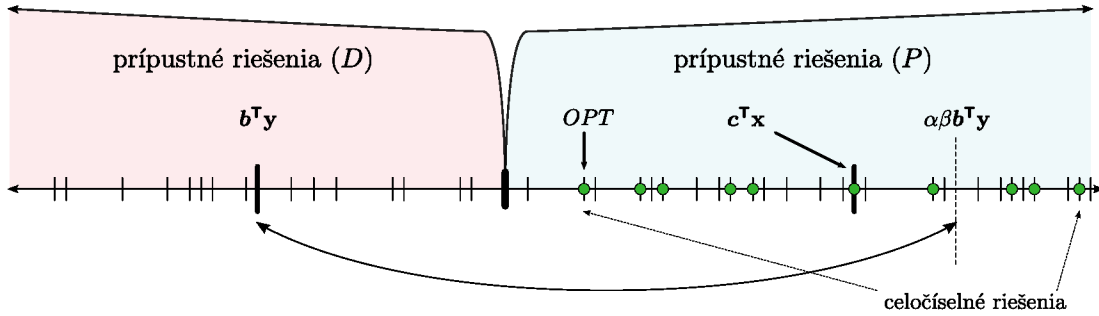
Potom $\mathbf{c}^T \mathbf{x} \leq \alpha \beta \mathbf{b}^T \mathbf{y}$.

Dôkaz:

$$\sum_{j=1}^n c_j x_j \leq \sum_{j=1}^n \alpha \left(\sum_{i=1}^m a_{ij} y_i \right) x_j \leq \alpha \sum_{i=1}^m y_i \left(\sum_{j=1}^n a_{ij} x_j \right) \leq \alpha \beta \sum_{i=1}^m y_i b_i$$

□

Typické použitie vyzerá takto: hľadáme najmenšie celočíselné riešenie (P). Ak hocijakým algoritmom nájdeme celočíselné riešenie \mathbf{x} a k nemu nejaké \mathbf{y} spĺňajúce podmienky Vety 3.16 budeme mať situáciu



OPT aj nájdené riešenie $\mathbf{c}^T \mathbf{x}$ ležia medzi $\mathbf{b}^T \mathbf{y}$ a $\alpha \beta \mathbf{b}^T \mathbf{y}$, preto nájdené riešenie je najviac $\alpha \beta$ -násobok optima.

Ako konkrétny príklad navrhujeme (opäť) 2-aproximačný algoritmus na problém MIN-VERTEX-COVER (Definícia 2.10). Oproti algoritmu, ktorý sme už navrhli v časti o celočíselných programoch sa bude líšiť tým, že bude omnoho rýchlejší, lebo nebude potrebovať riešiť žiaden relaxovaný lineárny program. Začiatok je rovnaký: chceme riešiť celočíselný program

$$\begin{aligned} & \text{minimalizovať} && \sum_{v \in V} \omega_v x_v \\ & \text{pri obmedzeniach} && x_u + x_v \geq 1 \quad \forall e = (u, v) \in E \\ & && x_v \geq 0 \quad \forall v \in V \\ & && x_v \in \mathbb{Z} \end{aligned} \tag{22}$$

urobíme relaxovaný program:

$$\begin{aligned} & \text{minimalizovať} && \sum_{v \in V} \omega_v x_v \\ & \text{pri obmedzeniach} && x_u + x_v \geq 1 \quad \forall e = (u, v) \in E \\ & && x_v \geq 0 \quad \forall v \in V \end{aligned} \tag{23}$$

Teraz ale namiesto toho, aby sme riešili (23), zostrojíme duálny program:

$$\begin{aligned} & \text{maximalizovať} && \sum_{e \in E} y_e \\ & \text{pri obmedzeniach} && \sum_{\substack{e \in E \\ e=(u,v)}} y_e \leq \omega_u \quad \forall u \in V \\ & && y_e \geq 0 \quad \forall e \in E \end{aligned} \tag{59}$$

