

Teorie grafů a diskrétní optimalizace 2

Pracovní texty přednášek

<http://www.kma.zcu.cz/TGD2>

V předmětu KMA/TGD2 se využívá základní přehled hlavních pojmů a poznatků z teorie grafů, a (zejména) znalost základů teorie výpočetní složitosti, získaná v prerekvizitním předmětu KMA/TGD1.

Cílem předmětu je další prohloubení poznatků z teorie grafů a souvisejících oblastí se zaměřením především na optimalizační problémy a metody diskrétní optimalizace. Těžištěm probírané látky je lineární programování, které ovšem z hlediska koncepce předmětu není cílem, nýbrž prostředkem k pochopení vzájemné převoditelnosti většiny úloh diskrétní optimalizace. Důraz je kladen na algoritmickou stránku všech zkoumaných problémů. Hlavní dovedností, kterou by si student měl z předmětu odnést, je celkový přehled základních metod diskrétní optimalizace a jejich vzájemné převoditelnosti a cit pro výpočetní složitost optimalizačních problémů a příslušných algoritmů.

Podobně jako u textu pro předmět KMA/TGD1, je i tento text jen pomocným textem k přednáškám, který si neklade ambice být považován za ucelený učební text typu učebnice či skript. Obsahuje sice všechny základní pojmy, tvrzení a jejich důkazy, ale chybí v něm mnohé komentáře, které jsou pro pochopení látky nezbytné.

Děkuji studentovi 5. ročníku FAV Přemyslu Holubovi za přepis mých pracovních poznámek do TEXu, který se stal základem tohoto textu.

Únor 2001,

Z. Ryjáček

Oproti předchozí (sedmé) verzi, datované únor 2009, bylo provedeno několik oprav drobných chyb a překlepů.

Únor 2016,

Z. Ryjáček

Obsah

1	Optimální tok	3
1.1	Formulace úlohy	3
1.2	Záporné rezervní polocykly	5
1.3	Vyhledávání záporných rezervních polocyklů a Floydův algoritmus	9
1.4	Metoda potenciálů	11
1.5	Převod některých grafových úloh na úlohu optimálního toku	16
2	Lineární programování	18
2.1	Tři základní formulace úlohy lineárního programování	19
2.2	Simplexový algoritmus	22
2.3	Dvoufázová metoda	33
2.4	Degenerované úlohy a zacyklení	35
2.5	Výpočetní složitost simplexového algoritmu a úlohy lineárního programování	38
3	Dualita úloh lineárního programování	39
3.1	Primární a duální úloha lineárního programování	39
3.2	Věty o dualitě	40
3.3	Dobré charakteristiky	42
4	Celočíselné lineární programování	44
4.1	Formulace úlohy	44
4.2	Výpočetní složitost úlohy celočíselného lineárního programování	46
4.3	Celočíselné lineární programování a totálně unimodulární matice	49
4.4	Enumerativní metody a metoda větví a mezí	51
4.5	Metoda sečných nadrovin	54
5	Párování	57
5.1	Základní pojmy, perfektní párování	57
5.2	Rozšiřující cesty a Bergeova věta	60
5.3	Největší párování v bipartitním grafu a maďarská metoda	61
5.4	Optimální párování v ohodnocených bipartitních grafech	65
5.5	Perfektní párování v neorientovaném grafu a Tutteova věta	68
5.6	Největší párování v neorientovaném grafu, párovací číslo a deficience	70
5.7	Edmondsův algoritmus	71
6	Hranové grafy	74
6.1	Základní pojmy	74
6.2	Charakterizační věty	75
6.3	Hranové grafy a výpočetní složitost	76
7	Rovinné grafy	77
7.1	Uložení grafu na plochu	77
7.2	Eulerova věta a její důsledky	78
7.3	Kuratowského věta a další věty o rovinných grafech	80
8	Problém obchodního cestujícího	83
8.1	Metoda penalizací	83
8.2	Kružnice malé váhy v úplném ohodnoceném grafu	85

1 Optimální tok

1.1 Formulace úlohy

V této kapitole budeme uvažovat síť s více zdroji a více stoky s pevně danými intenzitami a budeme předpokládat, že v síti existuje tok. Obecně takových toků existuje více a je možno hledat tok z některého hlediska optimální. Přiřadíme proto každé hraně grafu \vec{G} další číslo c_{ij} , které budeme podle jednoho způsobu jeho interpretace nazývat cenou.

Definice 1.1. *Bud' \vec{G} síť s propustností hran r_{ij} a s intenzitami uzlů a_i . Necht' pro každou hranu $(i, j) \in H(\vec{G})$ je dáno číslo $c_{ij} \in \mathbf{R}$, nazývané cena. Je-li x tok v \vec{G} , pak se číslo*

$$c(x) = \sum_{(i,j) \in H(\vec{G})} c_{ij} x_{ij}$$

nazývá cena toku x . Tok v \vec{G} , který má minimální cenu, se nazývá optimální tok v síti \vec{G} .

Příklad. Plánování přepravy nákladů.

Má-li síť \vec{G} význam transportní sítě a čísla c_{ij} znamenají cenu za přepravu jednotky produktu po hraně (i, j) , pak má cena toku $c(x)$ význam celkových přepravních nákladů a optimální tok představuje rozvozní plán, který uspokojí dané přepravní nároky s minimálními celkovými náklady.

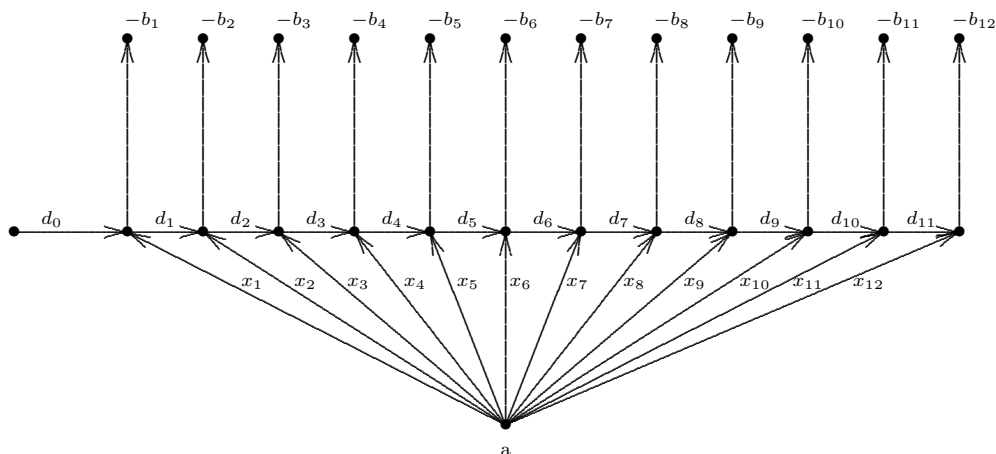
Příklad. Plánování výroby a skladování.

Uvažujme podnik, který musí každý měsíc plánovat svoji výrobu tak, aby byl schopen zabezpečit výrobu podle zadaného grafikonu, jehož hodnoty jsou kolísavé. V této situaci jsou různé způsoby řešení: je například možno každý měsíc vyrobit přesně tolik, kolik žádá grafikon - v tom případě však narůstají ztráty, způsobené kolísáním výroby. Je též možno udržovat konstantní výrobu a kolísání poptávky zabezpečovat pomocí skladu - při malé poptávce vyrábět pro sklad, při velké sklad vyprázdnit. V tomto případě narůstají skladovací náklady. Úkolem je nalézt takový plán výroby, aby celkové ztráty byly minimální.

Předpokládejme, že se na začátku ve skladu nachází d_0 jednotek produktu a označme:

- x_m produkci závodu v m -tém měsíci,
- b_m množství produktu potřebné v m -tém měsíci,
- d_m množství produktu nepoužitého na konci m -tého měsíce (tj. zásobu),
- c_m náklady na rozšíření výroby mezi m -tým a $m + 1$ -tým měsícem,
- p_m náklady na skladování jednotky produktu mezi m -tým a $m + 1$ -tým měsícem.

Pak se snadno přesvědčíme, že řešení naší úlohy se převede na úlohu o optimálním toku v síti na následujícím obrázku:



Čísla x_m a d_m interpretujeme jako hodnoty toku a čísla c_m a p_m jako ceny; cena toku (kterou minimalizujeme) je dána výrazem

$$\sum_{m=1}^{11} c_m(x_{m+1} - x_m) + \sum_{m=1}^{11} p_m d_m;$$

propustnosti všech hran uvažujeme nekonečné.

Intenzita a uzlu, z něhož vycházejí toky x_1, \dots, x_{12} , je dána výrazem $a = \sum_{i=1}^{12} b_i - d_0$; v i -tém neutrálním uzlu je $x_i + d_{i-1} - d_i = b_i$, $i = 1, \dots, 11$ (pro $i = 12$ je $x_i + d_{i-1} = b_i$), což je v souladu s formulací úlohy.

V předchozích dvou příkladech z podstaty úlohy vyplývalo, že ceny c_{ij} musí být nezáporná čísla. V následujícím příkladu uvidíme, že v praktických situacích se mohou vyskytnout i záporné ceny a tedy náš předpoklad $c_{ij} \in \mathbf{R}$ z definice optimálního toku má opodstatnění.

Příklad. Úloha o skladu.

Ve skladu, do něhož se vejde nejvýše k výrobků, je k dispozici k_0 výrobků. V průběhu n dnů se realizuje prodej a nákup těchto výrobků. Předpokládejme, že jsou známy:

- p_i - cena při prodeji jednoho výrobku v i -tý den,
- q_i - kupní cena (od dodavatele) jednoho výrobku v i -tý den pro doplnění skladu,
- s_i - náklady na skladování jednoho výrobku v i -tý den,
- Q - celkový počet výrobků, které je možno od dodavatele obdržet za uvažovaných n dnů.

Naším úkolem je určit:

- α_i - počet výrobků, prodaných v i -tý den,
- β_i - počet výrobků, koupených od dodavatele v i -tý den,
- γ_i - počet skladovaných výrobků po prodeji skladovaných,
- δ_i - počet skladovaných výrobků po koupi nových od dodavatele;

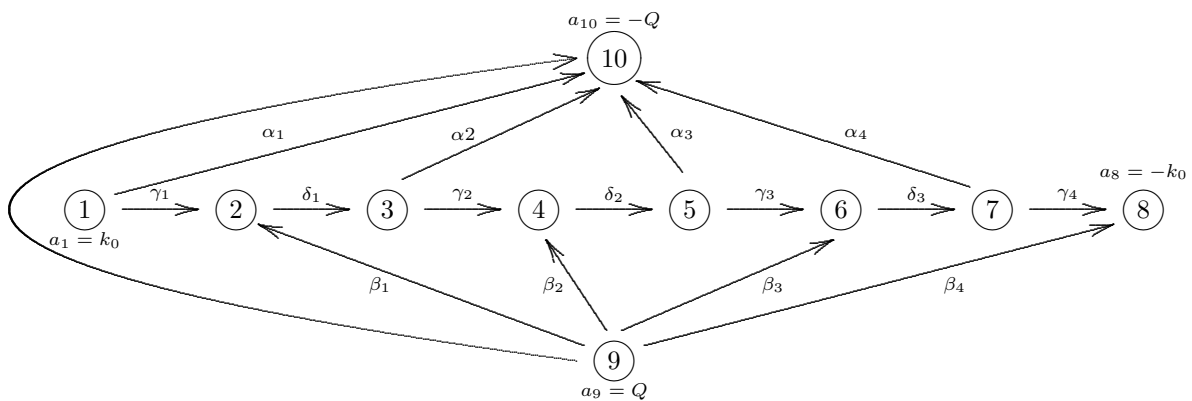
přičemž má být maximalizován celkový zisk

$$\sum_{i=1}^n (p_i \alpha_i - q_i \beta_i - s_i \delta_i),$$

což je zřejmě ekvivalentní s minimalizací výrazu

$$\sum_{i=1}^n (-p_i \alpha_i + q_i \beta_i + s_i \delta_i).$$

Pro zjednodušení předpokládejme, že se požaduje, aby v n -tý den bylo ve skladu opět k_0 výrobků. Úloha se převede na úlohu o optimálním toku v síti, jež je pro $n = 4$ na následujícím obrázku.



Uzly $1, \dots, 8$ odpovídají stavům, v nichž se sklad nachází v jednotlivé dny: uzel 1 popisuje výchozí stav a má proto intenzitu $a_1 = k_0$, uzel 2 popisuje stav po prodeji části výrobků v první den atd., až uzel 8 popisuje konečný stav skladu a má proto intenzitu $a_8 = -k_0$; intenzity uzlů 2 až 7 jsou rovny nule. Uzel 9 odpovídá dodavateli, uzel 10 zákazníkovi. Protože $a_1 = -a_8$, musí také být $a_9 = -a_{10} = Q$. Vzhledem k tomu, že je možné, aby se koupilo a prodalo i méně než Q výrobků, je v síti fiktivní hrana $(9, 10)$ s nulovou cenou a nekonečnou propustností. Propustnosti hran $(i, i + 1)$ pro $i = 1, \dots, 7$ jsou k , ostatní hrany mají nekonečnou propustnost. Hledaná čísla $\alpha_i, \beta_i, \delta_i$ interpretujeme jako hodnoty toku, čísla $-p_i, q_i, s_i$ jako ceny a hledáme tok, který minimalizuje cenovou funkci

$$\sum_{i=1}^n (-p_i \alpha_i + q_i \beta_i + s_i \delta_i).$$

Je však otázkou, jak poznáme, že daný tok je optimální, chybí nám tedy charakterizační věta. K její formulaci potřebujeme nejprve zavést některé pojmy.

1.2 Záporné rezervní polocykly

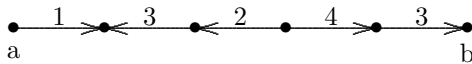
Definice 1.2. *Bud' \vec{G} orientovaný graf. Potom polocestou v grafu \vec{G} z uzlu a do b nazveme cestu z a do b v symetrizaci grafu \vec{G} . Hrany orientované ve směru z a do b nazveme souhlasné, ostatní hrany polocesty, které mají opačnou orientaci, nazveme nesouhlasnými.*

Definice 1.3. *Analogicky jako polocestu můžeme definovat i polocyklus v grafu \vec{G} .*

Definice 1.4. *Cena polocesty (polocyklu) je číslo*

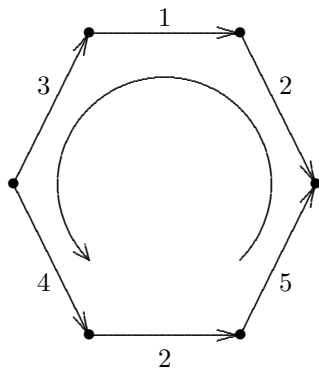
$$c(P) = \sum_{h \in H(P)}^{suhl.} c(h) - \sum_{h \in H(P)}^{nesuhl.} c(h).$$

Příklad.



Cena polocesty: $1 - 3 - 2 + 4 + 3 = 3$.

Příklad.



Cena polocyklu: $5 - 2 - 1 - 3 + 4 + 2 = 5$.

Následující (téměř samozřejmé) tvrzení ukazuje význam ceny polocesty, resp. polocyklu.

Tvrzení 1.1. Cena $c(P)$ (resp. $c(C)$) představuje změnu ceny toku, změníme-li na polocestě P (resp. na polocyklu C) tok o 1.

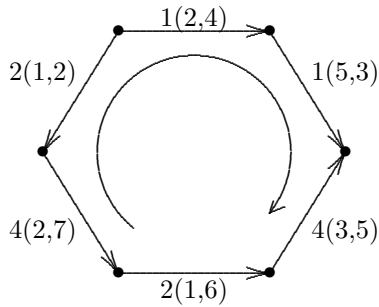
Definice 1.5. Řekneme, že tok x lze zvětšit na polocyklu C (na polocestě P) o δ , jestliže

$$x_{ij} + \delta \leq r_{ij} \text{ pro každou hranu } (i, j) \text{ souhlasnou,}$$

$$x_{ij} - \delta \geq 0 \text{ pro každou hranu } (i, j) \text{ nesouhlasnou.}$$

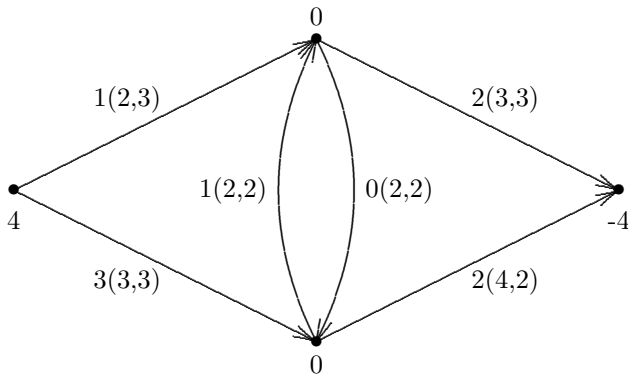
Označení. V dalších příkladech význam čísel u hran bude vždy $x_{ij}(c_{ij}, r_{ij})$.

Příklad.

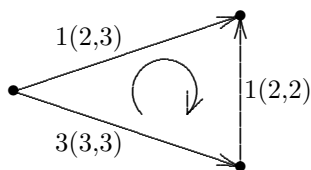


x lze zvětšit na C o 2.

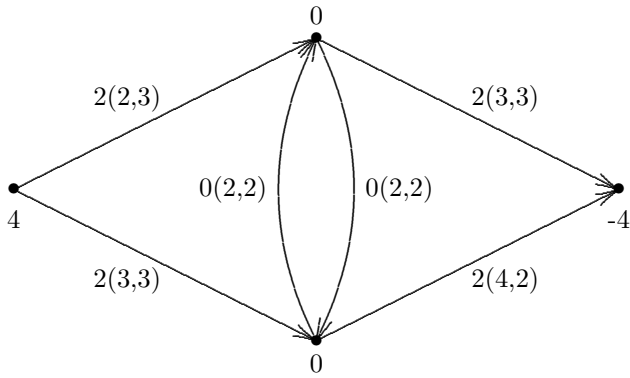
Příklad. Mějme danu následující síť.



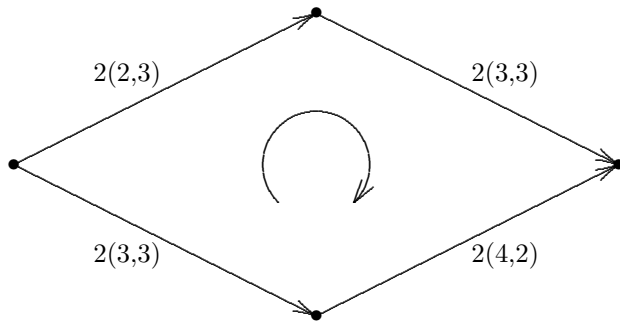
Je zřejmé, že ohodnocení x_{ij} je tok (ověříme podmínky z definice toku). Cena tohoto toku je $c(x) = 27$. Podívejme se nyní na následující polocyklus C .



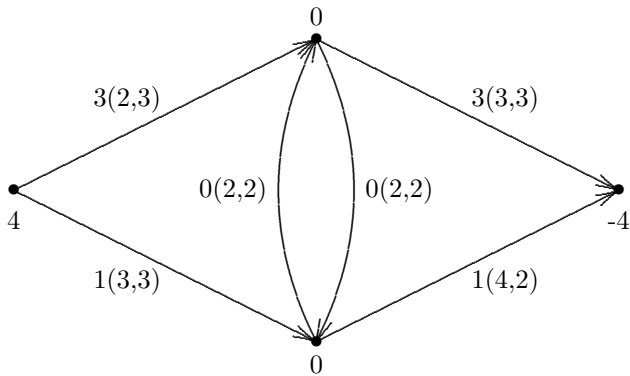
Cena polocyklu je záporná a lze na něm zvětšit tok o 1. Po této úpravě bude naše síť vypadat následovně:



Je evidentní, že nové hranové ohodnocení x_{ij} je opět tokem (proč?). Jeho cena je skutečně nižší, platí $c(x) = 24$. Podívejme se teď na polocyklus:



Cena tohoto polocyklu je opět záporná, a i na tomto polocyklu lze přidat 1. Po úpravě toku dostaneme:



Nové ohodnocení je opět tokem, jeho cena je $c(x) = 22$. Tento tok je již optimální, ale je otázkou, jak optimalitu toku obecně poznat.

Z tohoto příkladu lze vyvodit některé závěry a otázky.

- Cenu toku lze pravděpodobně snižovat zvýšením toku podél polocyklu záporné ceny.
- Jak lze takový polocyklus nalézt ?
- Jak poznáme, že proces končí, tj. že tok je již optimální?

Na některé z těchto otázek nám dá odpověď následující věta.

Věta 1.1. Tok x je optimální, právě když x nelze zvětšit na polocyklu záporné ceny.

Poznámka. Necht' x_{ij} je tok. Polocyklus záporné ceny C , na němž lze x_{ij} zvětšit, se nazývá *záporný rezervní polocyklus* (ZRP). Věta 1.1 tedy říká, že tok je optimální, právě když pro něj neexistuje ZRP.

Důkaz.

- a) x je optimální $\Rightarrow x$ nelze zvětšit na záporném polocyklu. To je zřejmé, neboť kdyby jej šlo zvětšit na nějakém takovém polocyklu C s $c(C) < 0$, tak snížíme $c(x)$ a x by nebyl optimální.

- b) Dokážeme, že jestliže x není optimální, pak x lze zvětšit na polocyklu záporné ceny. Necht' tedy x není optimální. Potom existuje tok y takový, že $c(x) > c(y)$. Položme $z = y - x$. Jaké má z vlastnosti?

1. V každém uzlu j platí

$$\sum_{(j,i) \in H(\vec{G})} z_{ji} - \sum_{(i,j) \in H(\vec{G})} z_{ij} = \left(\sum y_{ji} - \sum y_{ij} \right) - \left(\sum x_{ji} - \sum x_{ij} \right) = a_j - a_j = 0.$$

Ohodnocení z tedy splňuje „podmínky kontinuity“ s nulovými intenzitami.

2. Ale není zaručeno, že na každé hraně e bude $z(e) \geq 0$. To znamená, že z obecně nemusí být tok. To se dá spravit následujícím způsobem.

Definujeme nový graf \vec{G}_1 :

$$U(\vec{G}_1) = U(\vec{G}),$$

$$(i, j) \in H(\vec{G}_1) \Leftrightarrow z_{ij} > 0 \text{ nebo } z_{ji} < 0,$$

a na \vec{G}_1 definujeme tok z^1 předpisem:

$$z_{ij}^1 = z_{ij}, \text{ pokud } z_{ij} > 0, \text{ a}$$

$$z_{ij}^1 = -z_{ji}, \text{ pokud } z_{ji} < 0.$$

Nyní na \vec{G}_1 už z^1 je skutečně tok, a navíc je to tok s nulovými intenzitami (tzv. *circulace*). Platí:

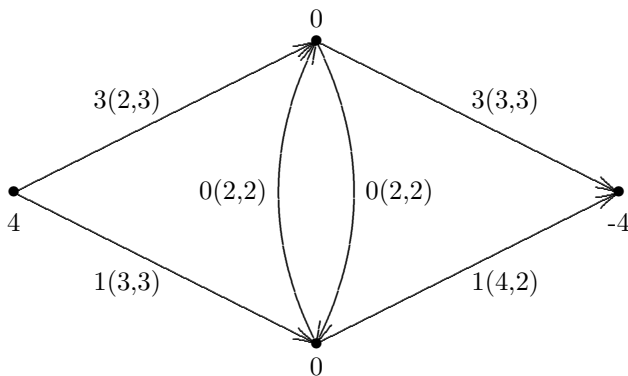
Tok z^1 lze rozložit na součet toků v cyklech grafu \vec{G}_1 .

Důkaz tohoto tvrzení je poměrně jednoduchý a má konstruktivní charakter. Graf \vec{G}_1 obsahuje nějaký cyklus C_1 ; položíme $\delta_1 = \min\{z_{ij}^1, (i, j) \in C_1\}$, na C_1 zmenšíme tok z o δ_1 , čímž získáme nový tok z_1^1 . K němu sestrojíme graf \vec{G}_2 s hranami, na nichž $z_1^1 > 0$. Tento graf \vec{G}_2 obsahuje nějaký cyklus C_2 , a postup opakujeme. Po konečném počtu kroků dostaneme tok z^1 jako součet toků na cyklech C_1, \dots, C_k .

Vraťme se zpět do původního grafu. Každý cyklus C_i odpovídá v \vec{G} nějakému polocyklu Q_i , na kterém lze zvětšit x o δ_i . Ale $y = x + z$, a $c(y) < c(x)$, a tedy některý z polocyklů Q_i musí být záporný.

□

Vraťme se nyní k příkladu ze str. 6, resp. k jeho výsledné podobě.



Nyní již víme, že tok je optimální, pokud pro něj neexistuje ZRP. Otázka je, jak ale algoritmicky poznat záporný rezervní polocyklus.