Kým program (22) vyžaduje, aby sme vybrali vrcholy s najmenšou váhou tak, aby z koncových vrcholov každej hrany bol aspoň jeden vrchol vybraný, program (59) môžeme interpretovať tak, že každej hrane chceme priradiť nezáporný náboj y_e , pričom ω_v bude kapacita vrchola. Cieľom je napumpovať do grafu čo najviac náboja, ale žiaden vrchol sa nesmie preťažiť: súčet nábojov incidentných hrán musí byť najviac kapacita vrchola. Napíšeme si podmienky komplementarity:

$$\mathbf{S1} \quad \forall v \in V : \quad x_v > 0 \Rightarrow \sum_{\substack{e \in E \\ e=(u,v)}} y_e = \omega_u$$

$$\mathbf{S2} \quad \forall e = (u, v) \in E : \quad y_e > 0 \Rightarrow x_u + x_v = 1$$

Z Vety 3.8 vieme povedať toto: keby sme vedeli vybrať množinu vrcholov (t.j. celočíselné hodnoty \mathbf{x}) a dali hranám náboj tak, že žiaden vrchol nie je preťažený (prípustné duálne riešenie), každý vybraný vrchol je plný (podmienky **S1**) a z každej hrany s nenulovým nábojom je vybraný práve jeden vrchol (podmienky **S2**), mali by sme optimálne riešenie. Zjavne najväčší problém robí splnenie podmienky **S2**. Ak si povieme, že sa o **S2** vôbec nebudeme starať, určite bude platiť $x_u + x_v \leq 2$, preto budú splnené podmienky Vety 3.16 pre $\alpha = 1$ a $\beta = 2$.

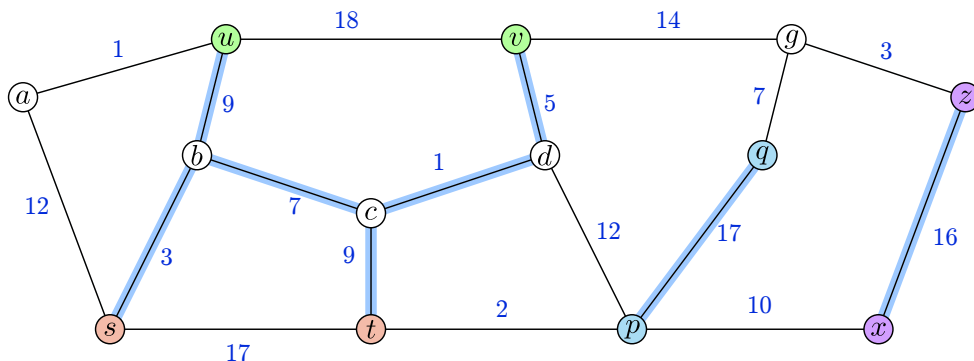
Zostrojíme jednoduchý greedy algoritmus: bude si pamätať množinu vybraných vrcholov C a pre každú hranu e jej aktuálne priradený náboj y_e . Začne s tým, že $C = \emptyset$ a $y_e = 0$ pre všetky e . Algoritmus bude pracovať v cykle: kým C netvorí pokrytie, vyberie ľubovoľnú nepokrytú hranu $e = (u, v)$ a zvýši y_e na maximálnu hodnotu, ktorá nepreťaží vrchol u ani v . Po tejto operácii je aspoň jeden z vrcholov u, v plný (môžu byť aj obidva) a algoritmus ho pridá (ak sú plné obidva, pridá ľubovoľný z nich) do vytváraného pokrytia C .

Po skončení algoritmu platí, že C je pokrytie (hlavný cyklus skončil až keď každá hrana bola pokrytá), žiadny vrchol nie je preťažený a každý vybraný vrchol je plný (obe tieto podmienky sa udržiavajú v platnosti počas celého behu), a preto podľa predchádzajúcich úvah máme garantované, že nájdené riešenie je najviac dvojnásobok optima. Navyše algoritmus pracuje v čase $O(m + n)$, kde n je počet vrcholov a m je počet hrán.

MIN-STEINER-FOREST

V tejto časti si ukážeme, ako využiť relaxované podmienky komplementarity aj v prípade, keď sú aj predpoklady Vety 3.16 príliš silné. Zoberme si nasledovný zjednodušený modelový príklad: máme daný neorientovaný graf, ktorý reprezentuje železničnú sieť medzi mestami: vrcholy zodpovedajú mestám a hrany železničným tratiam. Dopravca, ktorý chce poskytovať prepravné služby, si musí prenajať trate od vlastníka, pričom pre každú trať je daná cena za prenájom. Dopravca sa snaží prenajať trate tak, aby mohol splniť všetky svoje plánované spojenia a pritom zaplatil čo najmenej. Formálnejšie zapísané, chceme riešiť nasledovný problém:

Definícia 3.17. Majme daný graf $G = (V, E)$ a ceny hrán $\omega : E \mapsto \mathbb{R}^+$. Ďalej je daná (symetrická) funkcia požadovaných spojení $r : V \times V \mapsto \{0, 1\}$. Cieľom problému MIN-STEINER-FOREST je vybrať množinu hrán $F \subseteq E$ tak, aby pre každú dvojicu vrcholov u, v takú, že $r(u, v) = 1$, existovala u - v cesta používajúca iba hrany z F . Navyše požadujeme, aby celková cena hrán v F bola najmenšia možná.



Príklad grafu s váhami hrán (modré). Požadované prepojenia sú $u - v$, $s - t$, $p - q$ a $x - z$ (t.j. $r(u, v) = r(v, u) = r(s, t) = r(t, s) = r(p, q) = r(q, p) = r(x, z) = r(z, x) = 1$ a ostatné hodnoty $r(\cdot, \cdot)$ sú 0). Zvýraznené hrany tvoria optimálne riešenie, ktoré má cenu 51 a je tvorené tromi stromami (je dobré si uvedomiť, že komponenty súvislosti optimálneho riešenia budú vždy stromy).

Pre čitateľa oboznámeného s NP -úplnými problémami je jednoduchým cvičením ukázať, že problém MIN-STEINER-FOREST je NP -ťažký, preto ani nebude očakávať, že prezentujeme algoritmus, ktorý ho optimálne rieši. Ukážeme 2-aproximačný algoritmus, t.j. ukážeme, že riešenie, ktoré algoritmus vráti nebude nikdy viac ako dvojnásobok optima. Začneme tradične – problém formulujeme ako celočíselný lineárny program. Celkom prirodzene pre každú hranu $e \in E$ zavedieme premennú $x_e \in \{0, 1\}$, ktorá udáva, či je e vybratá do riešenia. Potrebujeme ešte sformulovať požiadavky na existenciu spojenia formou lineárnych ohraničení. Keďže sa chystáme použiť primárno-duálnu metódu, neostýchame sa použiť exponenciálne veľa obmedzení. Nejakú množinu $S \subseteq V$ nazveme *hladná*, ak z nej musí vo výslednom riešení odchádzať nejaká hrana, t.j. ak existuje $u \in S$, $v \in V \setminus S$, kde $r(u, v) = 1$. Naša formulácia bude založená na tomto pozorovaní:

Lema 3.18. *Ak z každej hladnej množiny odchádza aspoň jedna vybratá hrana, potom vybrané hrany tvoria prípustné riešenie.*

Dôkaz: Zoberme si ľubovoľné $u, v \in V$, také, že $r(u, v) = 1$. Treba nájsť $u - v$ cestu z vybraných hrán. Budeme indukciou konštruovať postupnosť množín $\{u\} = S_0 \subseteq S_1 \subseteq \dots$ s tou vlastnosťou, že pre každý vrchol $w \in S_i$ existuje $u - w$ cesta z vybraných hrán. Pre $\{u\} = S_0$ to zjavne platí. Zoberme si teraz nejakú množinu S_i . Ak $v \in S_i$, máme $u - v$ cestu z vybraných hrán a skončíme dôkaz. Ak nie, S_i je hladná a teda z nej odchádza vybratá hrana do nejakého vrchola $w \in V \setminus S_i$. Zoberme $S_{i+1} := S_i \cup \{w\}$ a vlastnosť, že z u existuje cesta z vybraných hrán do každého vrchola z S_{i+1} , ostane zachovaná. Keďže vrcholov je n , a v každom kroku pridávame do S_i jeden vrchol, časom musíme prísť do situácie, že $v \in S_i$. \square