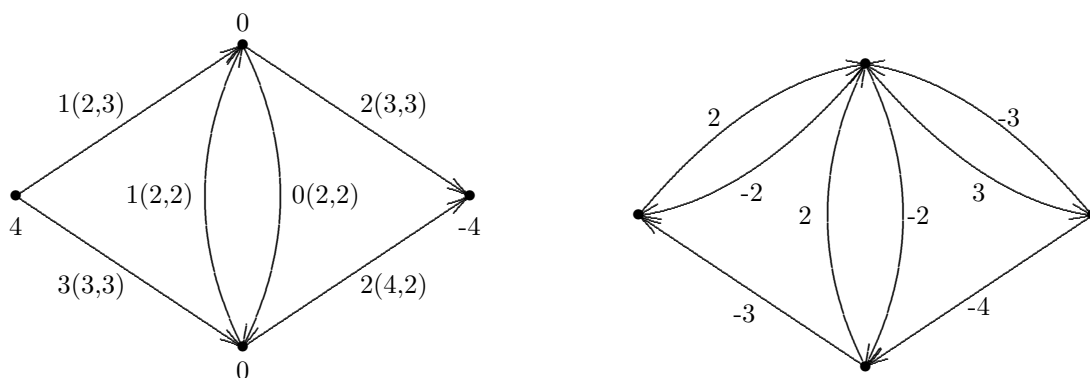
1.3 Vyhledávání záporných rezervních polocyklů a Floydův algoritmus

Pro síť \vec{G} s tokem x_{ij} definujeme pomocný graf \hat{G} takto:

1. \hat{G} je úplný orientovaný graf (bez smyček) na $U(\vec{G})$,
2. ohodnocení hran grafu \hat{G} definujeme předpisem:

$$w_{ij} = \begin{cases} c_{ij} & (x_{ij} < r_{ij}) \wedge (x_{ji} = 0 \text{ nebo } (j, i) \notin H(\vec{G})) \\ -c_{ji} & (x_{ij} = r_{ij} \text{ nebo } (i, j) \notin H(\vec{G})) \wedge x_{ji} > 0 \\ \min\{c_{ij}, -c_{ji}\} & x_{ij} < r_{ij} \wedge x_{ji} > 0 \\ \infty & \text{jinak} \end{cases}$$

Příklad. Síť z našeho příkladu odpovídá následující graf \hat{G} .



A je vidět, že záporný rezervní polocyklus v \vec{G} odpovídá zápornému cyklu v \hat{G} (a naopak).

Problém nalezení ZRP jsme tedy zjednodušili na nalezení algoritmu na vyhledávání záporných cyklů. Připomeňme Floydův algoritmus na nalezení w -distanční matice grafu, který známe z předmětu TGD1 (viz [4], kap. 5.4, str. 180, nebo [11], kap. 4C, str. 122).

$$d_{ij}^0 = \begin{cases} 0 & i = j \\ w_{ij} & i \neq j, (i, j) \in H(\vec{G}) \\ \infty & i \neq j, (i, j) \notin H(\vec{G}) \end{cases}$$

$$d_{ij}^k = \min\{d_{ij}^{k-1}, d_{ik}^{k-1} + d_{kj}^{k-1}\}. \quad (*)$$

Připomeňme, že význam čísel d_{ij}^k je délka minimální cesty množinou uzlů $\{1, \dots, k\}$.

My nyní budeme Floydův algoritmus modifikovat pro účely nalezení záporného cyklu, a to následujícím způsobem:

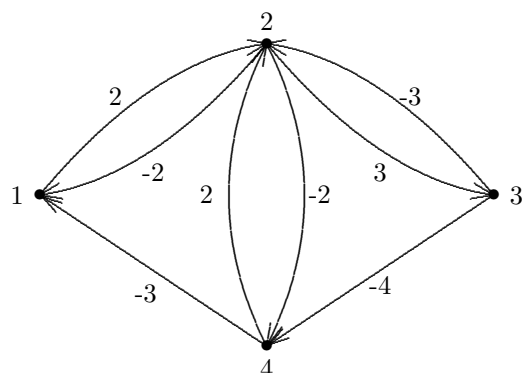
$$d_{ij}^0 = \begin{cases} w_{ij} & \text{je-li hrana } (i, j) \in H(G), \\ \infty & \text{jinak} \end{cases}$$

(tj. na diagonále budou také nekonečna). Budeme počítat matice D^1, D^2, \dots, D^n podle vztahu (*). Protože d_{ij}^k jsou délky minimálních i, j -cest, bude pro $i = j$ hodnota d_{ii}^k mít význam w -délky minimálního i, i -cyklu. Dostáváme tak následující větu.

Věta 1.2. Graf \hat{G} obsahuje záporný cyklus, právě když $d_{ii}^k < 0$ pro některá $i, k \in \{1, 2, \dots, n\}$.

Poznámka. Tento algoritmus nalezne záporné polocykly v čase $O(n^3)$.

Příklad. Pokračování řešení příkladu ze str. 6.



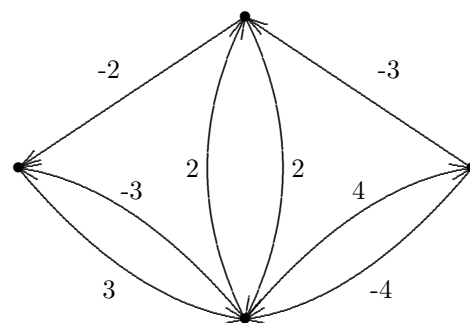
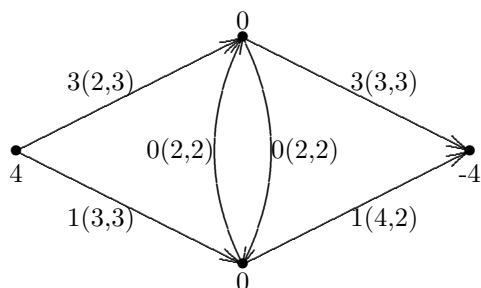
$$D^0 = \begin{bmatrix} \infty & 2 & \infty & \infty \\ -2 & \infty & 3 & -2 \\ \infty & -3 & \infty & -4 \\ -3 & 2 & \infty & \infty \end{bmatrix}$$

$$D^1 = \begin{bmatrix} \infty & 2 & \infty & \infty \\ -2 & 0 & 3 & -2 \\ \infty & -3 & \infty & -4 \\ -3 & -1 & \infty & \infty \end{bmatrix}$$

$$D^2 = \begin{bmatrix} 0 & 2 & 5 & 0 \\ -2 & 0 & 3 & -2 \\ -5 & -3 & 0 & -5 \\ -3 & -1 & 2 & -3 \end{bmatrix}$$

V síti tedy existuje záporný rezervní polocyklus, zpětným prohledáváním matic D^2 , D^1 a D^0 lze získat informaci o tom, které hrany do tohoto polocyklu patří. V našem případě se jedná o polocyklus $1-2-4-1$.

Příklad. Ověřme, že poslední tok v síti z našeho příkladu je optimální.



$$D^0 = \begin{bmatrix} \infty & \infty & \infty & 3 \\ -2 & \infty & \infty & 2 \\ \infty & -3 & \infty & -4 \\ -3 & 2 & 4 & \infty \end{bmatrix}$$

$$D^1 = \begin{bmatrix} \infty & \infty & \infty & 3 \\ -2 & \infty & \infty & 1 \\ \infty & -3 & \infty & -4 \\ -3 & 2 & 4 & 0 \end{bmatrix}$$

$$D^2 = \begin{bmatrix} \infty & \infty & \infty & 3 \\ -2 & \infty & \infty & 1 \\ -5 & -3 & \infty & -4 \\ -3 & 2 & 4 & 0 \end{bmatrix}$$

$$D^3 = \begin{bmatrix} \infty & \infty & \infty & 3 \\ -2 & \infty & \infty & 1 \\ -5 & -3 & \infty & -4 \\ -3 & 1 & 4 & 0 \end{bmatrix}$$

$$D^4 = \begin{bmatrix} \mathbf{0} & 4 & 7 & 3 \\ -2 & \mathbf{2} & 5 & 1 \\ -7 & -3 & \mathbf{0} & -4 \\ -3 & 1 & 4 & \mathbf{0} \end{bmatrix}$$

$d_{ii}^k \geq 0 \forall i \Rightarrow$ je dokázána optimalita toku.

Tato metoda dává polynomiální a prakticky použitelný algoritmus. Při „ruční práci“ je však poněkud zdlouhavá. Jiná metoda bude založena na následující větě.

1.4 Metoda potenciálů

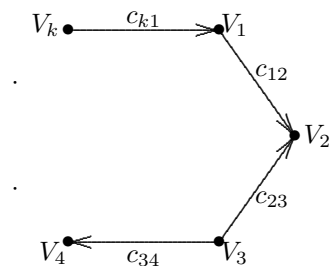
Věta 1.3. Tok x_{ij} je optimální, právě když existuje takové ohodnocení uzlů čísla $V_i, i = 1 \dots n$, že platí

$$\begin{aligned} V_j - V_i &\leq c_{ij}, \text{ jestliže } x_{ij} = 0, \\ V_j - V_i &= c_{ij}, \text{ jestliže } 0 < x_{ij} < r_{ij}, \\ V_j - V_i &\geq c_{ij}, \text{ jestliže } x_{ij} = r_{ij}. \end{aligned}$$

Důkaz.

- Ukážeme, že jestliže existují potenciály, pak je tok optimální.

Kdyby tok nebyl optimální, existoval by záporný rezervní polocyklus C :



Tento polocyklus je záporný, platí tedy $c(C) = \sum_{h \in H(C)}^{suhl.} c(h) - \sum_{h \in H(C)}^{nesuhl.} c(h) < 0$. Označíme-li

$\hat{c}_{i,i+1} = c_{i,i+1}$, je-li $(i, i+1)$ souhlasná,

$\hat{c}_{i,i+1} = -c_{i+1,i}$, je-li $(i, i+1)$ nesouhlasná,

pak platí $\hat{c}_{12} + \hat{c}_{23} + \dots + \hat{c}_{k1} < 0$.

Z vlastností potenciálu vyplývá následující:

- jestliže hrana $(i, i+1)$ je souhlasná, platí $x_{i,i+1} < r_{i,i+1}$ (neboť C je rezervní polocyklus), a z vlastností potenciálů plyne, že $V_{i+1} - V_i \leq \hat{c}_{i,i+1}$, po úpravě $V_{i+1} \leq V_i + \hat{c}_{i,i+1}$,
- jestliže hrana $(i, i+1)$ je nesouhlasná, platí $x_{i+1,i} > 0$, a tedy $V_i - V_{i+1} \geq c_{i+1,i} = -\hat{c}_{i,i+1}$, odkud $V_{i+1} \leq V_i + \hat{c}_{i,i+1}$.

Celkem (kolem celého cyklu) dostaneme

$$V_2 \leq V_1 + \hat{c}_{12}; \quad V_3 \leq V_2 + \hat{c}_{23} \leq V_1 + \hat{c}_{12} + \hat{c}_{23};$$

atd.

$$V_1 \leq V_k + \hat{c}_{k1} \leq V_1 + \hat{c}_{12} + \hat{c}_{23} + \dots + \hat{c}_{k1}.$$

Odtud $\hat{c}_{12} + \hat{c}_{23} + \dots + \hat{c}_{k1} \geq 0$, což je ve sporu s předpokladem, že cyklus C je záporný.

2. Nyní naopak předpokládejme, že tok x_{ij} je optimální, tj. neexistuje v příslušné síti žádný záporný rezervní polocyklus. Pokusíme se nalézt potenciály uzlů V_i , splňující tvrzení věty. Označme $H(x)$ množinu všech hran (i, j) , pro které platí vztah $0 < x_{ij} < r_{ij}$ (tj. protéká jimi nenulový tok, ale hrana není nasycena). Faktor $\vec{G}_x = (U(\vec{G}), H(x))$ grafu sítě \vec{G} nazveme *opora toku* x_{ij} .

Nejprve předpokládejme, že opora je souvislý (slabě) faktor. Potenciály uzlů sestrojíme nyní následujícím postupem. Zvolíme jeden uzel $i_0 \in U(\vec{G})$ a položíme $V_{i_0} = 0$. Pro všechny uzly $j \in U(\vec{G})$, pro něž je $(i_0, j) \in H(x)$, položíme

$$V_j = V_{i_0} + c_{i_0j}$$

a pro všechny uzly $j \in U(\vec{G})$, pro něž je $(j, i_0) \in H(x)$, položíme

$$V_j = V_{i_0} - c_{ji_0}.$$

Tímto způsobem lze na základě hodnoty potenciálu každého uzlu určit hodnoty potenciálu pro všechny uzly, které s ním jsou v opoře spojeny hranou a pro které ještě není určen. Protože opora je souvislý faktor grafu \vec{G} , jsou tím určeny potenciály pro všechny uzly sítě.

Dokážeme, že při takto definovaných číslech V_i platí pro každou hranu (i, j) podmínky věty.

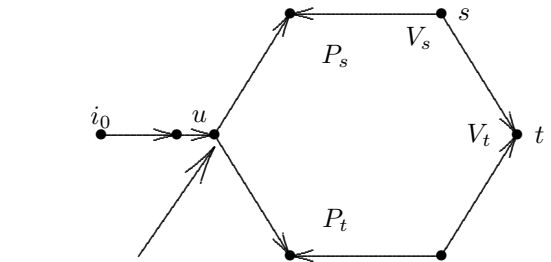
Nechť tedy je dána hrana (s, t) . Spojme v opoře $(U(\vec{G}), H(x))$ uzly s a t s uzlem i_0 polocykly P_s, P_t , pomocí nichž byl potenciál definován. Patří-li hrana (s, t) některé z těchto polocykly, pak na ní podle konstrukce potenciálů platí $V_t - V_s = c_{s,t}$ a jsme hotovi. Předpokládejme tedy, že hrana (s, t) nepatří žádné z těchto polocykly. Pak získáme v symetrizaci grafu \vec{G} kružnici, obsahující hrany těchto dvou cest P_s, P_t a hranu $\{s, t\}$. Tuto kružnici označíme C .

Podle konstrukce potenciálů platí na hranách polocykly P_s a P_t vztah $V_j - V_i = c_{ij}$. Lze tedy psát

$$V_s = V_{i_0} + c(P_s),$$

$$V_t = V_{i_0} + c(P_t).$$

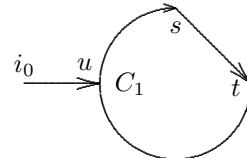
Označme u poslední společný uzel polocykly P_s, P_t .



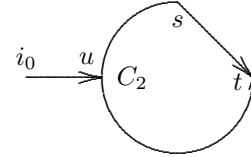
poslední společný uzel

- a) Je-li hrana $(s, t) \in H(x)$, tj. taková, že pro ni platí $0 < x_{st} < r_{st}$, pak by mělo platit $V_t - V_s = c_{st}$.

Je-li $V_t - V_s > c_{st}$, pak je polocyklus $C_1 = st\overleftarrow{P_t}uP_s s$ (viz sousední obrázek) záporný rezervní, což je ve sporu s optimalitou toku.



Je-li $V_t - V_s < c_{st}$, je polocyklus $C_2 = s\overleftarrow{P}_s u P_t t s$ (viz sousední obrázek) záporný rezervní, což je rovněž spor.



- b) Je-li hrana (s, t) nasycená, tj. platí $x_{st} = r_{st}$, pak budeme chtít ukázat, že platí nerovnost $V_t - V_s \geq c_{st}$.

Kdyby naopak platilo $V_t - V_s < c_{st}$, potom by bylo $V_t < V_s + c_{st}$. Pak by však byla polocesta P_t do uzlu t levnější než polocesta $P_s \cup (s, t)$. Vznikl by nám polocyklus $s\overleftarrow{P}_s u P_t t s$, který by byl záporný rezervní. To je však opět spor s optimalitou toku.

- c) Příklad, kdy (s, t) je hrana s nulovým tokem, tj. $x_{st} = 0$, je analogický jako v b).

Tím jsou požadované potenciály v případě souvislé opory sestrojeny. Poznamenejme, že v tomto případě jsou sestrojena čísla V_i určena jednoznačně až na aditivní konstantu a lze je učinit nezápornými.

Zbývá ukázat konstrukci potenciálů v případě nesouvislé opory. Myšlenku jen naznačíme bez podrobného provádění technických detailů důkazu.

- 1) Podle první části důkazu jsou v každé komponentě opory potenciály dány jednoznačně až na aditivní konstantu.
- 2) Vybereme komponentu opory H_0 , zvolíme v ní konstantu c_0 (třeba tak, že v některém uzlu položíme $V = 0$).
- 3) Tvrdím, že ve všech komponentách opory lze zvolit aditivní konstanty tak, že jsou splněny podmínky věty. To ukážeme takto.
 - Označíme G_0 podgraf indukovaný komponentou H_0 opory.
 - Zvolíme komponentu H_1 opory, z níž do H_0 vede alespoň jedna hrana. V H_1 aditivní konstanta c_1 existuje: některé hrany mezi H_0 a H_1 implikují horní odhad c_1 , některé dolní odhad c_1 . Vezmeme ty z nich, které určují nejmenší horní odhad c_1^+ a největší dolní odhad c_1^- („určující hrany“), a komponentě H_1 přiřadíme interval $\langle c_1^-, c_1^+ \rangle$ (tento interval je neprázdný, neboť kdyby $c_1^+ < c_1^-$, tak by určující hrany spolu s některými cestami v H_0 a v H_1 dávaly ZRP).
 - Označíme G_1 podgraf indukovaný sjednocením komponent H_0 a H_1 .
 - Obecně: v i -tém kroku máme $G_{i-1} = H_0 \cup \dots \cup H_{i-1}$, připojujeme H_i , pro každou hranu vedoucí do H_i z G_{i-1} vyhodnotíme příslušný horní či dolní odhad na c_i , vezmeme minimum horních odhadů c_i^+ a maximum dolních odhadů c_i^- . Kdyby $c_i^+ < c_i^-$, existuje ZRP – tedy $\langle c_i^-, c_i^+ \rangle$ je neprázdný interval. Položíme $G_i = G_{i-1} \cup H_i$ a vše opakujeme.
- 4) Po k krocích (kde k je počet komponent opory) zvolíme čísla c_i uvnitř intervalů $\langle c_i^-, c_i^+ \rangle$, $i = 1, \dots, k$.

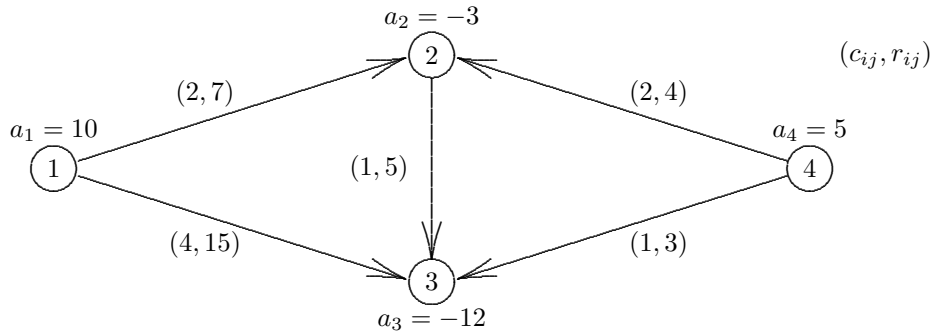
□

Postup, kterým jsme v důkazu věty sestrojili potenciály V_i , poskytuje metodu pro praktické sestrojení optimálního toku v síti. Známe-li v síti nějaký výchozí tok (jenž se dá sestrojit například pomocí algoritmu na sestrojení maximálního toku), pak definujeme postupně (za předpokladu souvislosti opory) potenciály jednotlivých uzlů sítě, a narazíme-li při jejich konstrukci na hranu, na níž nejsou splněny podmínky optimality, pak cenu toku podél některého polocyklu snížíme. K takto „vylepšenému“ toku pak opět sestrojujeme potenciály a celý postup opakujeme tak dlouho, dokud v síti nacházíme hrany, na nichž nejsou splněny podmínky optimality. Výsledkem tohoto iteračního postupu je optimální tok. Popsaná metoda nalezení optimálního toku se nazývá *metoda potenciálů*.

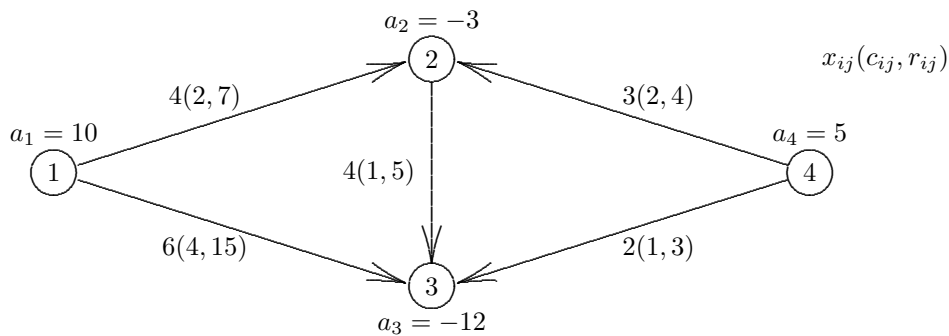
Poznámka. Charakterizační věta 1.1 (o ZRP) je užitečná díky tomu, že o existenci či neexistenci ZRP lze (díky Floydovu algoritmu) rozhodnout v polynomiálním čase.

Poznamenejme však již nyní, že přesto (sama o sobě) tato věta není tzv. dobrou charakteristikou, zatímco věta o potenciálech (Věta 1.3) dobrou charakteristikou je (o dobrých charakteristikách podrobněji viz odst. 3.3).

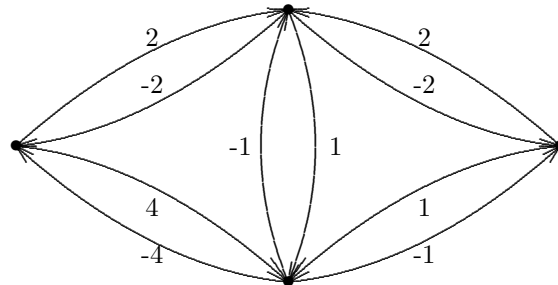
Příklad. V síti na obrázku nalezněte optimální tok. (Čísla v kroužcích mají význam pořadových čísel uzlů, ohodnocení hran je $(c_{i,j}, r_{i,j})$).



Nejprve v naší síti nalezneme (nějaký) výchozí tok.



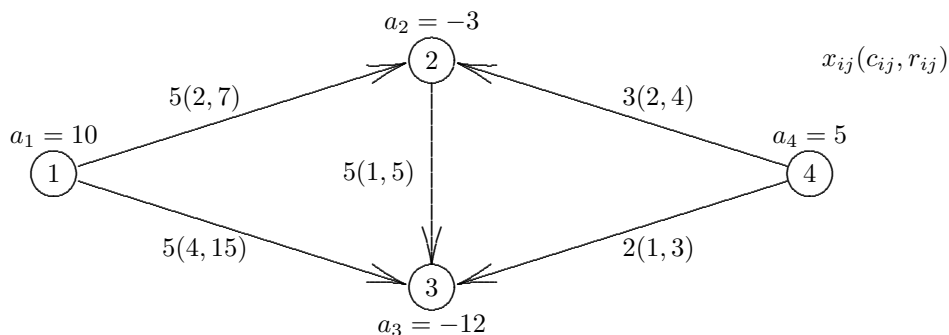
Pro porovnání obou metod zkusíme pomocí Floydova algoritmu, zda v síti existuje záporný rezervní polocyklus. Pomocný graf \hat{G} :



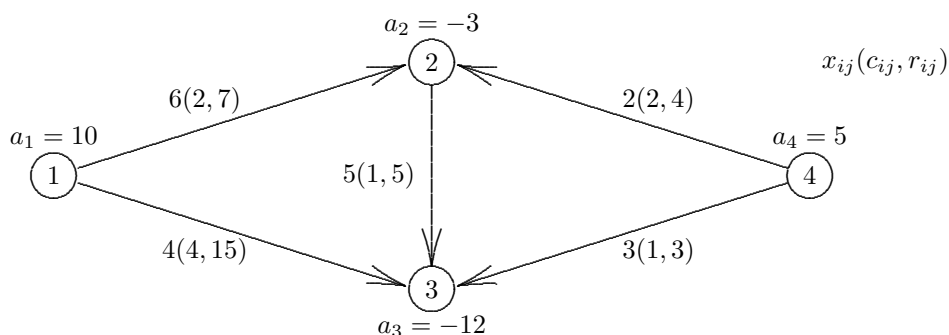
$$D^0 = \begin{bmatrix} \infty & 2 & 4 & \infty \\ -2 & \infty & 1 & -2 \\ -4 & -1 & \infty & -1 \\ \infty & 2 & 1 & \infty \end{bmatrix} \quad D^1 = \begin{bmatrix} \infty & 2 & 4 & \infty \\ -2 & 0 & 1 & -2 \\ -4 & -2 & 0 & -1 \\ \infty & 2 & 1 & \infty \end{bmatrix} \quad D^2 = \begin{bmatrix} 0 & 2 & 3 & 0 \\ -2 & 0 & 1 & -2 \\ -4 & -2 & -1 & \end{bmatrix}$$

Dál není třeba počítat, neboť $d_{33}^2 < 0$. Zpětně z matic D^0, D^1 najdeme ZRP, tvořený hranami s ohodnoceními $w_{12} = 2, w_{23} = 1$ a $w_{31} = -4$.