Označme $f(S) = 1$, ak S je hladná a $f(S) = 0$ inak. Podľa Definície 3.11 označme $\delta(S)$ hranovú hranicu množiny S . Problém MIN-STEINER-FOREST môžeme zapísať nasledovným celočíselným programom:

$$\begin{aligned} & \text{minimalizovať} && \sum_{e \in E} \omega_e x_e \\ & \text{pri obmedzeniach} && \sum_{e \in \delta(S)} x_e \geq f(S) \quad \forall S \subseteq V \\ & && x_e \in \{0, 1\} \quad \forall e \in E \end{aligned} \tag{60}$$

ktorý následne relaxujeme tak, že $x_e \geq 0$; podmienka $x_e \leq 1$ je splnená v každom minimálnom riešení, lebo nemá zmysel zobrať $x_e > 1$. Dostaneme tak

$$\begin{aligned} & \text{minimalizovať} && \sum_{e \in E} \omega_e x_e \\ & \text{pri obmedzeniach} && \sum_{e \in \delta(S)} x_e \geq f(S) \quad \forall S \subseteq V \\ & && x_e \geq 0 \quad \forall e \in E \end{aligned} \quad (61)$$

K programu (61) zostrojíme štandardným spôsobom duálny program: budeme mať premennú y_S pre každú množinu $S \subseteq V$ a zapíšeme

$$\begin{aligned} & \text{maximalizovať} && \sum_{S \subseteq V} y_S f(S) \\ & \text{pri obmedzeniach} && \sum_{S: e \in \delta(S)} y_S \leq \omega_e \quad \forall e \in E \\ & && y_S \geq 0 \quad \forall S \subseteq V \end{aligned} \quad (62)$$

Programy s rovnakou štruktúrou sme tu už stretli niekoľkokrát a tak interpretácia bude celkom prirodzená: okolo každej množiny je bublina s nábojom. Cieľom je maximalizovať náboj na hladných množinách (množiny, ktoré nie sú hladné majú nulový vklad do účelovej funkcie, a tak im nikdy nebudeme žiaden náboj prideľovať) tak, aby sme žiadnu hranu nepreťažili: súčet nábojov na bublinách, ktoré danú hranu pretínajú, musí byť najviac kapacita hrany. Podotýkame, že zvýšiť hodnotu y_S je to isté, ako zvýšiť hodnotu $y_{V \setminus S}$ (lebo pretína tie isté hrany). Napíšme si ešte podmienky komplementarity:

$$\mathbf{S1} \quad \forall e \in E : \quad x_e > 0 \Rightarrow \sum_{S: e \in \delta(S)} y_S = \omega_e$$

$$\mathbf{S2} \quad \forall S \subseteq V : \quad y_S > 0 \Rightarrow \sum_{e \in \delta(S)} x_e = f(S)$$

Z podmienok komplementarity vidíme, že ak by existovalo optimálne celočíselné riešenie, tak každá vybraná hrana by bola plná a z každej nenulovej bubliny okolo hladnej množiny by odchádzala práve jedna hrana (z bublín okolo množín, ktoré nie sú hladné, nejde nič).

Algoritmus, ktorý zostrojíme, bude postupne vyberať hrany, až kým nebudú splnené všetky požiadavky na spojenia. Zároveň sa bude udržiavať invariant, že všetky vybrané hrany sú plné a žiadna hrana nie je preplnená. Preto keď algoritmus skončí, bude mať prípustné riešenie programu (61) a prípustné riešenie programu (62), ktoré zároveň spĺňajú podmienky **S1**. Keby sme chceli použiť Vetu 3.16 na dosiahnutie 2-aproximačného algoritmu, potrebovali by sme navyše zaručiť splnenie podmienok

$$\mathbf{S2'} \quad \forall S \subseteq V : \quad y_S > 0 \Rightarrow \sum_{e \in \delta(S)} x_e \leq 2f(S),$$

t.j. z každej nenulovej bubliny (okolo hladnej množiny) odchádzajú najviac dve vybrané hrany. Toto nebudeme schopní zaručiť, ale ako uvidíme neskôr, bude stačiť slabšie tvrdenie, že *v priemere* z nenulovej bubliny odchádzajú najviac dve vybrané hrany.