A nyní totéž provedeme metodou potenciálů. Opora toku je souvislý graf, a proto můžeme zvolit $V_1 = 0$ a dále definovat: $V_1 = 0 \Rightarrow V_2 = 2$ (neboť $c_{12} = 2$) $\Rightarrow V_3 = 3$ (neboť $c_{23} = 1$), ale to je spor, protože $V_3 - V_1 = 3$, ale $c_{13} = 4$. Tok hranou $(1, 3)$ je tedy nutno snížit podél polocyklu s uzly 1, 2, 3, 1. To lze provést nejvýše o 1, protože pak se hrana $(2, 3)$ nasýtí. Vylepšený tok vypadá takto:



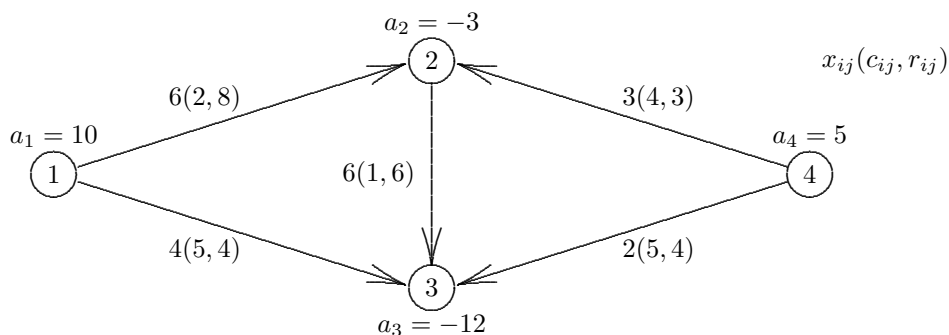
Pro tento tok opět definujeme potenciály uzlů: $V_1 = 0 \Rightarrow V_2 = 2$ (podél hrany $(1, 2)$), a $V_1 = 0 \Rightarrow V_3 = 4$ (podél hrany $(1, 3)$). Na hraně $(2, 3)$ platí $V_3 - V_2 = 2$ a $c_{23} = 1$, což je v souladu s podmínkami optimality ($V_3 - V_2 \geq c_{23}$, neboť hrana $(2, 3)$ již je nasycená a tedy nepatří do opory). Definujeme tedy dále podél hrany $(4, 2)$: $V_4 = 0$. Na hraně $(4, 3)$ dostaneme $V_3 - V_4 = 4$ a $c_{43} = 1$, což je spor. Rozdíl potenciálů je větší než cena, a tedy je x_{43} třeba zvětšit. Podél kružnice s uzly $1, 2, 4, 3, 1$ zvětšíme tok o 1 (více nelze, protože hrana $(4, 3)$ se nasytí). Dostaneme tak následující tok.



Definujeme opět potenciály: $V_1 = 0 \Rightarrow V_3 = 4$ a $V_1 = 0 \Rightarrow V_2 = 2 \Rightarrow V_4 = 0$. Podmínky optimality (věta 1.3) jsou nyní splněny, takže nalezený tok je optimální.

Povšimněme si toho, že celková cena výchozího toku byla $8 + 24 + 4 + 6 + 2 = 44$, prvního opraveného toku $10 + 20 + 5 + 6 + 2 = 43$, celková cena výsledného optimálního toku je $12 + 16 + 5 + 4 + 3 = 40$.

Příklad. V síti na obrázku naleznete optimální tok. Jedná se o mírně číselně modifikovaný předchozí příklad; síť uvádíme již s nalezeným výchozím tokem.

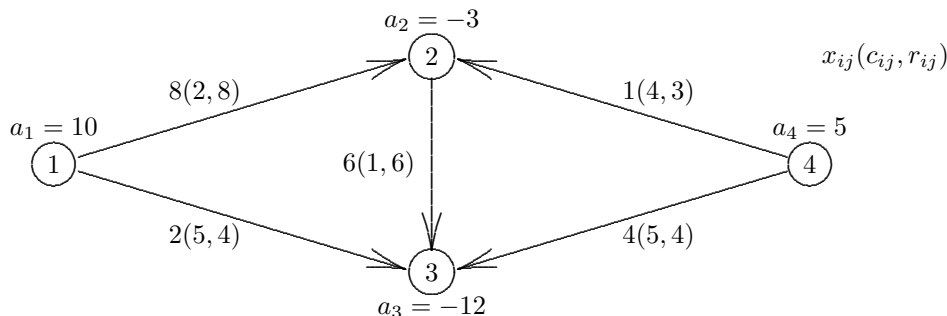


Vidíme, že opora toku je nesouvislý graf, jehož jednu komponentu tvoří uzly $1, 2$ a hrana $(1, 2)$, druhou uzly $3, 4$ a hrana $(4, 3)$. Zvolíme-li $V_1 = 0$, dostáváme (podél hrany $(1, 2)$) $V_2 = 2$, ale tím proces definování

potenciálů končí, protože hrany (1, 3), (2, 3) a (4, 2) jsou nasycené a máme tedy na nich k dispozici pouze nerovnosti. Použijeme postup, který jsme poznali v závěrečné části důkazu věty 1.3. Zvolíme-li $V_4 = c$, pak podél hrany (4, 3) dostáváme $V_3 = c + 5$. Hrany (1, 3), (2, 3) a (4, 2) nám nyní dávají soustavu nerovností:

$$\begin{aligned} \text{hrana (1, 3):} & \quad (c + 5) - 0 \geq 5, & \text{odkud } c \geq 0, \\ \text{hrana (2, 3):} & \quad (c + 5) - 2 \geq 1, & \text{odkud } c \geq -2, \\ \text{hrana (4, 2):} & \quad 2 - c \geq 4, & \text{odkud } c \leq -2. \end{aligned}$$

První a třetí nerovnost si odporují, tedy příslušné hrany, tj. (1, 3) a (4, 2) spolu s cestami uvnitř komponent opory určují ZRP s uzly 1, 2, 4, 3, 1. Zvětšením toku o 2 podél tohoto ZRP dostáváme následující opravený tok.



Opora je opět nesouvislá, její komponenty nyní tvoří uzly 1, 3 s hranou (1, 3) a uzly 2, 4 s hranou (4, 2). Volíme $V_1 = 0$ a podél hrany (1, 3) dostáváme $V_3 = 5$; ve druhé komponentě volíme $V_4 = c$ a podél hrany (4, 2) dostáváme $V_2 = c + 4$. Hrany (1, 2), (2, 3) a (4, 3) nám nyní dávají soustavu nerovností:

$$\begin{aligned} \text{hrana (1, 2):} & \quad (c + 4) - 0 \geq 2, & \text{odkud } c \geq -2, \\ \text{hrana (2, 3):} & \quad 5 - (c + 4) \geq 1, & \text{odkud } c \leq 0, \\ \text{hrana (4, 3):} & \quad 5 - c \geq 5, & \text{odkud } c \leq 0. \end{aligned}$$

Soustava nyní má řešení (vyhovuje jí kterákoliv hodnota c pro $-2 \leq c \leq 0$), nalezený tok je tedy podle věty 1.3 optimální.

Poznamenejme, že cena výchozího toku byla $12 + 20 + 6 + 12 + 10 = 60$ a cena výsledného optimálního toku je $16 + 10 + 6 + 4 + 20 = 56$.

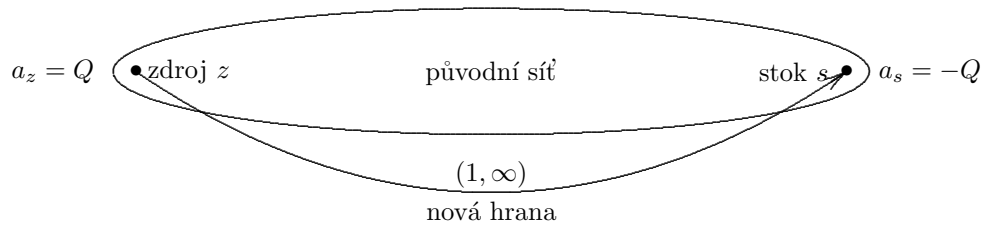
1.5 Převod některých grafových úloh na úlohu optimálního toku

Na úlohu nalezení optimálního toku v síti je možno převést nejen úlohu maximálního toku, ale i řadu grafových úloh, v nichž „nikde nic neteče“. Ukážeme si několik příkladů takových převodů. Praktický význam těchto převodů je v tom, že umožňují použít tokové algoritmy (a příslušný speciální software) na mnoho dalších úloh. Navíc, v příští kapitole uvidíme, že úlohu optimálního toku lze převést na úlohu lineárního programování. To znamená, že všechny úlohy, převoditelné na toky, bude možno řešit algoritmy lineárního programování (zejména simplexovým algoritmem).

Maximální tok v síti. Sestrojíme novou síť následující konstrukcí (viz obrázek).

- k původní síti přidáme novou pomocnou hranu, vedoucí ze zdroje z do stoku s ,
- cena nové hrany je 1, propustnost nové hrany je ∞ ,
- intenzita uzlu z v nové síti je Q , intenzita uzlu s v nové síti je $-Q$, kde Q je „dost velké“ číslo (stačí zvolit Q větší než součet propustností všech hran původní sítě),
- ceny všech hran původní sítě jsou rovny 0,
- intenzity všech uzlů původní sítě kromě z a s jsou rovny 0.

Je zřejmé, že optimální tok v nové síti se na původní síti shoduje s maximálním tokem ze z do s .



Hranový (uzlový) stupeň souvislosti grafu. Otázkou, zda je (pro dané k) daný graf hranově či uzlově k -souvislý, lze převést na úlohu nalezení maximálního toku v síti způsobem známým z předmětu TGD1, kap. 2.3. Kombinace s předchozím převodem tedy dává převod úlohy na úlohu nalezení optimálního toku.

Minimální cesta z uzlu u do uzlu v grafu \vec{G} . Sestrojíme síť takto:

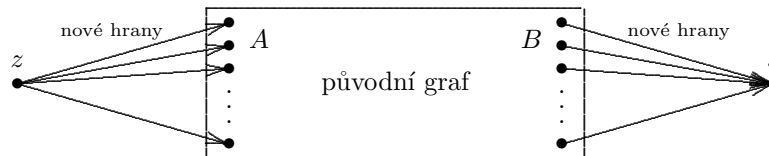
- intenzita u je 1, intenzita v je -1, intenzity ostatních uzlů jsou nulové,
- propustnosti všech hran jsou rovny jedné,
- ceny všech hran jsou rovny jedné.

Pak minimální cesta z uzlu u do uzlu v je určena těmi hranami, na nichž je optimální tok nenulový.

Největší lineární podbigraf (největší párování) v bipartitním grafu. Nechtě A, B jsou partity $U(G)$ (tj. $U(G) = A \cup B$ a A, B jsou nezávislé). Sestrojíme síť takto:

- ke grafu přidáme dva nové uzly z, s ,
- ke grafu přidáme nové hrany (z, x) pro všechny uzly $x \in A$,
- ke grafu přidáme nové hrany (y, s) pro všechny uzly $y \in B$,
- propustnosti všech hran sítě jsou rovny jedné.

V takto sestavené síti hledáme (celočíselný) maximální tok ze z do s .



Je zřejmé, že největší párování v původním grafu je určeno těmi jeho hranami, na nichž má nalezený maximální tok nenulovou hodnotu.

Generická hodnota matice je, jak víme z předmětu TGD1, rovna počtu hran největšího lineárního podbigrafu (největšího párování) v bigrafu dané matice. K jejímu určení lze tedy použít předchozí převody na toky.

2 Lineární programování

Lineární programování má, jak uvidíme, v úlohách diskretní optimalizace význačnou roli, z hlediska koncepce tohoto předmětu však je prostředkem, nikoliv cílem. Proto se v této kapitole budeme více než v jiných odkazovat v důkazech na příklady či geometrický názor.

Začneme dvěma motivačními příklady.

Příklad. Úloha o dietě.

Máme několik druhů potravin a chceme z nich vytvořit (jen teoreticky) krmnou směs. Známe:

- obsah i -té složky (i -tého vitamínu) v j -té potravine - tato čísla označíme a_{ij} , lze je zapsat do matice \mathbf{A} typu m/n :

	brambory	kukuřice
vitamín A	a_{11}	a_{12}	a_{1n}
vitamín B	a_{21}	a_{22}	a_{2n}
⋮	⋮	⋮	⋮
vitamín K	a_{m1}	a_{m2}	a_{mn}

- cenu c_j jednotky j -té potraviny - ceny c_j tvoří vektor $\mathbf{c} \in \mathbf{R}_n$,
- požadovaný minimální obsah i -tého vitamínu ve směsi - čísla b_i tvoří vektor $\mathbf{b} \in \mathbf{R}_m$.

Neznámými v úloze jsou čísla x_i , mající význam množství i -té potraviny ve směsi. Čísla x_i tedy musí dále splňovat následující podmínky:

$$\begin{aligned} a_{11}x_1 + a_{21}x_2 + \dots + a_{1n}x_n &\geq b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &\geq b_2 \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n &\geq b_m \end{aligned}$$

Je-li $\mathbf{x} \in \mathbf{R}_n$ vektor čísel x_i , pak tyto podmínky lze zapsat maticově ve tvaru $\mathbf{A} \mathbf{x} \geq \mathbf{b}$.

Dále pro každé i platí $x_i \geq 0$.

Další podmínkou je skutečnost, že takovou směs chceme pořídit co nejlevněji. Chceme tedy (na množině všech vektorů, splňujících předchozí podmínky) minimalizovat výraz $c_1x_1 + \dots + c_nx_n$, tj. hledáme $\min \mathbf{c}^T \mathbf{x}$. Takovou úlohu můžeme stručně zapsat v následujícím tvaru:

$$\begin{aligned} \min \mathbf{c}^T \mathbf{x} \\ \mathbf{A} \mathbf{x} &\geq \mathbf{b} \\ \mathbf{x} &\geq \mathbf{0} \end{aligned}$$

Příklad. Pokud bychom uvažovali, že ve směsi musí být přesně tolik vitamínů, kolik je předepsáno vektorem \mathbf{b} , bude možno výše uvedenou úlohu přepsat do tvaru, v němž soustava podmínek bude mít tvar $\mathbf{A} \mathbf{x} = \mathbf{b}$.

Jinou (a přirozenější) úlohou s podmínkami typu rovnosti je dopravní problém, tj. úloha optimálního toku. V této úloze hledáme

$$\min \sum_{(i,j) \in H(\vec{C})} c_{ij} x_{ij}$$

za podmínek

$$\sum_j x_{ij} - \sum_j x_{ji} = a_i \quad \forall i.$$

a samozřejmě $x_{ij} \geq 0$. Přesný maticový zápis zatím odložíme, zatím si pouze povšimneme toho, že vznikne něco ve smyslu

$$\begin{aligned} \min \mathbf{c}^T \mathbf{x} \\ \mathbf{A} \mathbf{x} = \mathbf{b} \\ \mathbf{x} \geq \mathbf{0} \end{aligned}$$

V praxi se vyskytují i úlohy, v nichž jsou podmínky obou typů, tedy rovnosti i nerovnosti. Nyní zformulujeme zadání úlohy lineárního programování (dále jen LP) přesně.

2.1 Tři základní formulace úlohy lineárního programování

1. Obecná úloha.

Nechť \mathbf{A} je reálná matice typu m/n , \mathbf{N} je podmnožina množiny sloupcových indexů $\{1, \dots, n\}$ matice \mathbf{A} , $\bar{\mathbf{N}}$ je její doplněk, \mathbf{M} je podmnožina množiny řádkových indexů $\{1, \dots, m\}$ matice \mathbf{A} , $\bar{\mathbf{M}}$ je její doplněk, \mathbf{a}_i^T jsou řádky matice \mathbf{A} ($i = 1, \dots, m$), \mathbf{b} je reálný vektor délky m , \mathbf{c} je reálný vektor délky n .

Úkolem je nalézt vektor $\mathbf{x} \in \mathbf{R}^n$, který minimalizuje výraz $\mathbf{c}^T \mathbf{x}$ za podmínek

$$\begin{aligned} \mathbf{a}_i^T \mathbf{x} &= b_i && \text{pro } i \in \mathbf{M}, \\ \mathbf{a}_i^T \mathbf{x} &\geq b_i && \text{pro } i \in \bar{\mathbf{M}}, \\ x_j &\geq 0 && \text{pro } j \in \mathbf{N}, \\ x_j &\in \mathbf{R} && \text{pro } j \in \bar{\mathbf{N}}. \end{aligned}$$

2. Kanonický tvar úlohy.

Úkolem je nalézt vektor $\mathbf{x} \in \mathbf{R}^n$, který minimalizuje výraz $\mathbf{c}^T \mathbf{x}$ za podmínek $\mathbf{A} \mathbf{x} \geq \mathbf{b}$, $\mathbf{x} \geq \mathbf{0}$.

3. Standardní tvar úlohy.

Úkolem je nalézt vektor $\mathbf{x} \in \mathbf{R}^n$, který minimalizuje výraz $\mathbf{c}^T \mathbf{x}$ za podmínek $\mathbf{A} \mathbf{x} = \mathbf{b}$, $\mathbf{x} \geq \mathbf{0}$.

Tvrzení 2.1. *Uvedené tři základní tvary úlohy LP jsou ekvivalentní.*

Důkaz. Kanonický i standardní tvar úlohy jsou zřejmě speciálními případy obecné úlohy. Je tedy třeba ukázat, že obecnou úlohu lze převést na oba ostatní tvary.

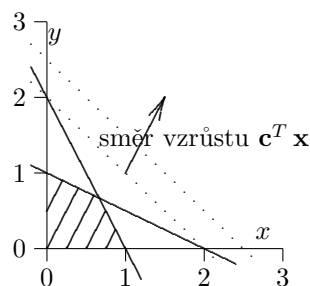
1. Převod obecné úlohy na kanonický tvar: je třeba vyloučit podmínky typu rovnosti a neohrazené proměnné. Rovnosti $\sum_{j=1}^n a_{ij}x_j = b_i$ vyloučíme tak, že je nahradíme dvojicemi nerovností $\sum_{j=1}^n a_{ij}x_j \leq b_i$ a $\sum_{j=1}^n a_{ij}x_j \geq b_i$. Pro každou neohrazenou proměnnou $x_j \in \mathbf{R}$ zavedeme (a dosadíme do podmínek) dvě nové proměnné x_j^+ a x_j^- tak, aby $x_j = x_j^+ - x_j^-$ a aby obě tyto proměnné byly nezáporné. Tyto proměnné se definují předpisem $x_j^+ = \max\{0, x_j\}$, resp. $x_j^- = \max\{0, -x_j\}$.
2. Převod obecné úlohy na standardní tvar: nyní potřebujeme vyloučit všechny podmínky typu nerovnosti a nahradit je rovnostmi. Zavedeme proto nové nezáporné proměnné s_i tak, aby pro nerovnosti typu $\sum a_{ij}x_j \geq b_i$ platilo $\sum a_{ij}x_j - s_i = b_i$. Obdobně nerovnosti $\sum a_{ij}x_j \leq b_i$ nahradíme rovnostmi $\sum a_{ij}x_j + s_i = b_i$. Neohrazené proměnné vyloučíme analogicky. □

Poznámka. V praxi se vyskytují i takové úlohy, v nichž místo minimalizace výrazu $\mathbf{c}^T \mathbf{x}$ hledáme jeho maximum. Vzhledem k tomu, že uvažujeme vektor \mathbf{c} reálný, stačí hledat $\min -\mathbf{c}^T \mathbf{x}$. Dále se může stát, že místo podmínek $\mathbf{A} \mathbf{x} \geq \mathbf{b}$ máme omezení $\mathbf{A} \mathbf{x} \leq \mathbf{b}$. I to není vzhledem k reálnosti matice \mathbf{A} problém. Stačí převést tyto podmínky do tvaru $(-\mathbf{A})\mathbf{x} \geq (-\mathbf{b})$. Při převodu úlohy na standardní tvar je třeba obrátit znaménka v nerovnostech i u nových proměnných s_i .

Příklady. V následujících příkladech (úmyslně jednoduchých) si ukážeme geometrický význam řešení těchto úloh. U těchto jednoduchých úloh je grafické řešení velmi názorné a snadno si uvědomíme geometrickou podstatu řešení úloh LP. Oblast vymezená podmínkami je na každém obrázku příslušné úlohy vyšrafována, tečkovaně jsou naznačeny vrstevnice cenové funkce, jejíž extrém hledáme. Je také naznačen směr růstu (poklesu) ceny u úloh, v nichž hledáme maximum (minimum).

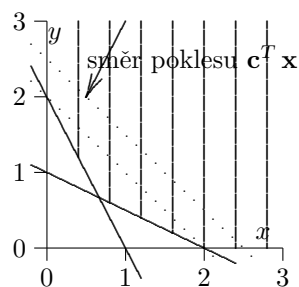
$$\begin{aligned} 1. \quad & \max x + y \\ & x + 2y \leq 2 \\ & 2x + y \leq 2 \\ & x \geq 0 \\ & y \geq 0 \end{aligned}$$

Řešení je zřejmě $x = y = \frac{2}{3}$.



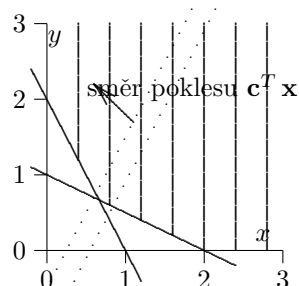
$$\begin{aligned} 2. \quad & \min x + y \\ & x + 2y \geq 2 \\ & 2x + y \geq 2 \\ & x \geq 0 \\ & y \geq 0 \end{aligned}$$

Řešení je zřejmě opět $x = y = \frac{2}{3}$.



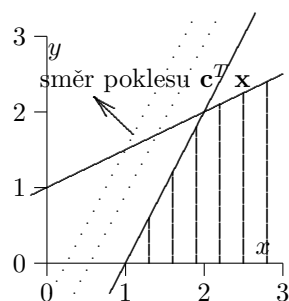
$$\begin{aligned} 3. \quad & \min x - y \\ & 2x + y \geq 2 \\ & x + 2y \geq 2 \\ & x \geq 0 \\ & y \geq 0 \end{aligned}$$

Množina přípustných řešení je ve směru poklesu cenové funkce neomezená, optimum úlohy je „v nekonečnu“, tj. řešení neexistuje.



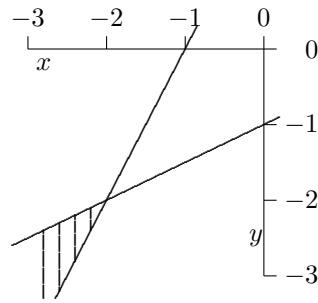
$$\begin{aligned} 4. \quad & \min x - y \\ & 2x - y \geq 2 \\ & x - 2y \geq -2 \\ & x \geq 0 \\ & y \geq 0 \end{aligned}$$

Množina přípustných řešení je neomezená, je však omezená ve směru poklesu $\mathbf{c}^T \mathbf{x}$ (obdobně jako v příkladu 2). Úloha LP má řešení $x = y = 2$.



$$\begin{aligned}
5. \quad & \min x - y \\
& -2x + y \geq 2 \\
& x - 2y \geq 2 \\
& x \geq 0 \\
& y \geq 0
\end{aligned}$$

Neexistuje žádné přípustné řešení, úloha nemá řešení.



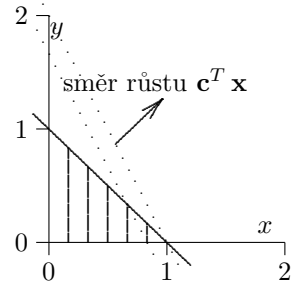
6. Nyní si ukážeme, jak vypadá převod úlohy v kanonickém tvaru na úlohu ve standardním tvaru.

Kanonický tvar:

$$\begin{aligned}
& \max 2x + y \\
& x + y \leq 1 \\
& x \geq 0 \\
& y \geq 0
\end{aligned}$$

Optimální řešení této kanonické úlohy LP je

$$x = 1, y = 0.$$

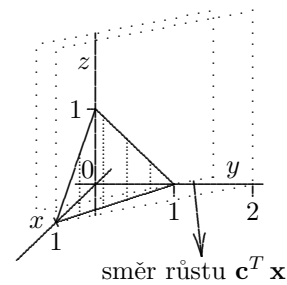


Totéž ve standardním tvaru:

$$\begin{aligned}
& \max 2x + y + 0z \\
& x + y + z = 1 \\
& x \geq 0 \\
& y \geq 0 \\
& z \geq 0
\end{aligned}$$

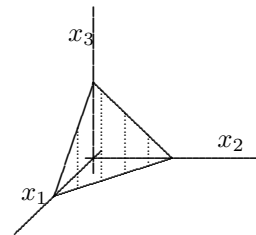
Optimální řešení této standardní úlohy LP je

$$x = 1, y = 0, z = 0.$$



Je vidět, že kanonická úloha je kolmým průmětem standardní úlohy LP.

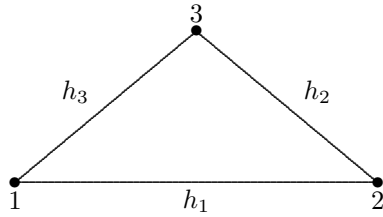
$$\begin{aligned}
7. \quad & \min c_1 x_1 + c_2 x_2 + c_3 x_3 \\
& x_1 + x_2 + x_3 = 2 \\
& x_1 \geq 0 \\
& x_2 \geq 0 \\
& x_3 \geq 0
\end{aligned}$$



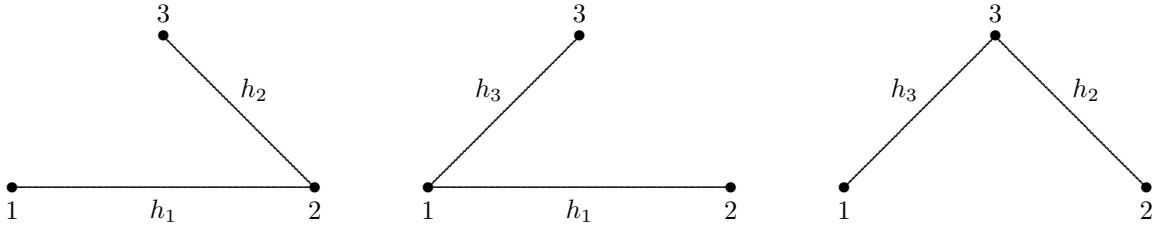
Všechny body $x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$, odpovídající přípustným řešením, vyplní trojúhelník (viz obrázek).