Štruktúra algoritmu bude podobná ako pri algoritme pre MIN-VERTEX-COVER z minulej časti: začne s prázdnu množinou hrán a nulovými bublinami. Kým ale algoritmus pre MIN-VERTEX-COVER v jednej iterácii vybral jednu duálnu premennú, zväčšil ju koľko sa dalo a z naplnených vrcholov vybral jeden, teraz algoritmus v jednej iterácii zvýši veľa duálnych premenných naraz a z naplnených hrán vyberie jednu do riešenia.

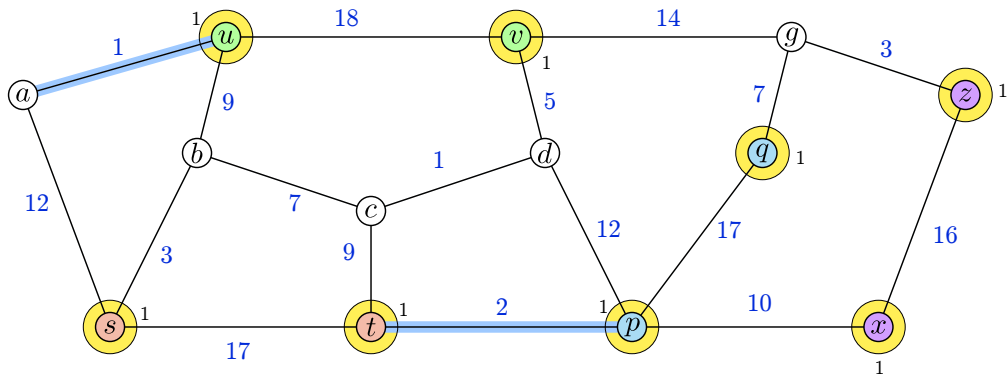
Aké duálne premenné chceme zväčšovať? Určite musia zodpovedať hladným množinám (dohodli sme sa, že množinám, ktoré nie sú hladné, žiaden náboj nedávame). Zároveň, pretože všetky vybrané hrany sú plné, bublinu okolo hladnej množiny, z ktorej už odchádza vybraná hrana, nemôžeme zväčšiť. Zväčšovať teda môžeme bubliny okolo množín, ktoré sú *nespokojné*: sú hladné

(t.j. vo výslednom riešení z nich musí odchádzať nejaká hrana), ale zatiaľ z nich nijaká vybratá hrana neodchádza (inými slovami, sú to množiny, ktoré porušujú prípustnosť programu (61)). Nespokojných množín môže byť potenciálne veľa, ale jednoduché pozorovanie nám povie, že

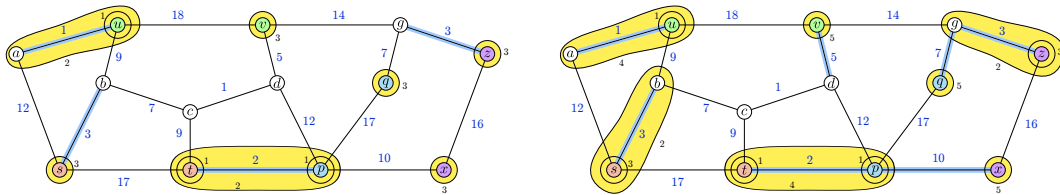
Lema 3.19. *Nespokojné množiny, ktoré sú minimálne vzhľadom na inklúziu, sú komponenty súvislosti podgrafu indukovaného doteraz vybratými hranami.*

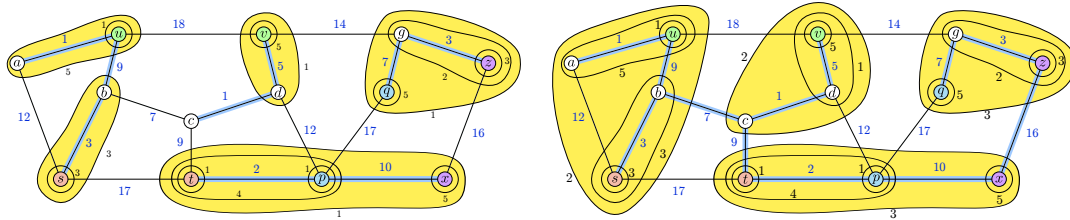
Dôkaz: Množina je nespokojná, keď je hladná a neodchádza z nej vybratá hrana. Preto každá nespokojná množina je zjednotenie nejakých komponentov súvislosti grafu, indukovaného vybratými hranami. \square

Algoritmus bude v každom kroku zväčšovať duálne premenné, ktoré zodpovedajú nespokojným komponentom súvislosti. Na začiatku nie sú vybraté žiadne hrany, a teda nespokojné komponenty súvislosti sú jednoprvkové množiny okolo vrcholov, ktoré potrebujú byť spojené. Pre zadanie z úvodného príkladu sú to množiny $\{u\}, \{v\}, \{s\}, \{t\}, \{p\}, \{q\}, \{x\}, \{z\}$. Algoritmus zväčšuje príslušné duálne premenné, až kým sa niektoré hrany nenaplnia. V našom prípade sa duálne premenné zvýšia na 1 a naplnia sa hrany $\{a, u\}$ a $\{t, p\}$.



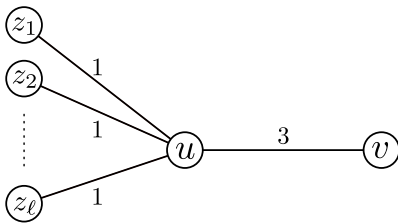
V nasledujúcej iterácii sú nespokojné komponenty $\{a, u\}, \{s\}, \{v\}, \{t, p\}, \{q\}, \{z\}, \{x\}$ a algoritmus zvýši ich hodnoty o 2, čím sa naplnia hrany $\{s, b\}$ a $\{g, z\}$, vzniknú nové nespokojné komponenty a výpočet pokračuje ako na nasledovných obrázkoch.





Algoritmus skončí, keď sú všetky komponenty súvislosti, tvorené vybratými hranami, spokojné. Z technických dôvodov nám bude príjemnejšie uvažovať, že v jednej iterácii sa vyberie iba jedna z naplnených hrán. Ak sa naplnilo viacero hrán, nasledujúce iterácie zvýšia duálne premenné o nulovú hodnotu a postupne poprikladávajú potrebné hrany.

Takto popísaný algoritmus má ešte jeden problém, a to, že nefunguje. Ako príklad zoberme nasledovný graf, v ktorom je jediná požiadavka $r(u, v) = 1$.



Na začiatku sú nespokojné komponenty $\{u\}$ a $\{v\}$ a algoritmus dá každému z nich hodnotu 1. Tým sa ale zaplnia hrany $\{z_i, u\}$, takže algoritmus nakoniec vyberie všetky hrany, a teda vyprodukuje riešenie s cenou $\ell + 3$, ktoré ale nie je dobrá aproximácia optima s cenou 3. Pokúsime sa náš algoritmus zachrániť priamočiarou úpravou: po skončení algoritmu získané riešenie upravíme tak, že odstránime prebytočné hrany,

t.j. hrany, po ktorých odobratí ostanú všetky požiadavky na spojenia splnené. Výsledný algoritmus bude vyzeráť takto:

-
- 1 $F := \emptyset$, $y_{\{v\}} := 0$ pre všetky $v \in V$
 - 2 kým existuje nespokojný komponent súvislosti indukovaný hranami z F
 - 3 zvýš hodnoty y_S pre všetky množiny S zodpovedajúce nespokojným komponentom súvislosti z F tak, aby sa naplnila nejaká hrana e
 - 4 $F := F \cup e$
 - 5 $F' := F$
 - 6 pre každú hranu $e \in F$
 - 7 ak $F - \{e\}$ je prípustné riešenie, $F' := F' - \{e\}$
-

Najprv sa presvedčíme, že algoritmus je korektný:

Lema 3.20. *Po skončení algoritmu tvoria hrany F' prípustné riešenie programu (60) a hodnoty y tvoria prípustné riešenie programu (62).*