Je zřejmé, že minimum lineární funkce $c_1 x_1 + c_2 x_2 + c_3 x_3$ bude ležet v některém z vrcholů trojúhelníka - tam jsou některá $x_i = 0$.

Příklad. Ukažme si ještě jeden motivační příklad. Hledáme minimální kostru v ohodnoceném grafu:



Tento graf má tři kostry, a to:

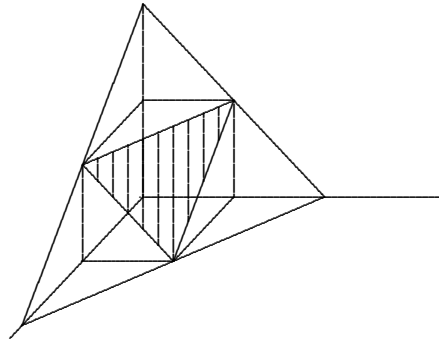


Položíme $x_j = \begin{cases} 1, & \text{jestliže hrana } h_j \text{ leží v kostře,} \\ 0, & \text{jestliže } h_j \text{ v kostře neleží.} \end{cases}$

Napišeme-li podmínky

$$\begin{aligned} x_1 + x_2 + x_3 &= 2, \\ x_1 &\geq 0, x_2 \geq 0, x_3 \geq 0, \\ x_1 &\leq 1, x_2 \leq 1, x_3 \leq 1, \end{aligned}$$

pak přípustná řešení vyplní trojúhelník na obrázku vpravo a naše kostry jsou jeho vrcholy. Hledání minimální kostry je tedy minimalizace lineární funkce $w_1x_1 + w_2x_2 + w_3x_3$, kde w_i je ohodnocení hrany. Minimum této funkce musí být ve vrcholu, jak již bylo zmíněno.



2.2 Simplexový algoritmus

Uvažujme úlohu LP ve standardní formě

$$\begin{aligned} \min \mathbf{c}^T \mathbf{x} \\ \mathbf{Ax} &= \mathbf{b} \\ x_i &\geq 0, \end{aligned}$$

kde \mathbf{A} je typu m/n a platí $m < n$. Dále budeme vždy předpokládat, že hodnost matice \mathbf{A} je rovna m (tento předpoklad prakticky nesnižuje obecnost).

Definice 2.1. Je-li $\mathcal{B} = \{A_{j_1}, A_{j_2}, \dots, A_{j_m}\}$ množina m lineárně nezávislých sloupců matice \mathbf{A} , pak vektor $\mathbf{x} \in \mathbf{R}^n$, splňující podmínky

- $x_j = 0$ pro $A_j \notin \mathcal{B}$,
- x_k je k -tá složka vektoru $\mathbf{B}^{-1}\mathbf{b}$, kde $\mathbf{B} = [A_{j_i}]_{i=1}^m$ pro $k = 1 \dots m$,

se nazývá bazické řešení, příslušné množině \mathcal{B} .

Poznámka. Bazické řešení dostaneme takto:

- 1) zvolíme množinu \mathcal{B} lineárně nezávislých sloupců matice \mathbf{A} (tj. některou bázi sloupcového prostoru matice \mathbf{A}),
- 2) položíme všechny složky vektoru \mathbf{x} pro sloupce, které nejsou v množině \mathcal{B} , rovny nule,
- 3) řešíme vzniklou soustavu rovnic s regulární maticí (jež má právě jedno řešení).

Definice 2.2. Bazické řešení, pro které platí $x_j \geq 0$, $j = 1 \dots n$, se nazývá přípustné bazické řešení.

Věta 2.1. Optimální řešení úlohy LP je některé z jejích přípustných bazických řešení.

Důkaz. Nebudeme provádět, geometricky je z předchozích příkladů zřejmý. □

Poznámka. Platí i opačné tvrzení: pro každé bazické řešení existuje vektor \mathbf{c} takový, že toto řešení minimalizuje výraz $\mathbf{c}^T \mathbf{x}$.

Důkaz. Položíme-li

$$\begin{aligned} c_j &= 0 \text{ pro } A_j \in \mathcal{B}, \\ c_j &= 1 \text{ pro } A_j \notin \mathcal{B}, \end{aligned}$$

pak pro naše bazické řešení je $\mathbf{c}^T \mathbf{x} = 0$; v ostatních případech je tento výraz vždy kladný. □

Při hledání minima cenové funkce $\mathbf{c}^T \mathbf{x}$ bychom tedy mohli probírat všechny m -prvkové podmnožiny sloupců matice \mathbf{A} , kterých je $\binom{n}{m}$, nalézt všechna bazická řešení a vybrat z nich optimální. V praxi by byl tento postup časově velmi náročný. Je třeba nějak zorganizovat přechod od jednoho přípustného bazického řešení ke druhému. Tento postup je znám jako *simplexový algoritmus*.

Celou myšlenku tohoto algoritmu si ukážeme na následujícím příkladu. Uvažujme úlohu ve standardním tvaru

$$\begin{aligned} \min \mathbf{c}^T \mathbf{x} \\ \mathbf{A} \mathbf{x} &= \mathbf{b} \\ x_i &\geq 0, i = 1 \dots n \end{aligned}$$

Představme si tuto úlohu jako úlohu o dietě s podmínkami na obsah vitaminů typu rovnosti. Předpokládejme rovněž, že jsme již našli nějaké výchozí přípustné bazické řešení (což samo o sobě může být problém, ale k tomu se vrátíme později). Příslušné bazické proměnné zapíšeme jako první a rozšířená matice soustavy $\mathbf{A} \mathbf{x} = \mathbf{b}$ nabude tvaru

$$\left[\begin{array}{ccc|ccc|c} a_{11} & a_{12} & \dots & \dots & a_{1n} & & b_1 \\ a_{21} & a_{22} & \dots & \dots & a_{2n} & & b_2 \\ \vdots & \vdots & \dots & \dots & \dots & & \vdots \\ a_{m1} & a_{m2} & \dots & \dots & a_{mn} & & b_m \end{array} \right],$$

kde prvních m sloupců odpovídá sloupcům bazických proměnných.

Přípustné bazické řešení $[x_1^1, x_2^1, \dots, x_m^1, 0, \dots, 0]^T$ obdržíme po Gauss-Jordanově eliminaci v této matici podle podmatice tvořené sloupci bazických proměnných.

$$\left[\begin{array}{cccc|ccc|c} 1 & 0 & \dots & 0 & a_{1\ m+1}^1 & \dots & a_{1s}^1 & \dots & a_{1n}^1 & x_1^1 \\ 0 & 1 & \vdots & 0 & a_{2\ m+1}^1 & \dots & a_{2s}^1 & \dots & a_{2n}^1 & x_2^1 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \dots & \dots & \dots & \vdots & \vdots \\ 0 & \dots & 0 & 1 & a_{m\ m+1}^1 & \dots & a_{ms}^1 & \dots & a_{mn}^1 & x_m^1 \end{array} \right]$$

Přechod k jinému přípustnému bazickému řešení uskutečníme náhradou některé bazické proměnné jinou, nebazickou. Není však jedno, kterou bazickou proměnnou z báze odstraníme a kterou nebazickou naopak do báze přidáme. Máme tedy dva dílčí problémy:

- a) kterou nebazickou proměnnou dát do báze,
- b) kterou bazickou proměnnou z báze vyjmout?

ad a): Uvažujme s -tou nebazickou proměnnou. Její jednotka stojí c_s a tato jednotka by nahradila a_{1s} jednotek první proměnné, a_{2s} jednotek druhé proměnné, atd., což celkem stojí $c_1 a_{1s} + \dots + c_m a_{ms}$ peněz. Pokud je tato suma větší než cena jednotky s -té proměnné, tak se s -tou nebazickou proměnnou vyplatí dát do báze. To tedy nastane, pokud platí nerovnost (horní index 1 zde pro jednoduchost nepíšeme):

$$c_s < c_1 a_{1s} + c_2 a_{2s} + \dots + c_m a_{ms}.$$

S formálním maticovým zápisem je trochu potíž, protože cen není m , ale n - pro nebazické proměnné, kterých je ve směsi nulové množství, nejsou na pravé straně nerovnosti příslušné sčítance uvedeny. Zavedeme vektor $\tilde{\mathbf{a}}_s$, který má význam s -tého sloupce matice \mathbf{A} , doplněného nulami na pozicích, odpovídajících nebazickým proměnným, tj.

$$\tilde{\mathbf{a}}_s = [a_{1s}, a_{2s}, \dots, a_{ms}, 0, \dots, 0]^T.$$

(Pokud by bazické proměnné nebyly v tabulce na prvních místech, byly by nuly na příslušných pozicích v tomto vektoru.)

Nyní tyto poznatky shrneme:

PRAVIDLO 1. Jestliže pro s -tý sloupec platí nerovnost

$$\mathbf{c}^T \tilde{\mathbf{a}}_s - c_s > 0,$$

pak s -tou proměnnou zahrneme do nové báze.

Poznámka. Hodnota $\mathbf{c}^T \tilde{\mathbf{a}}_s - c_s$ se nazývá *relativní cena* s -té nebazické proměnné vzhledem k dané bázi.

ad b): Kterou bazickou proměnnou z báze vyjmout? Když použijeme x_s jednotek s -té proměnné, tak ve směsi zůstane

$$\begin{array}{ll} x_1 - a_{1s}x_s & \text{jednotek 1. bazické proměnné,} \\ x_2 - a_{2s}x_s & \text{jednotek 2. bazické proměnné,} \\ \vdots & \\ x_m - a_{ms}x_s & \text{jednotek } m\text{-té bazické proměnné,} \end{array}$$

přičemž všechny tyto výrazy musí být nezáporné.

Platí tedy:

$$\begin{array}{llll} x_1 - a_{1s}x_s & \geq 0 & \Rightarrow & x_s \leq \frac{x_1}{a_{1s}} \\ x_2 - a_{2s}x_s & \geq 0 & \Rightarrow & x_s \leq \frac{x_2}{a_{2s}} \\ \dots & & & \dots \\ x_m - a_{ms}x_s & \geq 0 & \Rightarrow & x_s \leq \frac{x_m}{a_{ms}} \end{array}$$

Chceme použít nové bazické proměnné co nejvíce (vyplatí se, zmenšuje hodnotu cenové funkce). Největší možné množství x_s této s -té proměnné (nové bazické) je rovno nejmenšímu z výrazů na pravé straně nerovností. Při takovém x_s bude proměnná, kterou vyjímáme z báze, novou proměnnou plně nahrazena.

Opět shrneme tuto úvahu:

PRAVIDLO 2. Zahrnujeme-li do nové báze proměnnou x_s , tak jí nahradíme tu bazickou proměnnou x_i , pro kterou je výraz $\frac{x_i}{a_{is}}$ minimální kladný.

Dá se obecně dokázat, že tato dvě pravidla (intuitivně zřejmá) skutečně vedou k optimálnímu přípustnému bazickému řešení. Postup, založený na těchto pravidlech, se (z historických důvodů) nazývá *simplexová metoda* (nebo též *simplexový algoritmus*).

Příklad.

$$\begin{aligned} \min & x_1 + 6x_2 - 7x_3 + x_4 + 5x_5 \\ & 5x_1 - 4x_2 + 13x_3 - 2x_4 + x_5 = 20 \\ & x_1 - x_2 + 5x_3 - x_4 + x_5 = 8 \\ & x_i \geq 0, i = 1 \dots 5 \end{aligned}$$

Vektor cen \mathbf{c} je vektor $\mathbf{c}^T = [1, 6, -7, 1, 5]$. Soustavu zapíšeme do následující tabulky T_0 :

5	-4	13	-2	1	20
1	-1	5	-1	1	8

Potřebujeme nalézt nějaké výchozí přípustné bazické řešení. To může být obecně problém, uvidíme později (v odstavci 2.3), jak jej obecně řešit. V našem případě budeme postupovat takto: od 1. řádku odečteme druhý, od 2. řádku odečteme čtvrtinu prvního, 1. řádek dělíme čtyřmi. Tabulka T_0 se změní do podoby T_1 :

1	$-\frac{3}{4}$	2	$-\frac{1}{4}$	0	3
0	$-\frac{1}{4}$	3	$-\frac{3}{4}$	1	5

Ve sloupci napravo jsou kladná čísla, nalezené bazické řešení $\mathbf{x} = [3, 0, 0, 0, 5]^T$ je tedy přípustné. Pro další postup bude vhodné připsat k tabulce ceny jednotlivých proměnných. Nahoru ceny všech proměnných, vlevo ceny příslušných bazických proměnných, tj. v tomto případě ceny první a páté proměnné.

	1	6	-7	1	5	
1	1	$-\frac{3}{4}$	2	$-\frac{1}{4}$	0	3
5	0	$-\frac{1}{4}$	3	$-\frac{3}{4}$	1	5

Nyní můžeme zahájit optimalizaci. Pro jednotlivé sloupce všech proměnných vyčíslíme výrazy $\mathbf{c}^T \tilde{\mathbf{a}}_s - c_s$, pro poslední sloupec celkovou cenu. Vše opět připojíme k tabulce T_1 :

	1	6	-7	1	5	
1	1	$-\frac{3}{4}$	2	$-\frac{1}{4}$	0	3
5	0	$-\frac{1}{4}$	3	$-\frac{3}{4}$	1	5
$\mathbf{c}^T \tilde{\mathbf{a}}_s - c_s$	0	-8	24	-5	0	28

Podle pravidla 1 zařadíme třetí proměnnou do báze (vyplatí se). Ale kterou proměnnou z báze odstranit? Vyčíslíme výrazy $\frac{x_i}{\tilde{a}_{i3}}$ pro všechny bazické proměnné ($i = 1, 5$):

$$\frac{x_1}{\tilde{a}_{13}} = \frac{3}{2}; \quad \frac{x_5}{\tilde{a}_{53}} = \frac{5}{3}; \quad \frac{3}{2} < \frac{5}{3}.$$

Proto, podle pravidla 2, z báze vyjmeleme proměnnou x_1 . Dále budeme postupovat stejným způsobem. Vedoucím prvkem pro Jordanovu eliminaci bude prvek ve sloupci a řádku nové a nahrazované bazické proměnné, tedy prvek na pozici (1, 3) následující tabulky:

1	$-\frac{3}{4}$	2	$-\frac{1}{4}$	0	3
0	$-\frac{1}{4}$	3	$-\frac{3}{4}$	1	5

Po úpravách získáme tvar simplexové tabulky T_2 :

$\frac{1}{2}$	$-\frac{3}{8}$	1	$-\frac{1}{8}$	0	$\frac{3}{2}$
$-\frac{3}{2}$	$\frac{7}{8}$	0	$-\frac{3}{8}$	1	$\frac{1}{2}$

K této tabulce opět přidáme ceny jednotlivých proměnných a vyčíslíme relativní ceny všech proměnných. Tabulka T_2 tak nabude tvaru

	1	6	-7	1	5	
	$\frac{1}{2}$	$-\frac{3}{8}$	1	$-\frac{1}{8}$	0	$\frac{3}{2}$
	$-\frac{3}{2}$	$\frac{7}{8}$	0	$-\frac{3}{8}$	1	$\frac{1}{2}$
	-12	1	0	-2	0	-8

Všimněme si dvou faktů. Jednak nám cena řešení klesla z 28 na -8, skutečně se tedy blížíme k minimu cenové funkce. Dále vidíme, že bychom měli do báze zařadit proměnnou x_2 . Abychom zjistili, kterou proměnnou z báze vyřadit, opět vyčíslíme výrazy $\frac{x_i}{\tilde{a}_{i2}}$ pro $i = 3, 5$:

$$\frac{x_3}{\tilde{a}_{32}} = \frac{\frac{3}{2}}{-\frac{3}{8}} < 0; \quad \frac{x_5}{\tilde{a}_{52}} = \frac{\frac{1}{2}}{\frac{7}{8}} = \frac{4}{7} > 0;$$

Minimální kladný je výraz páté proměnné, kterou podle pravidla 2 vyřadíme z báze. Opět provedeme Jordanovu eliminaci s vedoucím prvkem vyznačeným v následující tabulce tučně:

$\frac{1}{2}$	$-\frac{3}{8}$	1	$-\frac{1}{8}$	0	$\frac{3}{2}$
$-\frac{3}{2}$	$\frac{7}{8}$	0	$-\frac{3}{8}$	1	$\frac{1}{2}$

Po úpravách získáme simplexovou tabulku T_3 :

$-\frac{1}{7}$	0	1	$-\frac{2}{7}$	$\frac{3}{7}$	$\frac{12}{7}$
$-\frac{12}{7}$	1	0	$-\frac{3}{7}$	$\frac{8}{7}$	$\frac{4}{7}$

Po přidání cen jednotlivých proměnných a výpočtu relativních cen přejde tabulka na následující tvar:

	1	6	-7	1	5	
$-\frac{1}{7}$	0	1	$-\frac{2}{7}$	$\frac{3}{7}$	$\frac{12}{7}$	
$-\frac{12}{7}$	1	0	$-\frac{3}{7}$	$\frac{8}{7}$	$\frac{4}{7}$	
$-\frac{72}{7}$	0	0	$-\frac{11}{7}$	$-\frac{8}{7}$	$-\frac{60}{7}$	

Řádek relativních cen nemá žádný kladný prvek, není tedy co nahrazovat, podle následující věty jsme našli optimum. Optimální cena úlohy je $-\frac{60}{7}$; optimální řešení má tvar $\mathbf{x} = [0, \frac{4}{7}, \frac{12}{7}, 0, 0]^T$.

Věta 2.2. Jestliže pro všechny sloupce platí $\mathbf{c}^T \tilde{\mathbf{a}}_s - c_s \leq 0$, pak je toto přípustné bazické řešení optimální.

Takto by šlo metodu prakticky provozovat. Lze však trikem zjednodušit výpočet relativních cen. Podívejme se na tabulku T_1 z předchozího příkladu:

	1	6	-7	1	5	
1	1	$-\frac{3}{4}$	2	$-\frac{1}{4}$	0	3
5	0	$-\frac{1}{4}$	3	$-\frac{3}{4}$	1	5
$\mathbf{c}^T \tilde{\mathbf{a}}_s - c_s$	0	-8	24	-5	0	28

1. Je-li x_s bazická proměnná, pak $\tilde{\mathbf{a}}_s$ je vektor samých nul, jen na s -té pozici bude mít jedničku. Pro relativní cenu každé bazické proměnné tedy platí $\mathbf{c}^T \tilde{\mathbf{a}}_s - c_s = c_s - c_s = 0$.
2. Co by se stalo, kdybychom v horním (nultém) řádku čísla ve sloupcích bazických proměnných vynulovali Jordanovou eliminací? Především si uvědomme, že při výpočtu relativních cen se hodnota c_s odečítá, bude tedy vhodné napsat nultý řádek s opačným znaménkem (ekvivalentní interpretace této úvahy: definujeme proměnnou ceny vztahem $z - \sum c_i x_i = 0$; nultý řádek tabulky je pak řádek pro tuto proměnnou z). Získáme tak tabulku:

	-1	-6	7	-1	-5	0
1	1	$-\frac{3}{4}$	2	$-\frac{1}{4}$	0	3
5	0	$-\frac{1}{4}$	3	$-\frac{3}{4}$	1	5
$\mathbf{c}^T \tilde{\mathbf{a}}_s - c_s$	0	-8	24	-5	0	28

V našem případě vynulujeme ceny u bazických proměnných tím, že k nultému řádku přičteme jednu první a pětkrát druhý.

Obecně: k nultému řádku přičteme c_i -násobek i -tého řádku. Pak hodnoty ve sloupcích bazických proměnných budou rovny nule a ve sloupcích nebazických proměnných vyjdou výrazy $\sum c_i \tilde{a}_{is} - c_s = \mathbf{c}^T \tilde{\mathbf{a}}_s - c_s$, což jsou přesně relativní ceny. Vypočteme je tedy přímo v tabulce při Jordanově eliminaci. V nultém řádku ve sloupci pravých stran přitom vyjde číslo $\sum c_i x_i$, kde x_i jsou hodnoty složek přípustného bazického řešení. A to je cena příslušného řešení.

(To vše ale platí pouze v případě, že sloupec s -té bazické proměnné je již po eliminaci, tj. je ve tvaru $[0, 0, \dots, 0, \mathbf{1}, 0, \dots, 0]^T$; je nutné si navíc uvědomit, že k nultému řádku lze jen přičítat či odečítat násobky ostatních řádků, ale nultý řádek samotný násobit číslem nelze, neboť by se změnila cena proměnných).

Příklad.

$$\begin{aligned} \min & 3x + y \\ & x + 2y \geq 3 \\ & 2x + y \geq 3 \\ & x \geq 0 \\ & y \geq 0 \end{aligned}$$

Nejprve převedeme tuto úlohu (v kanonickém tvaru) na úlohu ve standardním tvaru:

$$\begin{aligned} \min & 3x + y \\ & x + 2y - u = 3 \\ & 2x + y - v = 3 \\ & x \geq 0, \quad y \geq 0 \\ & u \geq 0, \quad v \geq 0 \end{aligned}$$

Sestavíme výchozí simplexovou tabulku T_0 :

-3	-1	0	0	0
1	2	-1	0	3
2	1	0	-1	3

Nyní je třeba určit výchozí přípustné bazické řešení. I v tomto příkladu se pokusíme problém obejít a zkusit toto řešení uhodnout. Položme (z první rovnice) $x = 3, y = 0, u = 0$. Ze druhé rovnice určíme poslední proměnnou $v = 3$. Bazickými proměnnými tedy budou proměnné x a v . Jordanovou eliminací a přidáním cen jednotlivých proměnných získáme tabulku T_1 :

0	5	-3	0	9
1	2	-1	0	3
0	3	-2	1	3

Mám přípustné bazické řešení $\mathbf{x} = [3, 0, 0, 3]^T$, cena tohoto řešení je 9. Toto řešení není optimální, jak je patrné z uvedené tabulky, do báze přidáme proměnnou y a vyjmemme proměnnou v . Po následné Jordanově eliminaci získáme tabulku T_2 :

0	0	$\frac{1}{3}$	$-\frac{5}{3}$	4
1	0	$\frac{1}{3}$	$-\frac{2}{3}$	1
0	1	$-\frac{2}{3}$	$\frac{1}{3}$	1

Nyní jsme získali přípustné bazické řešení $\mathbf{x} = [1, 1, 0, 0]^T$, jehož cena je 4. Opět se nejedná o optimální řešení, do báze přidáme proměnnou u a vyjmemme x . Jordanovou eliminací získáme tabulku T_3 :

-1	0	0	-1	3
3	0	1	-2	3
2	1	0	-1	3

Obdrželi jsme tak přípustné bazické řešení $\mathbf{x} = [0, 3, 3, 0]^T$. Toto řešení je již optimální, neboť relativní ceny všech nebazických proměnných jsou záporné. Optimální řešení původního problému je $[0, 3]^T$.

Příklad. V následujícím příkladu si ukážeme jednoduchou úlohu, která nemá řešení. Množina přípustných řešení je stejná jako v předchozím příkladu, jediný rozdíl bude v cenové funkci.

$$\begin{aligned} \min x - y \\ x + 2y &\geq 3 \\ 2x + y &\geq 3 \\ x &\geq 0 \\ y &\geq 0 \end{aligned}$$

Nejprve převedeme úlohu do standardního tvaru.

$$\begin{aligned} \min x - y \\ x + 2y - u &= 3 \\ 2x + y - v &= 3 \\ x &\geq 0, \quad y \geq 0 \\ u &\geq 0, \quad v \geq 0 \end{aligned}$$

Vyjdeme ze stejného počátečního přípustného bazického řešení, tedy $\mathbf{x} = [3, 0, 0, 3]^T$ a sestavíme výchozí simplexovou tabulku T_0 :

-1	1	0	0	0
1	2	-1	0	3
2	1	0	-1	3

Jordanovou eliminací, se stejným vedoucím prvkem jako v minulém příkladu, získáme tabulku T_1 :

0	3	-1	0	3
1	2	-1	0	3
0	3	-2	1	3

Mám přípustné bazické řešení $\mathbf{x} = [3, 0, 0, 3]^T$, cena tohoto řešení je 3. Toto řešení není optimální, jak je patrné z uvedené tabulky, do báze přidáme proměnnou y a vyjmeleme proměnnou v . Po následné Jordanově eliminaci získáme tabulku T_2 :

0	0	1	-1	0
1	0	$\frac{1}{3}$	$-\frac{2}{3}$	1
0	1	$-\frac{2}{3}$	$\frac{1}{3}$	1

Nyní jsme získali přípustné bazické řešení $\mathbf{x} = [1, 1, 0, 0]^T$, jehož cena je 0. Opět se nejedná o optimální řešení, do báze přidáme proměnnou u a vyjmeleme x . Jordanovou eliminací získáme tabulku T_3 :

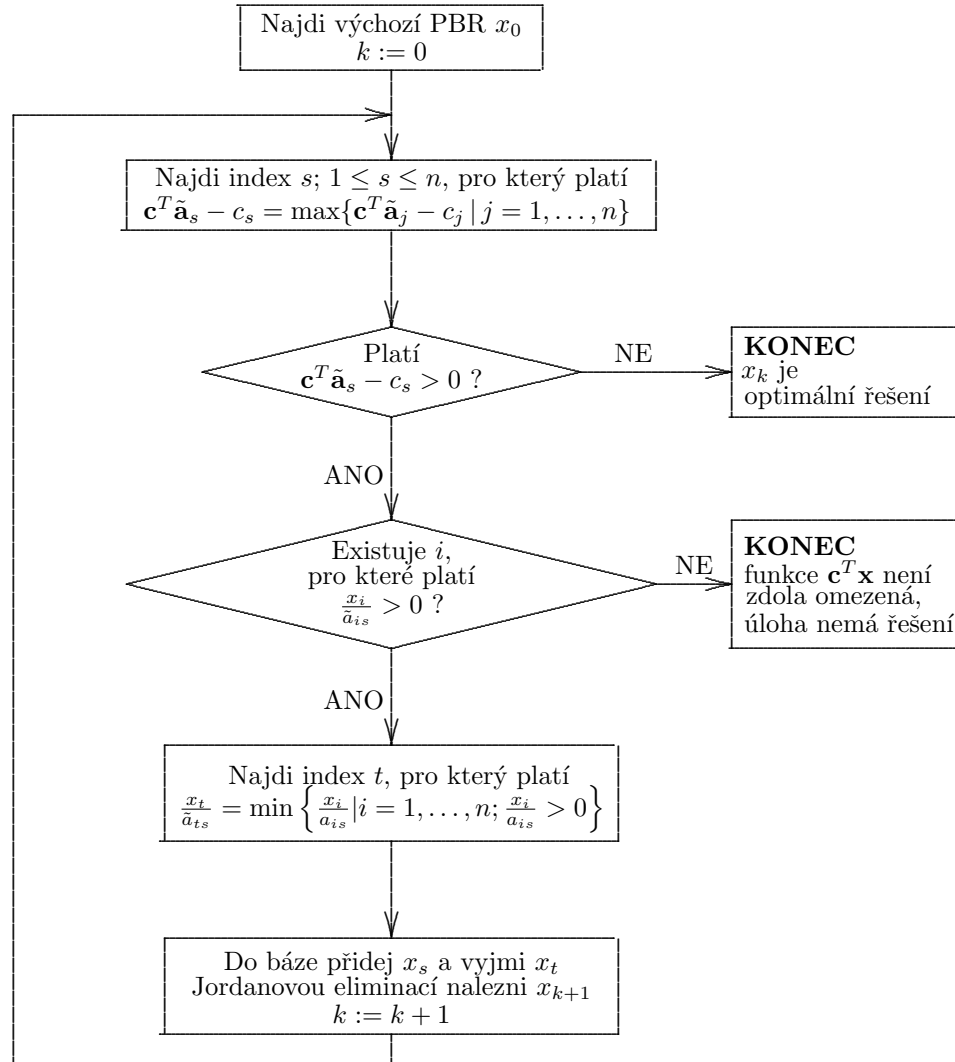
-3	0	0	1	-3
3	0	1	-2	3
2	1	0	-1	3

Přípustné bazické řešení je $\mathbf{x} = [0, 3, 3, 0]^T$. Do báze přidáme proměnnou v , ale pozor, není co z báze vyjmout. Tedy proměnná v nám nenahradí žádnou z bazických proměnných y, u . Množina přípustných řešení není ve směru gradientu cenové funkce omezená množina, tato úloha LP nemá řešení.

Tento příklad ilustruje následující tvrzení (jež nebudeme dokazovat).

Tvrzení 2.2. Jsou-li všechny výrazy $\frac{x_i}{a_{is}}$ záporné, pak funkce $\mathbf{c}^T \mathbf{x}$ není zdola omezená a úloha LP nemá řešení.

V následujícím blokovém schématu simplexového algoritmu je tento případ zahrnut. Pro kratší zápis budeme používat místo pojmu „přípustné bazické řešení“ zkratku „PBR“.



Příklad. Jsou dvě betonárny, označené písmeny X a Y , a tři staveniště, označená čísla 1,2,3, na která se vozí beton. Betonárna X produkuje 14 aut betonu denně, betonárna Y produkuje 6 aut denně. Na staveništi 1 se spotřebuje 8 aut betonu denně, na druhém staveništi 5 aut a na třetím 7 aut betonu denně. Dále jsou dány vzdálenosti jednotlivých stavenišť od obou betonáren, které jsou uvedeny v následující tabulce:

	1	2	3
X	15	7	18
Y	6	3	8

Naším úkolem je nalézt optimální plán rozvozu betonu.

Označme jednotlivá množství podle betonárny, ze které je beton převážěn a staveniště, na kterém se využije, x_1, x_2, x_3 a y_1, y_2, y_3 . Hodnota x_i tedy odpovídá počtu aut převážených z betonárny X na stavbu i (a podobně y_i). Cenová funkce je

$$\min 15x_1 + 7x_2 + 18x_3 + 6y_1 + 3y_2 + 8y_3.$$

Podmínky úlohy popisují, že na každou stavbu se musí celkem dostat přesně potřebné množství betonu denně (první tři rovnice), a všechen odvezený beton z betonáren musí být roven výrobní kapacitě betonáren (další dvě rovnice). Přitom samozřejmě veškerá převážená množství betonu jsou nezáporná.

$$\begin{aligned} x_1 + y_1 &= 8 \\ x_2 + y_2 &= 5 \\ x_3 + y_3 &= 7 \\ x_1 + x_2 + x_3 &= 14 \\ y_1 + y_2 + y_3 &= 6 \\ x_i \geq 0; y_i \geq 0 \quad i &= 1, 2, 3 \end{aligned}$$

Tuto úlohu zapíšeme do následující tabulky:

1	0	0	1	0	0	8
0	1	0	0	1	0	5
0	0	1	0	0	1	7
1	1	1	0	0	0	14
0	0	0	1	1	1	6

Protože celkové množství odvezeného betonu z betonáren se rovná celkovému množství betonu přivezeného na stavby, jsou rovnice lineárně závislé. Lze tedy vypustit první řádek, čímž se řešení úlohy nezmění. Získáme tak tabulku

0	1	0	0	1	0	5
0	0	1	0	0	1	7
1	1	1	0	0	0	14
0	0	0	1	1	1	6

Nyní je třeba nalézt výchozí PBR. To zatím neumíme, opět se tedy pokusíme řešení uhadnout. Nejprve přehodíme řádky a získáme tabulku

1	1	1	0	0	0	14
0	1	0	0	1	0	5
0	0	1	0	0	1	7
0	0	0	1	1	1	6

Nyní provedeme Jordanovu eliminaci a po přidání nultého řádku cen proměnných a Jordanově eliminaci i tohoto řádku získáme tabulku T_1 :

0	0	0	0	-5	1	227
1	0	0	0	-1	-1	2
0	1	0	0	1	0	5
0	0	1	0	0	1	7
0	0	0	1	1	1	6

Toto řešení je evidentně přípustným bazickým řešením této úlohy, není však optimální. Některá z relativních cen je totiž kladná. Tu přidáme do báze a po určení minima z výrazů $\frac{x_i}{a_{i,s}}$ určíme, kterou proměnnou z báze vyjmeme. Na tuto tabulku provedeme Jordanovu eliminaci s vedoucím prvkem, který je v tabulce T_0 vyznačen tučně. Po Jordanově eliminaci získáme tabulku T_2 :

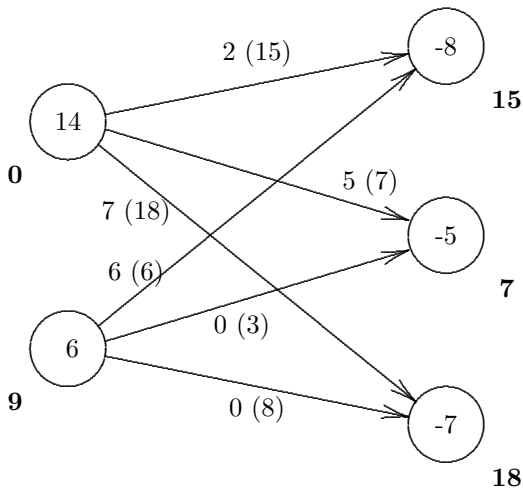
0	0	0	-1	-6	0	221
1	0	0	1	0	0	8
0	1	0	0	1	0	5
0	0	1	-1	-1	0	1
0	0	0	1	1	1	6

Získali jsme přípustné bazické řešení ve tvaru $\mathbf{x} = [8, 5, 1, 0, 0, 6]^T$. Jak je vidět ze simplexové tabulky T_2 , je toto řešení optimální, neboť relativní ceny všech nebazických proměnných jsou záporné.

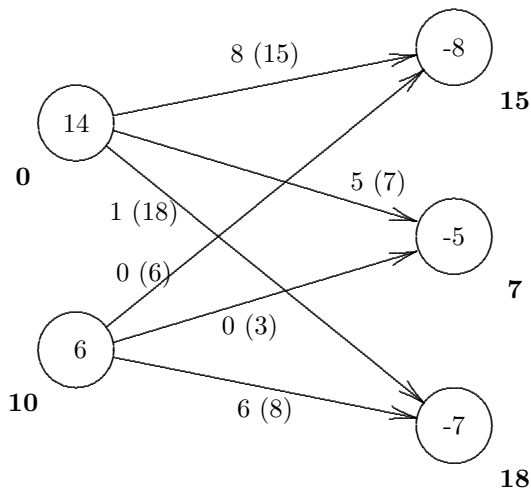
Povšimněme si malé zajímavosti: v optimálním rozvozním plánu je $y_1 = y_2 = 0$. To znamená, že při optimálním rozvozu se na dvě nejkratší vzdálenosti (3 a 6 km) nic nevozí.

Nyní si ukážeme, že tuto úlohu lze také řešit aparátem optimálního toku v síti. Sestrojíme nejprve síť podle zadání úlohy. Zdroje budou představovat obě betonárny, jejich intenzity budou denní produkce betonu v každé z betonáren. Stoky síť budou představovat staveniště, jejich intenzity budou rovny denní spotřebě betonu. Hrany budou vést z každého zdroje do každého stoku. Ceny hran budou dány příslušnými vzdálenostmi, propustnosti uvažujeme nekonečné. Význam čísel u hran na obrázcích je $x_{i,j}(c_{i,j})$ (nekonečné propustnosti neuvádíme).

Výchozí tok:



Potenciály jednotlivých uzlů jsou zapsány tučně jako uzlové ohodnocení. Je patrné, že na hraně vedoucí ze zdroje s intenzitou 6 do stoku s intenzitou -7 není splněna podmínka věty o potenciálech (věta 1.3). Je tedy třeba podél polocyklu daného hranami s intenzitami 14, -8 , 6, -7 , 14 zvýšit tok. Největší možné množství, které můžeme v daném rezervním polocyklu přidat, je 6. Získáme tím nový tok:



Tentokrát již v celé síti existují potenciály, vyhovující na všech hranách. Tím je ukázáno, že daný tok (tj. plán rozvozu betonu) je optimální.

2.3 Dvofázová metoda

Zbývá nám odpovědět na otázku, jak nalézt výchozí PBR, tj. jak nalézt nějaké nezáporné řešení soustavy $\mathbf{Ax} = \mathbf{b}$. Metodu řešení si ukážeme na následujícím příkladu.

Příklad. Hledáme nezáporné řešení soustavy

$$\begin{aligned}x_1 - x_3 + 4x_4 &= 3 \\2x_1 - x_2 &= 3 \\3x_1 - 2x_2 - x_4 &= 1\end{aligned}$$

Napišme si *pomocnou úlohu* s *umělými proměnnými* y_1, y_2, y_3 :

$$\begin{array}{cccccc}x_1 & & -x_3 & +4x_4 & +y_1 & & = & 3 \\2x_1 & -x_2 & & & & +y_2 & & = & 3 \\3x_1 & -2x_2 & & -x_4 & & & +y_3 & = & 1\end{array}$$

a řešme úlohu $\min y_1 + y_2 + y_3$, $x_i \geq 0$, $y_i \geq 0$ (tj. v této pomocné úloze mají proměnné x_i nulovou cenu, proměnné y_i mají cenu 1). Jak je patrné, výchozí bázi nemusíme hledat, je již složena ze sloupců umělých proměnných.

Tato pomocná úloha

- určitě má PBR: $\mathbf{x} = [0, 0, 0, 0, 3, 3, 1]^T$, takže nemusíme hledat výchozí PBR;
- původní soustava má nezáporné řešení právě když pomocná úloha má optimální řešení, pro něž platí $y_i = 0$, $i = 1, 2, 3$.

Tímto obecně získáme tzv. *dvofázovou metodu* řešení úlohy lineárního programování ve standardním tvaru:

$$\begin{aligned}\min \mathbf{c}^T \mathbf{x} \\ \mathbf{Ax} &= \mathbf{b} \\ x_i &\geq 0.\end{aligned}$$

- 1. fáze.** Zavedeme umělé proměnné $\mathbf{y} = [y_1, \dots, y_m]^T$, označíme \mathbf{o} nulový vektor z prostoru \mathbf{R}_n , a $\mathbf{v} = [1, 1, \dots, 1]^T \in \mathbf{R}_m$. Řešíme pomocnou úlohu

$$\begin{aligned}\min [\mathbf{o}^T \mid \mathbf{v}^T] \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \\ [\mathbf{A} \mid \mathbf{I}] \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} &= \mathbf{b} \\ \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} &\geq \mathbf{0}\end{aligned}$$

Tato úloha má určitě PBR, umíme ji tedy řešit. Dále platí následující tvrzení.

Tvrzení 2.3. *Původní úloha má PBR právě když pomocná úloha má optimální řešení, které splňuje podmínku $y_1 = \dots = y_m = 0$.*

Z tohoto optimálního řešení $\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}$ je pak část \mathbf{x} výchozím PBR pro řešení původní úlohy.

- 2. fáze.** S takto nalezeným výchozím PBR \mathbf{x} řešíme původní úlohu simplexovým algoritmem.

Poznámka. Pokud původní úloha je v kanonickém tvaru, tak se hledání PBR dá poněkud zjednodušit (viz např. [12], odst. 4.8).

Příklad. Mějme úlohu ve standardním tvaru

$$\begin{aligned} \min & 2x_1 + 4x_2 + x_3 + x_4 + 2x_5 \\ & x_1 + 2x_2 - x_3 + x_4 + 2x_5 = 8 \\ & 2x_1 + x_2 + x_4 + 2x_5 = 12 \\ & x_1 + x_2 + 2x_3 - x_5 = 4 \\ & x_i \geq 0, i = 1 \dots 5 \end{aligned}$$

Sestavíme následující tabulku:

-2	-4	-1	-1	-2	0
1	2	-1	1	2	8
2	1	0	1	2	12
1	1	2	0	-1	4

0	0	0	0	0	-1	-1	-1	0
1	2	-1	1	2	1	0	0	8
2	1	0	1	2	0	1	0	12
1	1	2	0	-1	0	0	1	4

4	4	1	2	3	0	0	0	24
1	2	-1	1	2	1	0	0	8
2	1	0	1	2	0	1	0	12
1	1	2	0	-1	0	0	1	4

T_2 :

0	0	-7	2	7	0	0	-4	8
0	1	-3	1	3	1	0	-1	4
0	-1	-4	1	4	0	1	-2	4
1	1	2	0	-1	0	0	1	4

T_4 :

0	0	0	0	0	-1	-1	-1	0
0	1	0	$\frac{1}{7}$	0	$\frac{4}{7}$	$-\frac{3}{7}$	$\frac{2}{7}$	$\frac{4}{7}$
0	0	-1	$\frac{2}{7}$	1	$\frac{1}{7}$	$\frac{1}{7}$	$-\frac{3}{7}$	$\frac{8}{7}$
1	0	1	$\frac{1}{7}$	0	$-\frac{3}{7}$	$\frac{4}{7}$	$\frac{2}{7}$	$\frac{32}{7}$

Ve druhé fázi použijeme toto PBR jako výchozí v původní úloze. Nejprve prohodíme vhodné řádky a získáme výchozí tabulku druhé fáze T_0 :

V tomto případě neumíme na první pohled uhádnout výchozí PBR. S tímto se již ale umíme vyrovnat, použijeme tedy dvoufázovou metodu. Sestavíme tabulku s umělými proměnnými a novými cenami.

Jordanova eliminace v matici popsané tabulkou není potřebná, je však nutno vynulovat relativní ceny bazických proměnných (umělých). Tím získáme simplexovou tabulku T_1

Nalezené řešení zřejmě není optimální, tučně vyznačený prvek je vedoucím prvkem pro následující Jordanovu eliminaci. Tento postup budeme opakovat až do té doby, než nalezneme optimální řešení této pomocné úlohy. Následující tabulky $T_2 \dots T_4$ ukazují tyto jednotlivé kroky.

T_3 :

0	$\frac{7}{4}$	0	$\frac{1}{4}$	0	0	$-\frac{7}{4}$	$-\frac{1}{2}$	1
0	$\frac{7}{4}$	0	$\frac{1}{4}$	0	1	$-\frac{3}{4}$	$\frac{1}{2}$	1
0	$-\frac{1}{4}$	-1	$\frac{1}{4}$	1	0	$\frac{1}{4}$	$-\frac{1}{2}$	1
1	$\frac{3}{4}$	1	$\frac{1}{4}$	0	0	$\frac{1}{4}$	$\frac{1}{2}$	5

Relativní ceny všech nebazických proměnných jsou nyní záporné, máme tedy optimální řešení pomocné úlohy a zároveň výchozí PBR původní úlohy $\mathbf{x}_0 = [\frac{32}{7}, \frac{4}{7}, 0, 0, \frac{8}{7}]^T$. Pokud by se v optimu vyskytla v bázi nějaká umělá proměnná, znamenalo by to, že původní úloha nemá žádné PBR.

-2	-4	-1	-1	-2	0
1	0	1	$\frac{1}{7}$	0	$\frac{32}{7}$
0	1	0	$\frac{1}{7}$	0	$\frac{4}{7}$
0	0	-1	$\frac{2}{7}$	1	$\frac{8}{7}$

Je třeba vynulovat relativní ceny u bazických proměnných. Výsledek je uveden v následující simplexové tabulce T_1 .

0	0	-1	$\frac{3}{7}$	0	$\frac{96}{7}$
1	0	1	$\frac{1}{7}$	0	$\frac{32}{7}$
0	1	0	$\frac{1}{7}$	0	$\frac{4}{7}$
0	0	-1	$\frac{2}{7}$	1	$\frac{8}{7}$

Již v této fázi řešení příkladu si můžeme povšimnout podivné věci, kterou je nejednoznačnost při použití pravidla 2. Pro 2. a 3. řádek má výraz $\frac{x_i}{a_{is}}$ stejnou hodnotu. Pokračujeme dále v řešení, tedy opět Jordanovou eliminací, a získáme tabulku T_2 :

0	-3	-1	0	0	12
1	-1	1	0	0	4
0	7	0	1	0	4
0	-2	-1	0	1	0

Nalezli jsme optimální řešení, neboť relativní ceny nebazických proměnných jsou záporné. Řešení je $\mathbf{x} = [4, 0, 0, 4, 0]^T$. Z hlediska použití dvoufázové metody je příklad hotov.

Je však podivné, že nalezené řešení, ač je přípustné i bazické, má jen dvě nenulové složky. To je důsledkem toho, že v původní soustavě je vektor pravých stran lineární kombinací méně než m sloupců matice podmínek (konkrétně zde platí $4a_1 + 4a_4 = \mathbf{b}$). Číslo m nám přitom udává počet lineárně nezávislých podmínek. Taková úloha se nazývá *degenerovaná*. Uvedená situace se v úlohách „z praxe“ vyskytuje poměrně zřídka, ale pokud nastane, mohou být problémy s konečností simplexového algoritmu.

2.4 Degenerované úlohy a zacyklení

Definice 2.3. *Přípustné bazické řešení \mathbf{x} se nazývá nedegenerované, pokud je každá bazická složka tohoto řešení nenulová. V opačném případě říkáme, že PBR je degenerované.*

Úloha LP se nazývá degenerovaná, pokud má aspoň jedno degenerované PBR, jinak se tato úloha nazývá nedegenerovaná.

Věta 2.3. *Pro nedegenerované úlohy je základní simplexový algoritmus konečný.*

Důkaz. Intuitivně - stačí si uvědomit, že u nedegenerovaných úloh jsou relativní ceny ostře větší než nula, takže celková cena po každém provedeném iteračním kroku klesne. Pro konečný počet proměnných je tedy tato metoda konečná. \square

Poznámka. Úloha poznat, zda daná úloha je degenerovaná, je NP-těžká.

Příklad. Mějme dānu následující ůlohu.

$$\begin{aligned} \min \quad & 5x_1 - 4x_2 - 7x_3 - 2x_4 \\ & x_1 + 3x_2 + 3x_3 - 2x_4 \leq 1 \\ & -x_1 + 3x_3 - x_4 \leq 2 \\ & 2x_1 + 4x_2 + 2x_3 - 2x_4 \leq 0 \\ & x_i \geq 0 \end{aligned}$$

Dĭky pomocnĕm promĕnnĕm, kterĕ je nutno zavĕst pŕi pŕevodu ůlohy v kanonickĕm tvaru na ůlohu ve standardnĕm tvaru, mĕme vĕchozĭ PBR a to i s nulovĕmi cenami bazickĕch promĕnnĕch. Tabulku zapĭšeme do následující podoby:

-5	4	7	2	0	0	0	0
1	3	3	-2	1	0	0	1
-1	0	3	-1	0	1	0	2
2	4	2	-2	0	0	1	0

Ve vektoru pravĕch stran se objevila nula. Jednĕ se tedy o degenerovanou ůlohu. Nenĭ to optimĕlnĭ řešenĭ, podle pravidla 2 jsme vybrali za vedoucí prvek následující Jordanovy eliminace tuĕnĕ vyznaĕenĕ prvek.

$-\frac{22}{3}$	-3	0	$\frac{20}{3}$	$-\frac{7}{3}$	0	0	$-\frac{7}{3}$
$\frac{1}{3}$	1	1	$-\frac{2}{3}$	$\frac{1}{3}$	0	0	$\frac{1}{3}$
-2	-3	0	1	-1	1	0	1
$\frac{4}{3}$	2	0	$-\frac{2}{3}$	$-\frac{2}{3}$	0	1	$-\frac{2}{3}$

Zĭskali jsme řešenĭ, kterĕ nenĭ pŕĭpustnĕ, neboť jedna jeho složka je zĕpornĕ. Kde je chyba?

Problĕm je v tom, že Pravidlo 2 (pro vĕbĕr řĕdku – tj. pravidlo „vyber řĕdek, v nĕmž je vĕraz $\frac{x_i}{a_{is}}$ minimĕlnĭ kladnĕ“) platĭ v tĕto podobĕ pouze pro nedegenerovanĕ ůlohy. Obecnĕ pravidlo 2 znĭ takto.

PRAVIDLO 2'. Vedoucí řĕdek následnĕ Jordanovy eliminace je ten, pro nĕjž se nabĕvĕ minimum $\min \left\{ \frac{x_i}{\tilde{a}_{is}} \mid \tilde{a}_{is} > 0 \right\}$.

Pro nedegenerovanĕ ůlohy je $x_i > 0$ vždĕ, vĕsledek je tedy stejnĕ. V tabulce 1 je tedy tŕeba vzĭt jako vedoucí poslednĭ řĕdek. Jordanovou eliminaci pak zĭskĕme tabulku T_2 :

-12	-10	0	9	0	0	$-\frac{7}{2}$	0
-2	-3	0	1	1	0	$-\frac{3}{2}$	1
-4	-6	0	2	0	1	$-\frac{3}{2}$	2
1	2	1	-1	0	0	$\frac{1}{2}$	0

Zĭskali jsme řešenĭ, kterĕ je pŕĭpustnĕ. Srovnĕnĭm s pŕedchozĭm PBR zjistĭme, že je stejnĕ, dokonce je i stejnĕ cena. Nejednĕ se vřak o optimum, existujĭ kladnĕ relativnĭ ceny nĕkterĕch nebazickĕch promĕnnĕch. Pravidlo 2 nĕm rovnĕž ukĕže nejednoznaĕnost pŕi vĕbĕru promĕnnĕ, kterou z bĕze vyjmeme, jednĕ se o prvĭ a druhĕ řĕdek. Tuĕnĕ vyznaĕenĕ prvek je vedoucĭm prvkem následnĕ Jordanovy eliminace. Tĭm zĭskĕme tabulku T_3 :

6	17	0	0	-9	0	10	-9
-2	-3	0	1	1	0	$-\frac{3}{2}$	1
0	0	0	0	-2	1	$\frac{3}{2}$	0
-1	-1	1	0	1	0	-1	1

Získali jsme nové PBR, které opět není optimální. Problém je, kterou bazickou proměnnou z báze vyjmout. S tímto jsme se již setkali, a tedy víme, že taková úloha nemá omezenou cenovou funkci, tudíž nemá optimální řešení.