Dôkaz:. Po skončení cyklu na riadku 2 neexistuje nespokojný komponent súvislosti tvorený hranami F . Pretože každá nespokojná množina je zjednotením komponentov súvislosti tvorených F , z každej hladnej množiny odchádza vybratá hrana, a teda F je prípustné riešenie (60). Ešte sa treba presvedčiť, že riadky 6 a 7 túto vlastnosť neporušia. Hranu $e \in F$, pre ktorú platí, že $F - \{e\}$ je prípustné riešenie, nazveme *zbytočná*. Ukážeme, že keď z F vyhodíme naraz všetky zbytočné hrany, dostaneme prípustné riešenie F' . Na to si treba uvedomiť, že hrany F indukujú acylický graf: na riadku 3 vždy zvyšujeme hodnoty pre komponenty súvislosti, preto sa nikdy nenaplní hrana vo vnútri komponentu; každá naplnená hrana spája dva komponenty súvislosti, takže nevytvorí cyklus. Preto ak máme dva vrcholy u, v , ktoré musia byť spojené, t.j. $r(u, v) = 1$, je v grafe indukovanom F práve jedna cesta z u do v , a všetky jej hrany sa preto ocitnú v F' . Na druhej strane, hodnoty y_S sme vždy zvyšovali tak, aby sa žiadna hrana nepreplnila, takže počas celého algoritmu tvoria prípustné riešenie (62). \square

Vidíme, že po skončení algoritmu máme nejaké riešenie F' , a nejaké riešenie \mathbf{y} duálneho programu. Na to, aby sme ohranili aproximáciu pomeru, potrebujeme porovnať hodnotu riešenia F' s hodnotou optima. Tú nepoznáme, ale vieme, že je určite väčšia (alebo rovná) ako hocikáka prípustná hodnota duálneho programu. Preto ak chceme dokázať, že algoritmus vždy vráti najviac dvojnásobok optima, stačí dokázať, že hodnota riešenia F' je najviac dvojnásobok hodnoty \mathbf{y} , t.j.

Veta 3.21.

$$\sum_{e \in F'} \omega_e \leq 2 \sum_{S \subseteq V} y_S f(S)$$

Dôkaz:. Pretože sa nikdy nezvyšovali hodnoty y_S pre množinu S , ktorá nie je hladná, $f(S) = 0$ implikuje $y_S = 0$, a teda chceme dokázať

$$\sum_{e \in F'} \omega_e \leq 2 \sum_{S \subseteq V} y_S.$$

Zavedme si teraz nasledovné označenie: pre nejaké množiny $W \subseteq E$ a $S \subseteq V$ označme $\deg_W(S) := |W \cap \delta(S)|$, t.j. počet hrán z W , ktoré majú jeden koniec v S a druhý mimo S . Každá hrana, ktorú sme vybrali do F (a teda aj do F') je plná (hodnoty y nikdy neznižujeme, teda ak hrana raz bola plná, je plná aj na konci), a preto na ľavej strane platí

$$\sum_{e \in F'} \omega_e = \sum_{e \in F'} \left(\sum_{S: e \in \delta(S)} y_S \right) = \sum_{S \subseteq V} \deg_{F'}(S) y_S.$$

Potrebujeme teda dokázať

$$\sum_{S \subseteq V} \deg_{F'}(S) y_S \leq 2 \sum_{S \subseteq V} y_S \quad (63)$$

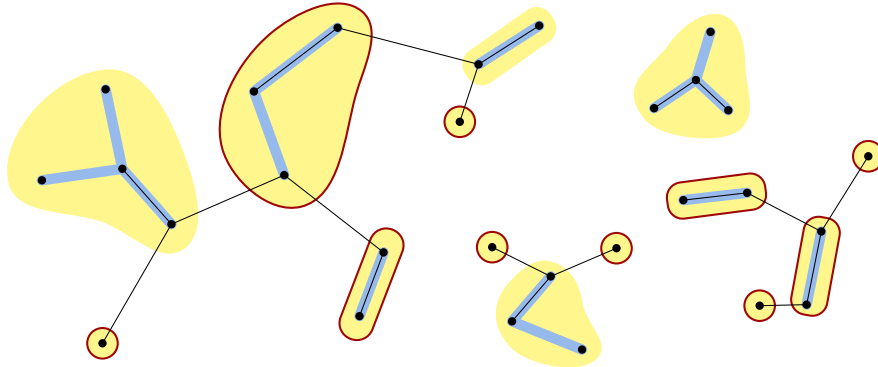
Ukážeme to indukciou na počet iterácií algoritmu. Na začiatku sú všetky hodnoty $y_S = 0$, takže (63) triviálne platí. Uvažujme teraz iteráciu ℓ , v ktorej sa na riadku 3 zvýšili y_S pre všetky nespokojné komponenty o hodnotu Δ . Ako sa zmení (63)? Pre každý nespokojný komponent S pribudne na ľavej strane $\Delta \deg_{F'}(S)$ a na pravej 2Δ . Aby sme dokázali (63), ukážeme, že na pravej strane pribudlo viac ako na ľavej, t.j.