Poznámka. Důležitým poznatkem je skutečnost, že v degenerovaných úlohách není zaručeno, že se v každé iteraci cena řešení sníží, a tedy není zaručena konečnost simplexového algoritmu.

Příklad. V příkladech se nám již stalo, že jsme mohli přidat do báze více než jedinou proměnnou; tuto nejednoznačnost jsme řešili přidáním té proměnné, která má nejvyšší relativní cenu (protože tím cenová funkce klesne nejvýrazněji). Nejednoznačnost v pravidle 2, (tj. kterou proměnnou máme z báze odstranit), je možno obejít dohodou, podle níž vezmeme jako vedoucí řádek následné eliminace ten, který je v matici nejvýše z těch, které připadají v úvahu.

S touto dohodou si ukážeme příklad degenerované úlohy, která se zacykluje (příklad je převzat z knihy [12]).

2	3	-1	-12	0	0	0
-2	-9	1	9	1	0	0
$\frac{1}{3}$	1	$-\frac{1}{3}$	-2	0	1	0

1	0	0	-6	0	-3	0
1	0	-2	-9	1	9	0
$\frac{1}{3}$	1	$-\frac{1}{3}$	-2	0	1	0

0	0	2	3	-1	-12	0
1	0	-2	-9	1	9	0
0	1	$\frac{1}{3}$	1	$-\frac{1}{3}$	-2	0

0	-3	1	0	0	-6	0
1	9	1	0	-2	-9	0
0	1	$\frac{1}{3}$	1	$-\frac{1}{3}$	-2	0

-1	-12	0	0	2	3	0
1	9	1	0	-2	-9	0
$-\frac{1}{3}$	-2	0	1	$\frac{1}{3}$	1	0

0	-6	0	-3	1	0	0
-2	-9	1	9	1	0	0
$-\frac{1}{3}$	-2	0	1	$\frac{1}{3}$	1	0

2	3	-1	-12	0	0	0
-2	-9	1	9	1	0	0
$\frac{1}{3}$	1	$-\frac{1}{3}$	-2	0	1	0

Vidíme, že po šesti iteracích jsme se vrátili k původní tabulce. Pohybujeme se tedy v cyklu délky 6.

Sestrojení podobných příkladů není vůbec snadné. Obecně je například dokázáno, že 6 je nejmenší možná délka cyklu při cyklení simplexového algoritmu.

Pro řešení degenerovaných úloh bylo vyvinuto několik metod, známých jako *anticyklické metody*. Uvedeme si několik příkladů.

1. Nejjednodušší je tzv. *perturbační metoda*. Její myšlenka je velmi jednoduchá: malou změnou parametrů úlohy dosáhneme toho, že pravá strana soustavy přestane být lineární kombinací méně než m sloupců matice \mathbf{A} , tj. úloha přestane být degenerovaná.
2. *Lexikografická metoda* spočívá v tom, že sloupec proměnné, kterou přidáváme do báze, se vybere podle nejvyšší relativní ceny, a v případě nejednoznačnosti volby řádku hledáme nejmenší k , pro které v některých řádcích i, j platí $\frac{a_{ik}}{a_{is}} \neq \frac{a_{jk}}{a_{js}}$, a volíme ten řádek, v němž je tento podíl nejmenší.
3. *Metoda nejmenších indexů* je založena na tom, že sloupce i řádky vybíráme s co nejmenšími indexy, tedy co nejbližše levému hornímu rohu.

Pro všechny tyto metody existují věty, které říkají, že s použitím těchto speciálních strategií výběru vedoucího sloupce a řádku se simplexový algoritmus nezacyklí.

Příklad. Na závěr tohoto odstavce si uvedeme příklad degenerované úlohy, při jejímž řešení problémy nenastanou. Mějme danu úlohu popsanou následující tabulkou T_0 :

-2	-1	-1	-1	-1	0
1	1	3	3	1	4
-1	2	3	0	2	2
1	-1	-1	1	1	0

Výchozí PBR nalezneme např. přímo Jordanovou eliminací, jíž nahradíme 1. fázi dvoufázové metody. Po následném vynulování relativních cen všech bazických proměnných získáme simplexovou tabulku T_1

0	0	3	4	0	6
1	0	1	2	0	2
0	1	2	1	0	2
0	0	0	0	1	0

Máme PBR $\mathbf{x} = [2, 2, 0, 0, 0]^T$. Nejedná se o optimální PBR, vedoucí prvek pro další Jordanovu eliminaci je opět vyznačen tučně. Získáme tak simplexovou tabulku T_2 :

-2	0	1	0	0	2
$\frac{1}{2}$	0	$\frac{1}{2}$	1	0	1
$-\frac{1}{2}$	1	$\frac{3}{2}$	0	0	1
0	0	0	0	1	0

Ani v tomto případě není vypočtené PBR optimální. Opět provedeme Jordanovu eliminaci a získáme tabulku T_3 ve tvaru

$-\frac{5}{3}$	$-\frac{1}{3}$	0	0	0	$\frac{4}{3}$
$\frac{2}{3}$	$-\frac{1}{3}$	0	1	0	$\frac{2}{3}$
$-\frac{1}{3}$	$\frac{1}{3}$	1	0	0	$\frac{2}{3}$
0	0	0	0	1	0

Toto řešení optimální je. Jeho cena je rovna $\frac{4}{3}$, výsledné řešení je ve tvaru $\mathbf{x} = [0, 0, \frac{2}{3}, \frac{2}{3}, 0]^T$.

2.5 Výpočetní složitost simplexového algoritmu a úlohy lineárního programování

Simplexový algoritmus je v praxi velmi často používán a v úlohách „ze života“ dává velmi dobré výsledky. Numerické experimenty ukazují, že na řešení úlohy LP s m omezujícími podmínkami (tj. s m řádky ma-

tice \mathbf{A}) zpravidla v průměru stačí m simplexových iterací. Simplexový algoritmus bývá proto praktickými uživateli považován za velice rychlý. Přesto se ale nedařilo dokázat jeho polynomialitu.

V r. 1972 se překvapivě podařilo ukázat, že pro speciálně zkonstruované „patologické“ úlohy může počet iterací základní verze algoritmu růst exponenciálně s rozměrem úlohy. Simplexový algoritmus se tedy v nejhorším případě chová exponenciálně. Simplexový algoritmus má však mnoho verzí, daných různými strategiemi výběru vedoucího sloupce a řádku v případech, kdy tato volba není jednoznačná. Postupně se podařilo sestavit podobné příklady pro další známé verze simplexového algoritmu. V současné době jsou pro všechny známé verze simplexového algoritmu sestaveny příklady, ukazující, že v nejhorším případě se chovají exponenciálně, ale není známo, zda pro simplexový algoritmus existuje jeho polynomiální verze. Z tohoto hlediska je simplexový algoritmus výjimkou mezi nepolynomiálními algoritmy – ačkoliv jeho chování je v nejhorším případě exponenciální, „v průměru“ je algoritmus velice rychlý.

Otázku polynomiality úlohy LP zodpověděl kladně v r. 1979 L.G. Chačijan, který publikoval tzv. *elipsoidový algoritmus*. Tento algoritmus je sice v průměru podstatně pomalejší než simplexový, ale podstatné je, že i v nejhorším případě je počet iterací polynomiální funkcí rozměru úlohy ¹. Elipsoidový algoritmus tedy teoreticky dokazuje polynomialitu úlohy LP. Pro praktické použití však zůstává simplexový algoritmus nadále výhodnější.

3 Dualita úloh lineárního programování

3.1 Primární a duální úloha lineárního programování

Uvažujme úlohu LP v kanonickém tvaru, tedy úlohu

$$\begin{aligned} \min \mathbf{c}^T \mathbf{x} \\ \mathbf{Ax} \geq \mathbf{b} \\ x_i \geq 0 \end{aligned} \tag{1}$$

Tuto (výchozí) úlohu budeme nazývat *primární úlohou* lineárního programování (též se někdy říká *přímá úloha*).

Zkusme nyní upravovat podmínku $\mathbf{Ax} \geq \mathbf{b}$. O matici \mathbf{A} víme, že je typu m/n , vektor \mathbf{b} má m složek. Vezmeme-li nyní nějaký vektor $\mathbf{y} \in \mathbf{R}^m$ s nezápornými složkami a vynásobme jím zleva uvedenou nerovnost, bude platit

$$\mathbf{y}^T \mathbf{b} \leq \mathbf{y}^T (\mathbf{Ax}).$$

O operaci násobení matic víme, že je asociativní, platí tedy dále

$$\mathbf{y}^T \mathbf{b} \leq \mathbf{y}^T (\mathbf{Ax}) = \mathbf{y}^T \mathbf{Ax} = (\mathbf{A}^T \mathbf{y})^T \mathbf{x}.$$

Tato nerovnost musí platit pro libovolný vektor $\mathbf{y} \in \mathbf{R}^m$ s nezápornými složkami. Kdybychom našli \mathbf{y} takový, pro který by byla splněna podmínka

$$\mathbf{A}^T \mathbf{y} \leq \mathbf{c},$$

měli bychom celkem

$$\mathbf{b}^T \mathbf{y} = \mathbf{y}^T \mathbf{b} \leq \mathbf{y}^T (\mathbf{Ax}) = \mathbf{y}^T \mathbf{Ax} = (\mathbf{A}^T \mathbf{y})^T \mathbf{x} \leq \mathbf{c}^T \mathbf{x}.$$

Speciálně tedy platí

$$\mathbf{b}^T \mathbf{y} = \mathbf{y}^T \mathbf{b} \leq \mathbf{c}^T \mathbf{x}.$$

To znamená, že výraz na levé straně je dolním odhadem cenové funkce $\mathbf{c}^T \mathbf{x}$ (kterou minimalizujeme). Všimněme si dále toho, že nerovnost je splněna pro libovolnou levou stranu. Nás pochopitelně zajímá

¹Přesněji: je polynomiální funkcí rozměru vstupních dat při jejich binárním kódování. Otázka existence polynomiálního algoritmu při aritmetickém kódování vstupních dat zůstává otevřená.

nejlepší dolní odhad cenové funkce, a ten je roven maximální hodnotě výrazu $\mathbf{b}^T \mathbf{y}$ na levé straně. Nejlepší dolní odhad tedy dostaneme nalezením následujícího maxima:

$$\begin{aligned} \max \mathbf{b}^T \mathbf{y} \\ \mathbf{A}^T \mathbf{y} \leq \mathbf{c} \\ y_i \geq 0 \end{aligned} \tag{2}$$

To je opět úloha lineárního programování; nazývá se *duální úloha* k původní (primární) úloze (1).

Lze snadno ukázat, že duální úloha k duální úloze (2) je původní úloha (1). Postupovali bychom stejným postupem, pouze by se obrátily nerovnosti a místo nejvyššího dolního odhadu bychom hledali nejnižší horní odhad. Proto říkáme, že úlohy (1) a (2) tvoří *dvojíci vzájemně duálních úloh*.

<p><i>Primární úloha (1)</i></p> $\begin{aligned} \min \mathbf{c}^T \mathbf{x} \\ \mathbf{A} \mathbf{x} \geq \mathbf{b} \\ x_i \geq 0 \end{aligned}$	<p><i>Duální úloha (2)</i></p> $\begin{aligned} \max \mathbf{b}^T \mathbf{y} \\ \mathbf{A}^T \mathbf{y} \leq \mathbf{c} \\ y_i \geq 0 \end{aligned}$
--	--

3.2 Věty o dualitě

Uvedené úvahy shrneme do následující věty, nazývané *slabá věta o dualitě*. Její důkaz byl vlastně již ukázán postupem, jímž jsme z primární úlohy odvodili úlohu duální.

Věta 3.1. (Slabá věta o dualitě) *Pro každé přípustné řešení \mathbf{x} primární úlohy (1) a pro každé řešení \mathbf{y} duální úlohy (2) platí $\mathbf{b}^T \mathbf{y} \leq \mathbf{c}^T \mathbf{x}$.*

Označme \mathbf{X} (resp. \mathbf{Y}) množinu všech přípustných řešení primární (resp. duální) úlohy, tj.

$$\begin{aligned} \mathbf{X} &= \{ \mathbf{x} \mid \mathbf{A} \mathbf{x} \geq \mathbf{b}, x_i \geq 0 \}, \\ \mathbf{Y} &= \{ \mathbf{y} \mid \mathbf{A}^T \mathbf{y} \leq \mathbf{c}, y_i \geq 0 \}. \end{aligned}$$

Ze slabé věty o dualitě nyní dostaneme následující důsledky.

Důsledek 3.1. *Platí-li pro $\mathbf{x}_0 \in \mathbf{X}$ a $\mathbf{y}_0 \in \mathbf{Y}$ rovnost*

$$\mathbf{b}^T \mathbf{y}_0 = \mathbf{c}^T \mathbf{x}_0,$$

pak jsou vektory \mathbf{x}_0 a \mathbf{y}_0 optimálními řešeními úloh (1) a (2).

Důsledek 3.2. *Je-li $\mathbf{X} \neq \emptyset$ a je-li cenová funkce $\mathbf{c}^T \mathbf{x}$ zdola neomezená na \mathbf{X} , pak je $\mathbf{Y} = \emptyset$.*

Důsledek 3.3. *Je-li množina $\mathbf{Y} \neq \emptyset$ a je-li funkce $\mathbf{b}^T \mathbf{y}$ shora neomezená na \mathbf{Y} , pak je $\mathbf{X} = \emptyset$.*

Z prvního důsledku plyne, že rovnost $\mathbf{b}^T \mathbf{y} = \mathbf{c}^T \mathbf{x}$ je postačující podmínkou pro optimalitu. Otázkou však zůstává, zda-li je tato podmínka nutná. Dá se dokázat, že ano.

Věta 3.2. (Silná věta o dualitě) *Vektory $\mathbf{x} \in \mathbf{X}$ a $\mathbf{y} \in \mathbf{Y}$ jsou optimální, právě když platí rovnost $\mathbf{b}^T \mathbf{y} = \mathbf{c}^T \mathbf{x}$.*

Důkaz. Je poměrně náročný, proto jej nebudeme uvádět. □

Důsledek 3.4. *Mají-li obě úlohy alespoň jedno přípustné řešení, pak obě úlohy mají optimální řešení \mathbf{x}_0 , resp. \mathbf{y}_0 a platí pro ně rovnost $\mathbf{b}^T \mathbf{y}_0 = \mathbf{c}^T \mathbf{x}_0$.*

Nyní si ukážeme dvojice duálních úloh v některých jiných případech.

1. Duální úloha k primární úloze ve standardním tvaru. Vyjdeme z primární úlohy

$$\begin{aligned} \min \mathbf{c}^T \mathbf{x} \\ \mathbf{Ax} = \mathbf{b} \\ x_i \geq 0 \end{aligned}$$

Postupujeme obdobně jako v případě úlohy v kanonickém tvaru, tj. vztah $\mathbf{Ax} = \mathbf{b}$ násobíme vektorem \mathbf{y} a upravujeme. Platí

$$\mathbf{Ax} = \mathbf{b} \Rightarrow \mathbf{y}^T \mathbf{Ax} = \mathbf{y}^T \mathbf{b} = \mathbf{b}^T \mathbf{y}.$$

Zde je ale významný rozdíl oproti úloze v kanonickém tvaru. Tam jsme museli předpokládat, že \mathbf{y} je (libovolný) vektor s nezápornými složkami (jinak by se nám mohlo změnit znaménko nerovnosti v opačné). Zde ale máme rovnost, a tedy \mathbf{y} může být libovolný vektor z \mathbf{R}^m .

Máme tedy vztah

$$\mathbf{b}^T \mathbf{y} = \mathbf{y}^T \mathbf{Ax} = (\mathbf{A}^T \mathbf{y})^T \mathbf{x}.$$

Pokud bude platit $\mathbf{A}^T \mathbf{y} \leq \mathbf{c}$, bude dále (s využitím nezápornosti všech složek vektoru \mathbf{x})

$$\mathbf{b}^T \mathbf{y} = \mathbf{y}^T \mathbf{Ax} = (\mathbf{A}^T \mathbf{y})^T \mathbf{x} \leq \mathbf{c}^T \mathbf{x},$$

čímž jsme opět získali vztah $\mathbf{b}^T \mathbf{y} \leq \mathbf{c}^T \mathbf{x}$. Zapišme tedy obě úlohy - primární i duální, kde primární úlohou je úloha ve standardním tvaru.

Primární úloha

$$\begin{aligned} \min \mathbf{c}^T \mathbf{x} \\ \mathbf{Ax} = \mathbf{b} \\ x_i \geq 0 \end{aligned}$$

Duální úloha

$$\begin{aligned} \max \mathbf{b}^T \mathbf{y} \\ \mathbf{A}^T \mathbf{y} \leq \mathbf{c} \\ \mathbf{y} \in \mathbf{R}^m \end{aligned}$$

2. Podobně bychom našli i duální úlohu k primární úloze ve standardním tvaru, ale s maximalizací ceny:

$$\begin{aligned} \max \mathbf{c}^T \mathbf{x} \\ \mathbf{Ax} = \mathbf{b} \\ x_i \geq 0 \end{aligned}$$

Analogicky jako v minulém případě se odvodí, že duální úloha má v tomto případě tvar

$$\begin{aligned} \min \mathbf{b}^T \mathbf{y} \\ \mathbf{A}^T \mathbf{y} \geq \mathbf{c} \\ \mathbf{y} \in \mathbf{R}^m \end{aligned}$$

Příklad. Ukažme si na příkladu úlohy o dietě tvar a praktickou interpretaci duální úlohy.

Připomeňme si nejprve úlohu o dietě. Řešili jsme úlohu

$$\begin{aligned} \min \mathbf{c}^T \mathbf{x} \\ \mathbf{Ax} \geq \mathbf{b} \\ x_i \geq 0, \end{aligned}$$

kde c_i byly jednotkové ceny jednotlivých potravin, a_{ij} obsah i -tého vitamínu v jednotce j -té potraviny, b_j minimální požadované množství j -tého vitamínu ve směsi. Jak bylo uvedeno, duální úloha k této úloze má tvar

$$\begin{aligned} \max \mathbf{b}^T \mathbf{y} \\ \mathbf{A}^T \mathbf{y} \leq \mathbf{c} \\ y_i \geq 0. \end{aligned}$$

Tato duální úloha má velmi přímou praktickou interpretaci. Výrobce chce dát na trh tablety se syntetickými vitamíny a nahradit jimi reálnou stravu, přičemž hledá maximální ceny, při nichž je schopen konkurovat reálné stravě. Zde y_i je hledaná cena jednotky i -tého syntetického vitamínu, výraz $\mathbf{b}^T \mathbf{y}$ je celková cena, kterou výrobce obrzí, když pokryje poptávku, a nerovnost $\mathbf{A}^T \mathbf{y} \leq \mathbf{c}$ garantuje konkurenceschopnost, protože říká, že dodávané vitamíny nesmí být dražší než vitamíny v reálné stravě. Podle této interpretace se duálním proměnným v ekonomických aplikacích říká *stínové ceny*.

3.3 Dobré charakteristiky

V tomto odstavci si v poněkud jiném světle ukážeme význam duality úloh LP. Nejprve ale začneme prohloubením poznatků z posledních dvou kapitol předmětu TGD1.

Známe pojem *jazyka třídy NP*. Jazyk J patří do třídy NP, jestliže existuje nedeterministický přijímací počítač, který pracuje v polynomiálně omezeném čase a přijímá jazyk J . Pojem abstraktního počítače je zde modelem algoritmu pro řešení konkrétního problému – na poněkud intuitivnější úrovni lze říci, že rozhodovací problém náleží do třídy NP, jestliže pro něj existuje algoritmus, který umí „uhodnout“ (nedeterminističnost) řešení, přičemž:

- možnost volby „uhodnutých“ (nedeterministicky zvolených) kroků *existuje*, právě když odpověď na otázku je kladná,
- každý konkrétní výpočet proběhne v polynomiálně omezeném čase.

Definice 3.1. Třída co-NP je třída všech jazyků, jejichž doplněk náleží do třídy NP.

Zhruba řečeno: problém je třídy co-NP, právě když odpověď na opačnou otázku náleží do třídy NP. Tedy zcela samozřejmě do třídy co-NP například náleží problém:

HAM

Vstup: neorientovaný graf G na n uzlech

Úkol: zjistit, zda G je nehamiltonovský.

Analogicky lze ovšem naformulovat negace všech známých úloh z třídy NP, a tyto (někdy jazykově dost kostrbaté) problémy budou ze třídy co-NP. Existují ale i problémy, jejichž příslušnost do třídy co-NP je přirozená. Uvedeme dva příklady.

1. *Prvočíselnost*. Otázka, zda dané číslo je prvočíslem, náleží do co-NP, protože k *negativní* odpovědi postačí napsat naše číslo jako součin dvou čísel.²
2. *Tuhost grafu*. Řekneme, že neorientovaný graf G je *tuhý*, jestliže pro každou množinu uzlů $S \subset U(G)$ je počet komponent grafu $G - S$ nejvýše roven počtu prvků množiny S . Pojem tuhosti grafu je důležitý v teorii hamiltonovských grafů, protože tuhost grafu je nutnou podmínkou jeho hamiltonovskosti (každý hamiltonovský graf je tuhý).

Problém rozhodnout, zda daný graf G je tuhý, náleží do co-NP, protože k rozhodnutí, že G *není* tuhý stačí ukázat množinu $S \subset U(G)$, pro niž má graf $G - S$ více než $|S|$ komponent.

Definice 3.2. Řekneme, že jazyk J je dobře charakterizovaný, jestliže $J \in NP \cap co-NP$. Věta, která pro daný jazyk J dokazuje jeho příslušnost do třídy $NP \cap co-NP$, se nazývá dobrá charakteristika.

V této podobě definice vypadá poněkud odtažitě. Pojem dobré charakteristiky zavedl J. Edmonds jako popis toho, co intuitivně chápeme jako „dobré pochopení“ či „dobré porozumění“ daného problému. Ukažme si význam a smysl dobrých charakteristik na dvou příkladech.

²Problém prvočíselnosti náleží i do NP, ale důkaz tohoto faktu je značně netriviální. Poznamenejme ještě, že v roce 2002 byl pro testování prvočíselnosti nalezen polynomiální algoritmus.

Příklad. 1. Mengerova věta.

- Jestliže graf G je k -souvislý mezi uzly u a v , můžeme toto (při troše štěstí) dokázat *pozitivním argumentem* – nalezneme k uzlově disjunktních cest, vedoucích z u do v . (Tedy otázka „Je G k -souvislý?“ je v NP).
- Jestliže graf G *není* k -souvislý mezi u a v , můžeme toto dokázat také *pozitivním argumentem*: (při troše štěstí) nalezneme uzlový řez, oddělující u a v , mající méně než k prvků. (Tedy otázka „Je G k -souvislý?“ je v co-NP).

Mengerova věta tedy je dobrá charakteristika.

2. Hamiltonovskost grafu.

- Jestliže G je hamiltonovský, můžeme to (při troše štěstí) dokázat tak, že v G najdeme hamiltonovskou kružnici. Tedy otázka „Je G hamiltonovský?“ je v NP.
- Jestliže ale G hamiltonovský *není*, jsme v koncích, protože nevíme, co hledat, abychom dokázali *neexistenci* hamiltonovské kružnice. Schází nám věta, dokazující příslušnost k co-NP.

Dobrá charakteristika není známa, jazyk **HAM** tedy není (dosud) dobře charakterizovaný.

Téměř samozřejmé je následující tvrzení.

Věta 3.3.

$$P \subset (NP \cap co-NP)$$

Historie ukazuje, že velmi často je nalezení dobré charakteristiky pro daný problém zároveň zásadním krokem k nalezení polynomiálního algoritmu. Zde je i hlavní význam *věty o dualitě lineárního programování*, protože dává dobrou charakteristiku úlohy LP. Uvažujme následující rozhodovací problém:

LP

Vstup: úloha LP ve standardním tvaru a její PBR x_0 .

Úkol: zjistit, zda x_0 je optimální.

Jestliže dané PBR není optimální, pak se o tom „snadno“ (ve smyslu NP) přesvědčíme nalezením lepšího řešení; dokázat optimalitu daného řešení přímo by ale znamenalo ukázat, že všechna ostatní přípustná řešení jsou horší.

S využitím duality může rozhodování vypadat takto.

- Jestliže x_0 není optimální, dokážeme to (pozitivním argumentem) nalezením jiného PBR s menší cenou.
- Jestliže x_0 je optimální, dokážeme to též pozitivním argumentem, a sice nalezením přípustného řešení y_0 duální úlohy takového, že $\mathbf{c}^T \mathbf{x}_0 = \mathbf{b}^T \mathbf{y}_0$.

Úloha LP, zformulovaná jako rozhodovací úloha, je tedy v $NP \cap co-NP$.

Pro *úlohu optimálního toku* je dobrou charakteristikou věta o potenciálech, tj. Věta 1.3. Abychom se o tom přesvědčili, uvažujme takto.

OPT-FLOW

Vstup: síť \vec{G} a v ní tok x_{ij} .

Úkol: zjistit, zda x_{ij} je optimální.

S využitím Věty 1.3 vypadá rozhodování takto.

- Jestliže x_{ij} není optimální, najdeme jiný tok s menší cenou.
- Jestliže x_{ij} je optimální, dokážeme to nalezením potenciálů.

Věta 1.3 tedy dokazuje příslušnost úlohy do $NP \cap co-NP$.

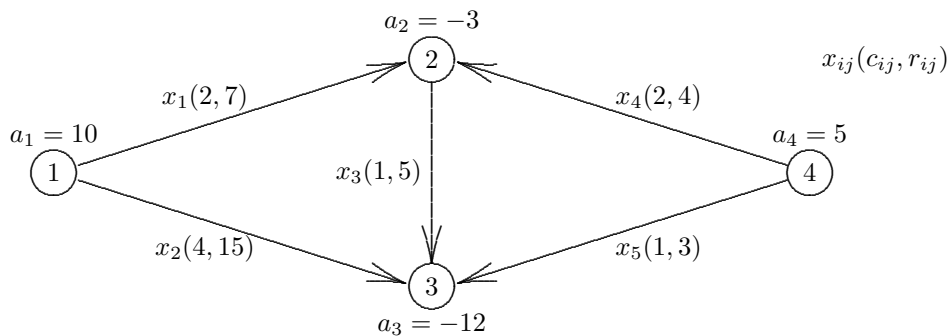
Poznamenejme ještě, že Věta 1.1 (o ZRP) sama o sobě dobrou charakteristikou není, protože v případě optimality toku svazuje *neexistenci* lepšího toku s *neexistencí* ZRP. Její praktickou použitelnost zajišťuje až Floydův algoritmus, díky němuž umíme v polynomiálním čase dokázat i neexistenci ZRP.

Podle Věty 3.3 víme, že $P \subset (NP \cap \text{co-NP})$. Otázka, zda zde platí rovnost, je dalším velkým otevřeným problémem v teorii výpočetní složitosti. Před nalezením Chačijanova algoritmu byla úloha LP jedním z možných „kandidátů na protipříklad“, protože byla dobře charakterizovaná, ale nebyl pro ni znám polynomiální algoritmus. Dalším podobným kandidátem byl problém prvočíselnosti, ale jak už jsme uvedli, v roce 2002 byl pro tento problém nalezen polynomiální algoritmus.

4 Celočíselné lineární programování

4.1 Formulace úlohy

Příklad. Uvažujme úlohu nalezení optimálního toku v síti na následujícím obrázku. Uzlové ohodnocení odpovídá intenzitě uzlů, hranové ohodnocení ve tvaru $x_i(c_i, r_i)$ znamená tok hranou, cenu a propustnost. Čísla v kroužcích jsou pořadová čísla uzlů.



Úlohu lze zřejmě převést na následující úlohu LP:

$$\min 2x_1 + 4x_2 + x_3 + 2x_4 + x_5$$

za podmínek

$$\begin{array}{rcccccl} x_1 & +x_2 & & & & = & 10 \\ -x_1 & & +x_3 & -x_4 & & = & -3 \\ & -x_2 & -x_3 & & -x_5 & = & -12 \\ & & & x_4 & +x_5 & = & 5 \\ x_1 & & & & & \leq & 7 \\ & x_2 & & & & \leq & 15 \\ & & x_3 & & & \leq & 5 \\ & & & x_4 & & \leq & 4 \\ & & & & x_5 & \leq & 3 \end{array}$$

$$x_i \geq 0, \quad i = 1 \dots 5$$

Jsou-li hodnoty r_{ij} a a_i celočíselné (např. kusy výrobků), pak by měl být celočíselný i optimální tok.

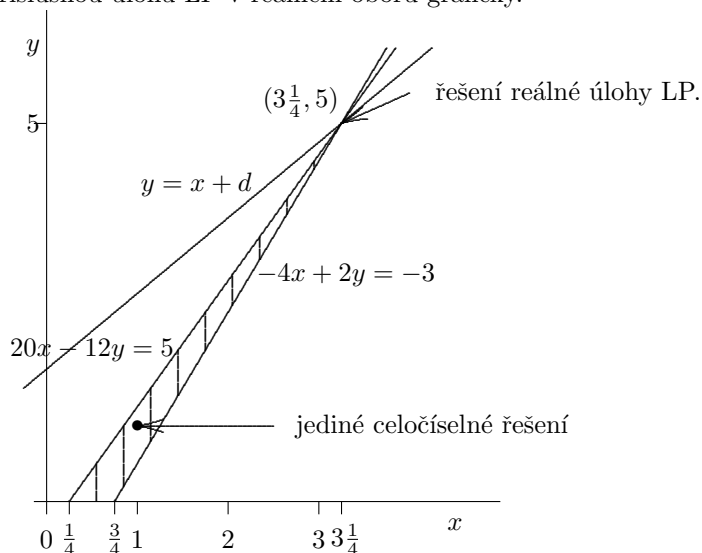
Příklad. Mějme dán ohodnocený graf (libovolný) s alespoň dvěma uzly a mějme úkol nalézt v něm minimální cestu z uzlu u do uzlu v ($u, v \in U(\vec{G})$). Definujme intenzity $a_u = 1$, $a_v = -1$ a $a_x = 0$ pro všechny ostatní uzly x , ceny hran budou rovny zadanému hranovému ohodnocení a propustnosti hran budou rovny jedné. Najdeme-li v této síti celočíselný optimální tok, pak hrany s tokem 1 určují minimální cestu z u do v . Poznamenejme, že v předchozím příkladu byla celočíselnost hledaného řešení vhodná, zde je však nezbytná.

Na první pohled by se mohlo zdát, že podmínka celočíselnosti hledaného řešení neznamená žádné zásadní omezení – vyřešíme úlohu v reálném oboru, a nalezené řešení vhodně zaokrouhlíme. Následující příklad ukazuje, že podobné postupy nemají naději na úspěch.

Příklad. Mějme dānu následující úlohu LP v kanonickém tvaru.

$$\begin{aligned} \min \quad & x - y \\ 20x - 12y & \geq 5 \\ -4x + 2y & \geq -3 \\ x \geq 0; y & \geq 0; \\ x, y & \text{ celočíselné} \end{aligned}$$

Řešme příslušnou úlohu LP v reálném oboru graficky.



Reálná úloha LP má řešení $x = 3,25$, $y = 5$, ale jediné celočíselné řešení dané úlohy je $x = y = 1$, což zřejmě nezískáme zaokrouhlením optima reálné úlohy LP.

Je zřejmé, že podobně by šlo sestavit příklad úlohy CLP, jejíž optimum je libovolně daleko od optima příslušné reálné úlohy. Z tohoto příkladu vidíme, že hledání celočíselného řešení úlohy LP je samostatný problém.

Definice 4.1. Úlohou celočíselného lineárního programování (CLP) rozumíme úlohu

$$\begin{aligned} \min \quad & \mathbf{c}^T \mathbf{x} \\ \mathbf{A} \mathbf{x} & = \mathbf{b} \\ x_i & \geq 0 \\ x_i & \text{ celočíselné,} \end{aligned}$$

kde $\mathbf{c} \in \mathbf{R}^n$, \mathbf{A} je celočíselná matice typu m/n a $\mathbf{b} \in \mathbf{Z}^m$ je celočíselný vektor.

Poznámka. 1. Je zřejmé, že o prvcích matice \mathbf{A} a vektoru \mathbf{b} stačí předpokládat, že jsou racionální, a vzniklé úlohy jsou ekvivalentní (vynásobíme rovnice společným jmenovatelem). Naproti tomu ceny předpokládáme reálné. Předpoklady celočíselnosti prvků matice \mathbf{A} a vektoru \mathbf{b} budeme požadovat vždy po zbytek této kapitoly.

2. Předchozí definice dává úlohu CLP ve standardním tvaru. Opět platí, že úlohu lze formulovat obecně i kanonicky a formulace jsou ekvivalentní.

4.2 Výpočetní složitost úlohy celočíselného lineárního programování

Jak již víme z kapitoly 2.5, úloha LP v reálném oboru je řešitelná v polynomiálním čase (Chačijanovým algoritmem). Uvidíme, že požadavkem celočíselnosti se situace zásadně mění.

Především, ptáme-li se, zda CLP patří do třídy NP, je třeba formulovat úlohu CLP jako rozhodovací problém. Převodeme tedy úlohu CLP na tvar CLP':

CLP'

Vstup: matice \mathbf{A} , vektory \mathbf{b} , \mathbf{c} a číslo L .

Úkol: zjistit, zda existuje vektor \mathbf{x} takový, že

$$\mathbf{c}^T \mathbf{x} \leq L$$

$$\mathbf{A} \mathbf{x} = \mathbf{b}$$

$$x_i \geq 0; x_i \text{ celočíselné.}$$

CLP' je již rozhodovací problém a otázka, zda patří do třídy NP, má smysl. Problémem může být to, že složky vektoru \mathbf{x} by teoreticky mohly být libovolně velké. Proto úlohu ještě jednou přeformulujeme – varianta CLP'' je obdobná jako úloha CLP', ale s omezenými proměnnými.

CLP''

Vstup: matice \mathbf{A} , vektory \mathbf{b} , \mathbf{c} , čísla L a K .

Úkol: zjistit, zda existuje vektor \mathbf{x} takový, že

$$\mathbf{c}^T \mathbf{x} \leq L$$

$$\mathbf{A} \mathbf{x} = \mathbf{b}$$

$$x_i \geq 0; x_i \leq K; x_i \text{ celočíselné.}$$

Platí následující věta.

Věta 4.1. *Úloha CLP'' je NP-úplná.*

Důkaz. 1. Ověříme, že $\text{CLP}'' \in \text{NP}$:

a) nedeterministicky generujeme celočíselný vektor \mathbf{x} ,

b) ověříme, zda \mathbf{x} je přípustné řešení,

c) pokud ano, ověříme, zda $\mathbf{c}^T \mathbf{x} \leq L$.

(Poznámka: bez omezení $x_i \leq K$ by v b) a c) byly problémy s polynomiálností).

2. Úplnost CLP'' dokážeme převodem $\text{SAT} \leq \text{CLP}''$. Myšlenku tohoto převodu si ukážeme na příkladu. Mějme dānu následující logickou formuli:

$$f(x_1, x_2, x_3) = (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \bar{x}_2) \wedge (\bar{x}_1 \vee x_3) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3).$$

Je zřejmé, že formule f je splnitelnā, právě když existují hodnoty proměnných x_1, x_2, x_3 , vyhovující nerovnostem:

$$x_1 + x_2 + x_3 \geq 1$$

$$x_1 + (1 - x_2) \geq 1$$

$$(1 - x_1) + x_3 \geq 1$$

$$(1 - x_1) + (1 - x_2) + (1 - x_3) \geq 1$$

$$x_1, x_2, x_3 \in \{0, 1\}$$

To však ještě není úloha CLP - nemāme žádnou cenovou funkci ani optimalizaci. To se však snadno dá obejít následujícím trikem. Zavedeme novou proměnnou y , např. první nerovnost nahradíme nerovností

$$x_1 + x_2 + x_3 \geq y$$

a maximalizujeme hodnotu proměnné y . Pokud je $y_{max} \geq 1$, jsme hotovi, formule je splnitelná. Hledanou úlohu CLP'', která vznikla z naší úlohy SAT, lze zapsat takto:

$$\begin{aligned} & \max y \\ & x_1 + x_2 + x_3 - y \geq 0 \\ & x_1 + (1 - x_2) \geq 1 \\ & (1 - x_1) + x_3 \geq 1 \\ & (1 - x_1) + (1 - x_2) + (1 - x_3) \geq 1 \\ & x_1, x_2, x_3 \in \{0, 1\} \end{aligned}$$

Zřejmě platí, že f je splnitelná, právě když úloha CLP'' má optimální řešení splňující podmínku $y_{max} \geq 1$. \square

Poznámka. Úloha CLP'' je zřejmě speciální případ úlohy CLP. Úloha CLP je tedy alespoň tak těžká, jako NP-úplná úloha CLP'', ale nevíme, zda je \in NP. Taková úloha se nazývá **NP-těžká**.

Důsledek 4.1. Úloha CLP je NP-těžká.

Lze dokázat i platnost následující věty, důkaz ovšem není lehký, lze jej nalézt např. v [10].

Věta 4.2. Úloha CLP' je NP-úplná.

Skutečnost, že úloha CLP' je NP-těžká, znamená, že na ni lze převést všechny ostatní NP-úplné úlohy. Jeden takový převod si ukážeme v následujícím příkladu.

Příklad. Formulace úlohy obchodního cestujícího jako úlohy CLP.

Mějme ohodnocený orientovaný graf \vec{G} . Uzly grafu \vec{G} označíme čísly $0 \dots n$; ohodnocení hrany (i, j) označíme c_{ij} . Pokud $(i, j) \notin H(\vec{G})$, položíme hodnotu c_{ij} rovnou nějakému velkému číslu, a dále položíme $c_{ii} = 0$, $i = 1, \dots, n$. Matice $[c_{ij}]$ obecně nemusí být symetrická. Hledáme cyklus \vec{C} , který prochází všemi uzly a minimalizuje výraz $\sum c_{ij}$.

$$\text{Položíme } x_{ij} = \begin{cases} 1, & \text{jestliže } (i, j) \in H(\vec{C}), \\ 0, & \text{jestliže } (i, j) \notin H(\vec{C}). \end{cases}$$

Pro naši úlohu obchodního cestujícího můžeme nyní zformulovat následující úlohu CLP.

$$\min \sum_{i,j=0}^n c_{ij} x_{ij}$$

za podmínek

$$\sum_{i=0}^n x_{ij} = 1 \text{ pro } j = 0 \dots n, \tag{a}$$

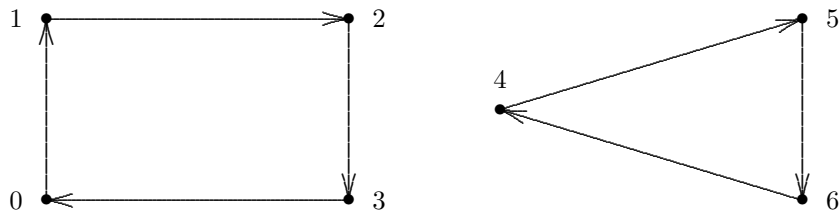
$$\sum_{j=0}^n x_{ij} = 1 \text{ pro } i = 1 \dots n, \tag{b}$$

$$0 \leq x_{ij} \leq 1 \forall i, j,$$

$$x_{ij} \text{ celočíselné.}$$

Podmínka (a) říká, že do každého z uzlů $1, \dots, n$ vchází právě jedna hrana, a podobně podmínka (b) říká, že z každého z uzlů $0, \dots, n$ vychází právě jedna hrana. Z obou podmínek plyne, že i do uzlu 0 vchází právě jedna hrana.

Zdálo by se tedy, že podmínky (a) a (b) vyjadřují požadavek, aby hledaný podgraf byl cyklem. Problém je ale v tom, že přípustné řešení nemusí odpovídat souvislému grafu (a tedy může vyjadřovat disjunktí sjednocení více cyklů). Jednu z možných situací ukazuje následující obrázek.



Pokusíme se o nápravu – je třeba přidat další podmínku, která by z množiny přípustných řešení vyloučila ta, jež odpovídají nesouvislým grafům. Jedna možná myšlenka spočívá v rozkladu množiny uzlů na dvě neprázdné disjunktí množiny S, \bar{S} (tj. $S \cup \bar{S} = U(\vec{G})$). Souvislost grafu pak znamená, že pro každý takovýto rozklad musí platit, že z S vede do \bar{S} alespoň jedna hrana. Tím dostaneme tuto podmínku:

$$\sum_{i \in S, j \in \bar{S}} x_{ij} \geq 1 \quad \forall S \subset \{0, 1, \dots, n\}, S \neq \emptyset, S \neq U(\vec{G}).$$

Takto získaná úloha CLP již je ekvivalentní s původní úlohou obchodního cestujícího. Potíž je ale jinde – možných rozkladů (S, \bar{S}) množiny $U(\vec{G})$ je $2^{n+1} - 2$. To znamená, že rozměr získané úlohy CLP roste exponenciálně s rozměrem původní úlohy obchodního cestujícího, a tedy náš převod není polynomiální. Musíme tedy hledat lepší podmínku.

Zkusíme následující podmínku.

$$u_i - u_j + nx_{ij} \leq n - 1 \quad \text{pro } 1 \leq i \neq j \leq n, u_i \in \mathbf{Z}, i = 1 \dots n. \quad (c)$$

Zde $u_i, i = 1, \dots, n$, jsou další celočíselné proměnné (ale ne nutně nezáporné – přidáním podmínky (c) dostáváme úlohu CLP v obecném tvaru).

Ukážeme, že podmínka (c) vylučuje kratší cykly.

- a) Existují-li čísla u_i , splňující podmínku (c), pak každý cyklus prochází uzlem 0 – kdybychom měli např. případ znázorněný na předchozím obrázku, pak by na cyklu 4, 5, 6 muselo platit

$$\left. \begin{array}{l} u_4 - u_5 + 6 \leq 5 \\ u_5 - u_6 + 6 \leq 5 \\ u_6 - u_4 + 6 \leq 5 \end{array} \right\} \Rightarrow \left. \begin{array}{l} u_4 - u_6 \leq -2 \\ u_4 - u_6 \geq 1 \end{array} \right\} \text{SPOR.}$$

- b) Pro každý cyklus \vec{C} , který prochází všemi uzly, čísla existují. Položme $u_0 = 0$ a $u_i = t$, jestliže uzel i je t -tý v pořadí na cyklu \vec{C} . Ukážeme, že podmínka (c) je splněna.

- Je-li $x_{ij} = 0$ (tj. hrana (i, j) není hranou cyklu \vec{C}), pak musí platit $u_i - u_j \leq n - 1$. To by mohlo neplatit jen pro $u_i = n, u_j = 0$, ale to by znamenalo, že $j = 0$ a hrana $(i, 0)$ je v \vec{C} , což je v předpokladu vyloučeno.
- Je-li $x_{ij} = 1$, tj. hrana (i, j) je hranou cyklu \vec{C} , pak musí platit

$$u_i - u_j + n \leq n - 1 \Leftrightarrow u_i - u_j \leq -1,$$

ale z konstrukce čísel u_i plyne, že $u_i - u_j = -1$ s výjimkou poslední hrany cyklu, tj. hrany $(n, 0)$. Tento případ se však v podmínce (c) neuvažuje; podmínka (c) je tedy splněna.

Poznámka. Všimněme si, že podmínek (c) je celkem n^2 . Rozměr získané úlohy CLP je tedy polynomiální funkcí rozměru grafu \vec{G} .

Poznámka. Považujeme-li NP-úplnost úlohy obchodního cestujícího za dokázanou již dříve, pak převod z předchozího příkladu dává alternativní důkaz Důsledku 4.1.

4.3 Celočíselné lineární programování a totálně unimodulární matice

Definice 4.2. Řekneme, že matice $\mathbf{A} = [a_{ij}]$ typu m/n je totálně unimodulární, jestliže

- a) $a_{ij} \in \{0, 1, -1\}$, $i = 1, \dots, n$, $j = 1, \dots, m$,
- b) determinant každé čtvercové podmatice matice \mathbf{A} je roven 0, 1 nebo -1 .

Věta 4.3. Je-li \mathbf{A} totálně unimodulární, pak pro každý celočíselný vektor \mathbf{b} je každé přípustné bazické řešení celočíselné.

Důkaz. Necht $B = \{A_{j1}, \dots, A_{jm}\}$ je m -tice lineárně nezávislých sloupců matice \mathbf{A} a \mathbf{B} je příslušná regulární podmatice matice \mathbf{A} . Bazické řešení \mathbf{x} (po vynechání nulových složek) splňuje

$$\mathbf{x} = \mathbf{B}^{-1}\mathbf{b} = \frac{1}{\det \mathbf{B}}\mathbf{B}^A\mathbf{b},$$

ale \mathbf{B}^A je celočíselná, a je-li \mathbf{b} celočíselný a $\det \mathbf{B} = \pm 1$, pak i \mathbf{x} je celočíselný. □

Poznámka. Platí i obrácené tvrzení.

Důsledek 4.2. Je-li \mathbf{A} totálně unimodulární, pak se optimální řešení úlohy CLP shoduje s řešením příslušné reálné úlohy LP.

Důsledek 4.3. Úlohu CLP s totálně unimodulární maticí lze řešit simplexovým algoritmem a výsledné řešení je celočíselné.

Důsledek 4.4. Úloha CLP s totálně unimodulární maticí je řešitelná v polynomiálním čase.

Věta 4.4. Necht \mathbf{A} je matice typu m/n taková, že

- a) $a_{ij} \in \{-1, 0, 1\}$, $i = 1, \dots, m$, $j = 1, \dots, n$,
- b) každý sloupec matice \mathbf{A} obsahuje
 - buďto nejvýše jeden nenulový prvek,
 - nebo právě dva nenulové prvky, a to $+1$ a -1 .

Pak je matice \mathbf{A} totálně unimodulární.

Důkaz. Necht \mathbf{B} je čtvercová podmatice řádu k matice \mathbf{A} . Indukcí podle k dokážeme, že $\det \mathbf{B} \in \{-1, 0, 1\}$.

1. $k = 1$: pak evidentně $\det \mathbf{B} \in \{-1, 0, 1\}$, neboť je to jednoprvková podmatice.
2. Necht věta platí pro všechny čtvercové podmatice řádu menšího než k , necht \mathbf{B} je matice řádu k .

Jsou následující tři možnosti.

1. Jestliže \mathbf{B} má některý sloupec nulový, potom je zřejmě $\det \mathbf{B} = 0$.
2. Jestliže \mathbf{B} má v některém sloupci (např. v i -tém) jediný nenulový prvek $a_{ij} = \pm 1$, pak rozvojem determinantu matice \mathbf{B} podle i -tého sloupce získáme $\det \mathbf{B} = (-1)^{i+j} a_{ij} \det \mathbf{B}'$, kde $\det \mathbf{B}'$ je determinant matice \mathbf{B}' , vzniklé z \mathbf{B} vynecháním i -tého řádku a j -tého sloupce. Determinant \mathbf{B}' je podle indukčního předpokladu roven některému z čísel $-1, 0, 1$, a protože $a_{ij} \in \{-1, 0, 1\}$, je i $\det \mathbf{B} \in \{-1, 0, 1\}$.
3. Jestliže \mathbf{B} má v každém sloupci právě dva nenulové prvky, pak součet všech řádků je nulový. Matice \mathbf{B} je tedy singulární, a proto $\det \mathbf{B} = 0$.

□

Věta 4.5. Incidenční matice orientovaného grafu \vec{G} je totálně unimodulární.

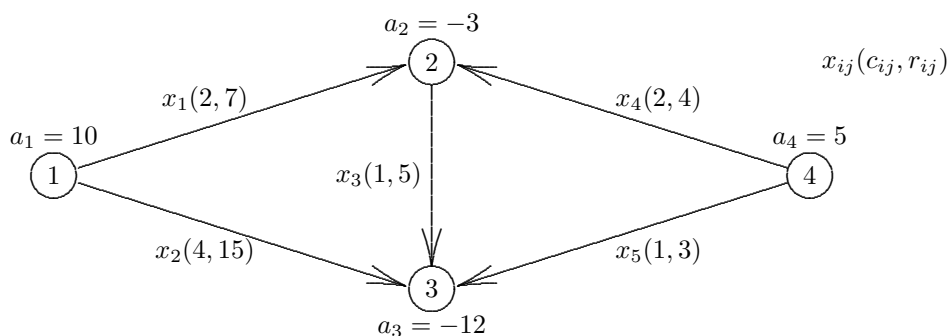
Důkaz. Incidenční matice orientovaného grafu má v každém sloupci právě jeden prvek rovný $+1$ a právě jeden prvek rovný -1 . Podle Věty 4.4 je tedy incidenční matice libovolného prostého orientovaného grafu totálně unimodulární. □

Následující věta shrnuje základní vlastnosti totálně unimodulárních matic. Větu uvádíme bez důkazu (indukcí), který je ponechán čtenáři jako cvičení.

Věta 4.6.

- (i) Je-li \mathbf{A} totálně unimodulární matice, pak matice $[\mathbf{A} | \mathbf{I}]$, kde \mathbf{I} je jednotková matice příslušného řádu, je také totálně unimodulární.
- (ii) Je-li \mathbf{A} totálně unimodulární matice, pak matice \mathbf{A}^T je rovněž totálně unimodulární.
- (iii) Je-li bloková matice $\mathbf{A} = [\mathbf{A}_1 | \mathbf{A}_2]$ totálně unimodulární, pak matice $\begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 \\ \pm \mathbf{I} & \mathbf{0} \end{bmatrix}$ je také totálně unimodulární.

Příklad. Vraťme se nyní k příkladu na převod úlohy nalezení optimálního toku na úlohu CLP z úvodu této kapitoly. Hledáme optimální tok v síti dle následujícího obrázku.



Úlohu jsme převedli na následující úlohu CLP:

$$\min 2x_1 + 4x_2 + x_3 + 2x_4 + x_5$$

za podmínek

$$\begin{array}{rcccccccl}
 x_1 & +x_2 & & & & & & = & 10 \\
 -x_1 & & +x_3 & -x_4 & & & & = & -3 \\
 & -x_2 & -x_3 & & -x_5 & & & = & -12 \\
 & & & x_4 & +x_5 & & & = & 5 \\
 x_1 & & & & & & & \leq & 7 \\
 & x_2 & & & & & & \leq & 15 \\
 & & x_3 & & & & & \leq & 5 \\
 & & & x_4 & & & & \leq & 4 \\
 & & & & x_5 & & & \leq & 3
 \end{array}$$

$$x_i \geq 0, \quad i = 1 \dots 5$$

V této soustavě je možno čtvrtou rovnici vyloučit, neboť je lineární kombinací předchozích třech. Ponecháme tak jen první tři řádky incidenční matice sítě. Matice soustavy pak bude mít následující tvar.