$$\Delta \left(\sum_{S \in \mathcal{S}_\ell} \deg_{F'}(S) \right) \leq 2\Delta |\mathcal{S}_\ell|, \quad (64)$$

kde \mathcal{S}_ℓ je množina všetkých nespokojných komponentov v tejto iterácii. Nerovnosť (64) sa dá napísať ako

$$\frac{\sum_{S \in \mathcal{S}_\ell} \deg_{F'}(S)}{|\mathcal{S}_\ell|} \leq 2,$$

t.j. potrebujeme ukázať, že priemerný stupeň nespokojného komponentu vzhľadom na F' je najvyššie 2. Označme F_ℓ množinu algoritmom vybratých hrán na začiatku iterácie ℓ . Situácia na začiatku iterácie vyzerá ako na obrázku:



Čierne hrany sú výsledné riešenie F' . Modré hrany sú algoritmom doteraz vybraté hrany F_ℓ .
Zvýraznené komponenty súvislosti F_ℓ sú nespokojné.

Hrany F' aj F_ℓ sú podmnožiny hrán F . Pretože F indukujú les (viď. dôkaz Lemy 3.20), aj $F' \cup F_\ell$ tvoria les. Ak skontrahujeme každý komponent súvislosti F_ℓ do jedného vrchola (žlté oblasti v predchádzajúcom obrázku), dostaneme les H , ktorého vrcholy zodpovedajú komponentom súvislosti F_ℓ a hrany sú tvorené hranami F' . Dokázať nerovnosť (64) znamená ukázať, že priemerný stupeň vrchola v H je nanajvýš dva, pričom priemer je braný cez vrcholy, ktoré zodpovedajú nespokojným komponentom. Keby sa priemer bral cez všetky vrcholy, mohli by sme dôkaz skončiť: les s n vrcholmi má najviac $n - 1$ hrán, takže súčet stupňov vrcholov je najviac $2(n - 1)$, a preto priemerný stupeň je menší ako 2. Na to, aby sme ohrančili priemerný stupeň nespokojných komponentov ukážeme, že spokojné komponenty nemajú stupeň 1: buď sú izolované, alebo majú stupeň aspoň 2. Keď to dokážeme, dôkaz dokončíme takto: žiaden nespokojný komponent nemá v H stupeň 0, pretože z F' z neho odchádza aspoň jedna hrana. Izolované vrcholy v H preto všetky zodpovedajú spokojným komponentom. Keď z H vyhodíme izolované vrcholy, dostaneme nový les H' : stačí nám ukázať, že priemerný stupeň nespokojných komponentov v H' je nanajvýš 2. Lenže priemerný stupeň všetkých komponentov je nanajvýš 2 a každý spokojný komponent má stupeň aspoň 2, takže priemerný stupeň nespokojných komponentov nemôže byť väčší ako 2.

Na záver ukážeme, že spokojný komponent nemôže mať v H stupeň 1. Predpokladajme sporom, že existuje komponent súvislosti C v F_ℓ , z ktorého v F' odchádza práve jedna hrana e . Hrana e sa do F' dostala preto, lebo nie je zbytočná, t.j. leží na jedinej ceste, ktorá v F spája dva vrcholy u a v , kde $r(u, v) = 1$. Lenže ak e je jediná hrana, ktorá v F' odchádza z C , potom vrcholy u a v musia ležať jeden v C a druhý mimo C , a preto C nemôže byť spokojný v F_ℓ . \square