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & & \\ -1 & 0 & 1 & -1 & 0 & \mathbf{0} & \\ 0 & -1 & -1 & 0 & -1 & & \\ & & & \mathbf{I} & & & \mathbf{I} \end{bmatrix}$$

Tato matice má hodnost $m = 8$ a podle vět 4.5 a 4.6 je totálně unimodulární. Je zřejmé, že dané pozorování platí obdobně pro každou úlohu CLP, která vznikla převodem úlohy na optimální tok.

Závěr: Úlohu nalezení optimálního toku v síti lze převést na úlohu CLP s totálně unimodulární maticí a řešení lze tedy nalézt simplexovým algoritmem, který při celočíselných hodnotách intenzit a propustností dá celočíselný optimální tok.

Poznámka.

1. Na CLP lze převést i jiné úlohy, které však tuto výhodu nemusí mít (např. i úloha obchodního cestujícího a jiné).
2. Pro některé třídy úloh na optimální tok je převod na CLP výhodný; pro jiné však výhodněji pracují speciální grafové algoritmy.
3. Uvažujme následující otázku: je dána matice \mathbf{A} s prvky $0, 1, -1$; úkol je zjistit, zda \mathbf{A} je totálně unimodulární. V r. 1980 P. Seymour dokázal hlubokou strukturální charakterizační větu, dávající polynomiální test totální unimodularity (viz [1], odst. 6.5).

4.4 Enumerativní metody a metoda větví a mezí

V předchozím odstavci jsme viděli, že je-li matice soustavy totálně unimodulární, řešení úlohy CLP se výrazně zjednodušuje. Otázkou je, co dělat, pokud matice soustavy totálně unimodulární není. Je jasné, že máme co do činění s NP-těžkou úlohou, a snadné řešení není známo. Používané metody lze v podstatě rozdělit do dvou skupin:

- enumerativní metody,
- metody sečných nadrovin.

Enumerativní metody využívají toho, že všech PR je konečně mnoho a lze se tedy (při „nepříliš velkém“ rozměru úlohy) pokusit je probírat. Při probírce se zpravidla pokoušíme počet probíraných PR redukovat použitím technik typu *metody větvi a mezí* (angl. „branch and bound“). Podstata metody větvi a mezí spočívá v tom, že množina přípustných řešení se opakovaně větví na podmnožiny (*větvě*) a pro každou z nich se vypočítává *horní mez* (při úlohách na minimalizaci dolní mez) účelové funkce. Známe-li již přípustné řešení s hodnotou účelové funkce vyšší, než je horní mez pro některou větev, nemusíme tuto větev dále probírat, protože v ní optimum být nemůže. Detailní strategie užití metody je ovšem případ od případu různá a zpravidla silně závisí na konkrétní situaci (je tedy jedna *metoda* větvi a mezí, ale hodně *algoritmů* větvi a mezí).

Metody sečných nadrovin jsou založeny na následující myšlence.

- Řešíme úlohu v reálném oboru, tj. bez podmínky celočíselnosti.
- Je-li optimum celočíselné, jsme hotovi.
- Není-li optimum celočíselné, přidáme novou podmínku („sečnou nadrovinu“), která odsekne část PR, ale zachová všechna celočíselná PR. Po jistém počtu kroků („oříznutí“) bude řešení nové úlohy LP v reálném oboru celočíselné.

Postupy tohoto typu jsou známy jako *Gomoryho algoritmy* a budeme se jim blíže věnovat v následujícím odstavci.

Jako příklad použití metody větvi a mezí uvedeme následující úlohu.

Příklad. Úloha o batohu.

Máme n předmětů a u každého z nich známe jeho *hmotnost* a_i a *užitečnost* c_i , $i = 1, \dots, n$. Batoh má nosnost b . Úkolem je vzít do batohu předměty maximální celkové užitečnosti a nepřekročit přitom nosnost batohu. (V jiné, „praktičtější“ formulaci úlohy může jít o n možných investic, z nichž u každé známe náklady a_i a užitečnost c_i , a nesmíme překročit daný objem investičních prostředků b).

Formálně úlohu zapíšeme takto: jsou dány vektory $\mathbf{a} = [a_1, \dots, a_n]^T \in \mathbf{Z}^n$, $\mathbf{c} = [c_1, \dots, c_n]^T \in \mathbf{R}^n$ a číslo $b \in \mathbf{Z}$. Význam hledaného vektoru $\mathbf{x} = [x_1, \dots, x_n]^T \in \mathbf{Z}^n$ je

$$x_i = \begin{cases} 0, & \text{jestliže } i\text{-tý předmět není v batohu,} \\ 1, & \text{jestliže } i\text{-tý předmět je v batohu.} \end{cases}$$

Odpovídající úloha CLP pak je tvaru

$$\begin{aligned} \max \quad & \mathbf{c}^T \mathbf{x} \\ \mathbf{a}^T \mathbf{x} \leq & \mathbf{b} \\ x_i \in & \{0, 1\} \end{aligned}$$

Je to tedy speciální případ úlohy CLP, v němž

- složky vektoru \mathbf{x} mohou nabývat pouze hodnot 0 a 1 (pro tento speciální případ CLP bývá používán termín *binární programování*),
- matice \mathbf{A} má jediný řádek.

Způsob aplikace metody větvi a mezí ukážeme na konkrétním příkladu. Použitý postup se nazývá *Kolesarův algoritmus*.

předmět č. i	1	2	3	4	5	6	7
hmotnost a_i	40	50	20	10	20	40	30
užitečnost c_i	40	60	10	9	3	20	60

Nosnost batohu je $b = 100$.

Před vlastním řešením provedeme dva přípravné kroky.

1. *Ověření řešitelnosti*: $a_4 = 10 < 100 = b$ (alespoň jeden předmět se do batohu vejde).
2. *Ověření netriviálnosti*: $\sum_{i=1}^7 a_i = 210 > 100 = b$ (všechny předměty se do batohu nevejdou).

Základní myšlenka Kolesarova algoritmu je založena na tom, že s využitím skutečnosti, že $c_i > 0$ pro všechna $i = 1, \dots, 7$, vytvoříme podíly $\frac{c_i}{a_i}$ (které lze interpretovat jako „relativní užitečnosti“ jednotlivých předmětů), předměty nově uspořádáme podle velikosti těchto podílů, a hodnot podílů $\frac{c_i}{a_i}$ využijeme při výpočtu horních mezí jednotlivých větví.

Po přecíslování předmětů dostaneme:

nové číslo předmětu i	1	2	3	4	5	6	7
staré číslo předmětu	7	2	1	4	6	3	5
hmotnost a_i	30	50	40	10	40	20	20
užitečnost c_i	60	60	40	9	20	10	3
poměr $\frac{c_i}{a_i}$	2	1,2	1	0,9	0,5	0,5	0,15

Nyní postupně probíráme jednotlivé možnosti a pro každou z nich určíme horní mez. Každé přípustné řešení je popsáno binárním vektorem délky 7, jehož i -tá složka vyjadřuje, zda je i -tý předmět naložen v batohu (například vektor [1010000] znamená naložení předmětů 1 a 3). Znak \times v i -té složce znamená, že o i -tém předmětu není ještě rozhodnuto.

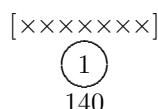
Postup probírání popisujeme *rozhodovacím stromem*. Značení uzlů:

[10×××××]	popis řešení (probíraná možnost)
(j)	pořadové číslo uzlu v rozhodovacím procesu
140	horní mez (pro celou větev, která z uzlu j vychází)

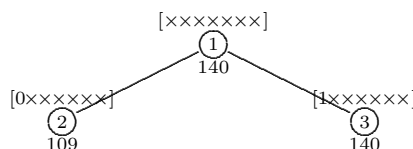
1. První uzel rozhodovacího stromu bude mít vektor řešení $\times\times\times\times\times\times$. Určení horní meze: naložíme-li (v pořadí poměrů $\frac{c_i}{a_i}$) předměty 1 a 2, zbývá volná kapacita batohu 20 – tu obsadíme „fiktivní půlkou“ předmětu 3. Náklad v batohu vypadá takto:

naložíme	1	2	fikt. půlka 3	Σ
hmotnost	30	50	20	100
užitečnost	60	60	20	140

Odhlédneme-li od nerealizovatelnosti takového řešení, je jisté, že (díky uspořádání podle $\frac{c_i}{a_i}$) více již do batohu nemůže jít naložit, tj. žádné řešení nemůže mít hodnotu $\mathbf{c}^T \mathbf{x}$ větší než 140. Horní mez je tedy 140, a máme první uzel rozhodovacího stromu:



2. První větvení.

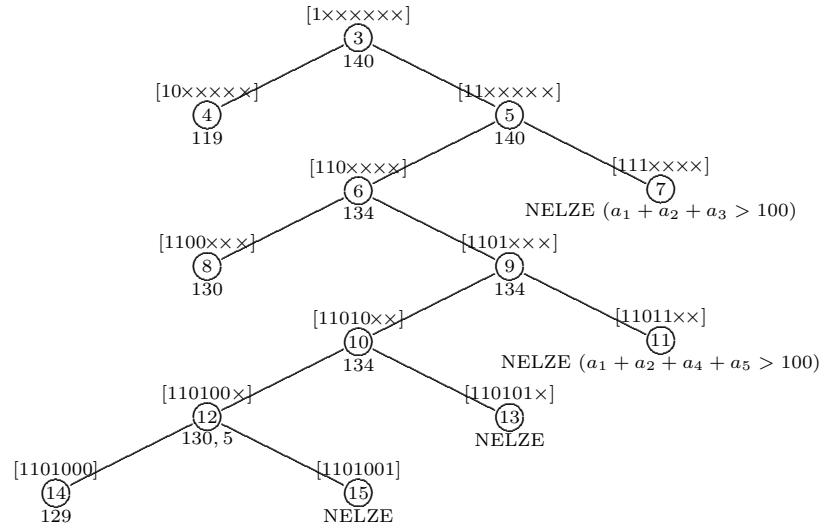


Horní meze: u uzlu 3 je horní mez evidentně stejná jako u uzlu 1, tj. 140; u uzlu 2 ji určíme takto.

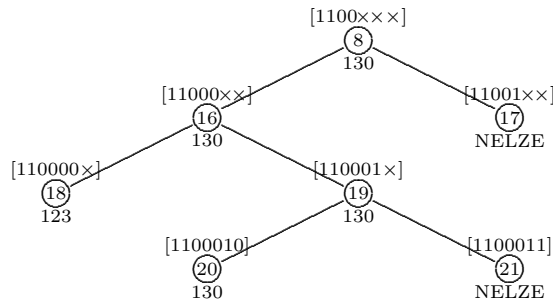
naložíme	2	3	4	Σ
hmotnost	50	40	10	100
užitečnost	60	40	9	109

3. Pro další větvení Kolesarův algoritmus používá empirické pravidlo větvení podle největší horní meze. Toto pravidlo samozřejmě nemusí vést nejrychleji k optimu (proč?³), ale podstatné je to, že jestliže aktuální největší horní meze odpovídá uzlu s vektorem řešení „bez ×“ (tj. konkrétní přípustné řešení), pak zbývající větve s menší horní mezí už není třeba procházet, protože v nich optimum být nemůže. Při nalezení přípustného řešení, jehož hodnota účelové funkce je větší nebo rovna horním mezím všech ještě neprobraných větví, výpočet končí.

Proces dalšího větvení z uzlu 3 pak vypadá takto.



V tomto okamžiku již známe konkrétní řešení s $\mathbf{c}^T \mathbf{x} = 129$, ale 129 není největší horní meze (viz uzel 8). Proto uzel 14 nemusí být optimem, a pro nalezení optima ještě musíme větvit z uzlu 8.



Hodnota 130 účelové funkce nalezeného řešení je nyní již větší než horní meze všech dosud neprohledaných větví. Nalezené řešení v uzlu 20, tj. [1100010], je tedy optimální a výpočet končí.

Povšimněme si toho, že explicitní probírání (hrubou silou) by znamenalo zkoumat $2^7 = 128$ možností, zatímco s použitím metody větví a mezí jsme počet probíraných možností zredukovali na 21.

4.5 Metoda sečných nadrovin

Mějme dānu ůlohu CLP ve tvaru

$$\begin{aligned} \min \quad & \mathbf{c}^T \mathbf{x} \\ \mathbf{A} \mathbf{x} = & \mathbf{b} \\ x_i \geq & 0 \quad \text{celoěiselné.} \end{aligned}$$

Předpokládāme, že všechny prvky matice \mathbf{A} a vektoru \mathbf{b} jsou celoěiselné.

³Kdyby ano, měli bychom polynomiālně algoritmus na řešení NP-tězkěho problěmu a byli bychom slavněi.

Odpovídající úloha LP v reálném oboru, tj. tatáž úloha, ale bez podmínky celočíselnosti, se nazývá *slabá úloha*, příslušná naší úloze.

První krok metody sečných nadrovin spočívá v tom, že spočítáme (simplexovým algoritmem) řešení příslušné slabé úlohy LP. Necht' výsledná simplexová tabulka je (po ev. přečíslování proměnných tak, aby první proměnné byly bazické) tvaru

$$\left[\begin{array}{cccc|ccc} 1 & 0 & \dots & 0 & a_{1\ m+1} & \dots & a_{1n} & | & x_1^* \\ 0 & 1 & \dots & 0 & a_{2\ m+1} & \dots & a_{2n} & | & x_2^* \\ \vdots & \vdots & \dots & \vdots & \vdots & \dots & \vdots & | & \vdots \\ 0 & 0 & \dots & 1 & a_{m\ m+1} & \dots & a_{mn} & | & x_m^* \end{array} \right] \quad (*)$$

To znamená, že bod $\mathbf{x}^* = [x_1^*, \dots, x_m^*, 0, \dots, 0]^T$ je optimální řešení slabé úlohy. Bod \mathbf{x}^* tedy splňuje rovnost

$$\mathbf{Ax} = \mathbf{b},$$

a stejně tak rovnosti dané tabulkou (*).

Ma-li původní úloha CLP nějaké *celočíselné přípustné* řešení \mathbf{x}^c , pak pro něj také platí rovnosti (*) (neboť \mathbf{x}^c splňuje rovnost $\mathbf{Ax}^c = \mathbf{b}$, a (*) z ní vznikly Jordanovou eliminací).

Zdůrazněme zde, že \mathbf{x}^c nemusí být bazické (a také zpravidla nebude), tj. může mít více než m nenulových složek.

Rozepíšeme-li rovnosti (*) pro \mathbf{x}^c , pak i -tý řádek těchto rovností je tvaru

$$x_i^c + a_{i\ m+1}x_{m+1}^c + \dots + a_{in}x_n^c = x_i^*. \quad (**)$$

Pro $a \in \mathbf{R}$ označíme $[a]$ celou část čísla a a a' necelou část čísla a (tj. $a = [a] + a'$, $[a] \in \mathbf{Z}$ a $0 < a' < 1$). Z rovnosti (**) plyne, že necelé části výrazů nalevo a napravo si musí být rovny. Protože x_i^c je celé číslo, neovlivní necelou část výrazu vlevo, a tedy platí

$$(a_{i\ m+1}x_{m+1}^c + \dots + a_{in}x_n^c)' = (x_i^*)'.$$

Z rovností (**) vybereme tu, pro kterou je $(x_i^*)'$ největší (protože \mathbf{x}^* není celočíselný, existuje takové i , pro něž je $(x_i^*)' > 0$). Protože složky \mathbf{x}^c jsou celá čísla, musí platit

$$a'_{i\ m+1}x_{m+1}^c + \dots + a'_{in}x_n^c = (x_i^*)' + N,$$

kde N je vhodné celé číslo, neboli

$$a'_{i\ m+1}x_{m+1}^c + \dots + a'_{in}x_n^c \geq (x_i^*)'. \quad (***)$$

Protože tuto podmínku splňuje každé celočíselné PR, jejím přidáním k naší úloze žádné celočíselné PR neztratíme. Podmínka (***) je tedy hledaná „ořezávací podmínka“.

Příklad.

$$\begin{array}{l} \min -x - y + z \\ 4x + 3y + 2z = 7 \\ x, y, z \geq 0 \quad \text{celočíselné.} \end{array}$$

Řešíme nejprve příslušnou slabou úlohu. S výchozím PBR nejsou problémy - volíme-li např. x jako bazickou, stačí dělit čtyřmi a dostaneme následující výchozí simplexovou tabulku.

1	1	-1	0
1	$\frac{3}{4}$	$\frac{1}{2}$	$\frac{7}{4}$

Simplexovým algoritmem postupně vypočítáváme:

0	$\frac{1}{4}$	$-\frac{3}{2}$	$-\frac{7}{4}$
1	$\frac{3}{4}$	$\frac{1}{2}$	$\frac{7}{4}$

0	$\frac{1}{4}$	$-\frac{3}{2}$	$-\frac{7}{4}$
$\frac{4}{3}$	1	$\frac{2}{3}$	$\frac{7}{3}$

$-\frac{1}{3}$	0	$-\frac{5}{3}$	$-\frac{7}{3}$
$\frac{4}{3}$	1	$\frac{2}{3}$	$\frac{7}{3}$

Optimální řešení slabé úlohy tedy je

$$\mathbf{x}^* = [0, \frac{7}{3}, 0]^T.$$

Platí $(\frac{4}{3})' = \frac{1}{3}$ a $(\frac{7}{3})' = \frac{1}{3}$, a tedy přídatná podmínka (***) je tvaru

$$\frac{1}{3}x + \frac{2}{3}z \geq \frac{1}{3},$$

neboli po úpravě

$$x + 2z \geq 1.$$

Tuto podmínku přidáme k původní úloze, samozřejmě jako rovnost s novou pomocnou proměnnou, tj. ve tvaru

$$x + 2z - u = 1.$$

Vzniklou úlohu opět řešíme simplexovým algoritmem:

1	1	-1	0	0
4	3	2	0	7
1	0	2	-1	1

1	1	-1	0	0
0	3	-6	4	3
1	0	2	-1	1

1	1	-1	0	0
0	1	-2	$\frac{4}{3}$	1
1	0	2	-1	1

0	0	-1	$-\frac{1}{3}$	-2
0	1	-2	$\frac{4}{3}$	1
1	0	2	-1	1

Optimálním řešením této úlohy je vektor $\mathbf{x} = [1, 1, 0, 0]^T$. Protože je celočíselný, hledaným optimálním řešením původní úlohy CLP je vektor $\mathbf{x}^c = [1, 1, 0]^T$.

Poznámka.

1. Počet nutných „oříznutí“ může být velký, a každým oříznutím se zvětšuje rozměr úlohy. Proto se při praktickém použití zpravidla počet iterací limituje, a když metoda nedá výsledek, zkouší se jiné postupy.

2. Uvedený postup je znám jako (první) *Gomoryho algoritmus*.
3. Postupy užívající metodu sečných nadrovin zpravidla nejsou příliš vhodné pro binární úlohy.
4. Praktickou stránku algoritmu lze ještě dost výrazně zjednodušit. V elementární podobě, v níž jsme metodu uvedli, totiž po přidání ořezávací podmínky vektor \mathbf{x}^* není PR, takže se při každé iteraci musí proběhnout celý simplexový algoritmus od začátku. To lze odstranit postupy, při nichž se současně řeší i duální úloha (tzv. duální simplexová metoda), protože \mathbf{x}^* zůstane duálně přípustné.

5 Párování

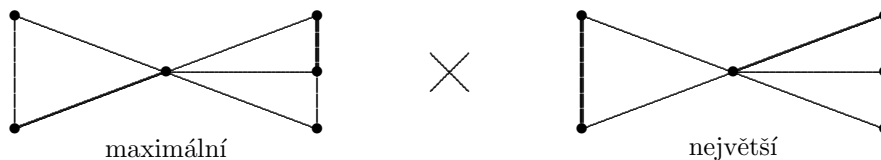
V celé této kapitole se budeme zabývat neorientovanými grafy; termín „graf“ tedy vždy znamená neorientovaný graf.

5.1 Základní pojmy, perfektní párování

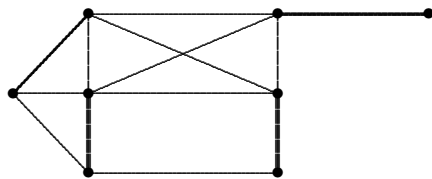
Definice 5.1. *Pravidelný podgraf stupně 1 grafu G se nazývá párování (angl. „matching“) v grafu G . Pravidelný faktor stupně 1 grafu G se nazývá perfektní párování v grafu G . Párování v grafu G , mající největší možný počet hran, se nazývá největší párování v grafu G .*

Poznámka. 1. Anglická terminologie: párování = matching, perfektní párování = perfect matching.

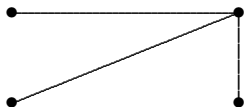
2. Opět si musíme uvědomit rozdíl mezi pojmy „maximální“ a „největší“ (angl. „maximal“ a „maximum“). Příklad je na následujícím obrázku, v němž jsou hrany párování vyznačeny silněji (maximální párování nemusí být nutně největší). Maximální párování (nějaké) lze samozřejmě snadno nalézt hládovým algoritmem, ale pro nalezení největšího párování je třeba použít jemnější metody (i když, jak uvidíme, je tato úloha ze třídy P, tj. řešitelná v polynomiálním čase i paměti).



Graf na následujícím obrázku je příkladem grafu, v němž existuje perfektní párování.



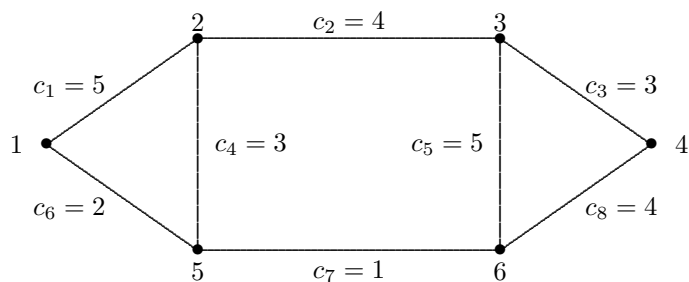
První a samozřejmou nutnou podmínkou existence perfektního párování je sudý počet uzlů grafu. Tato podmínka však zdaleka není postačující, což si můžeme ověřit na jednoduchém příkladu:



Příklady. Ukažme si několik příkladů aplikací pojmu párování a perfektního párování.

1. Představme si hudební skupinu na turné. Úkolem je ubytovat členy kapely v hotelu na dvouúžkových pokojích. V grafu uzly budou odpovídat členům kapely a hrany budou spojovat uzly těch členů kapely, kteří mohou mít společný pokoj. Je samozřejmě nutno ubytovat všechny členy, a to tak aby byli na pokojích vždy ve dvojici, která si sobě nevadí.
2. Mějme dānu množinu X chlapců a množinu Y dēvčat. Tyto množiny X, Y budou představovat uzly bipartitního grafu. Hranami spojíme ty dvojice, kdy chlapec x znā dívku y . Úkolem je spárovat všechny dēvčata i chlapce, např. v tanečních.
3. Pripomeňme si problém z předmětu TGD1: čtvercovā matice A je genericky regulārni, prāvě když její bigraf mā perfektní párování. Obecněji, generická hodnota matice A je rovna velikosti největšího párování v jejím bigrafu.
4. Jsou dāny dvě množiny X a Y . Množina X bude představovat množinu profesí a množina Y pracovníky. Sestrojíme bipartitni graf s množinou uzlů $U(G) = X \cup Y$ množinou hran tvořenou všemi dvojicemi $\{x, y\}$, pro něž pracovník $y_j \in Y$ umí profesi $x_i \in X$. Navíc je dāno hranové ohodnocení w_{ij} , mající význam efektivity pracovníka y_j v profesi x_i .
Úkolem je nalézt *optimální přiřazení* pracovníků profesím (tzv. *optimum assignment problem*). Úloha vede, jak již asi je patrné, na párování v bipartitním grafu, ale v ohodnoceném, a s podmínkou maximalizace sumy ohodnocení.
5. Obecnā formulace předchozí úlohy: je dān ohodnocený neorientovaný graf; úkolem je nalézt *optimální párování*, tj. perfektní párování M , které maximalizuje výraz $\sum_{h_i \in H(M)} w(h_i)$.

Příklad. Ukažme si na příkladu, jak lze úlohu nalezení optimálního párování převést na úlohu celočíselného lineárního programování. Mějme dān následující graf, kde c_i je ohodnocení i -té hrany h_i .



Pro párování M položíme $x_i = 0$, jestliže $h_i \notin H(M)$, a $x_i = 1$, pokud $h_i \in H(M)$. Úlohu můžeme zapsat jako úlohu celočíselného lineárního programování následujícím způsobem.

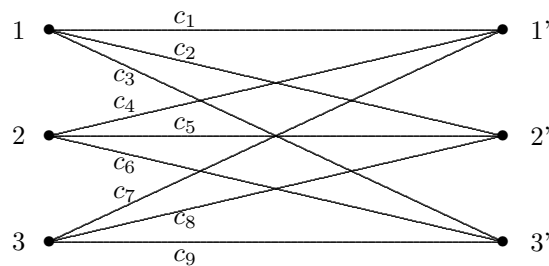
$$\begin{aligned} \max & 5x_1 + 4x_2 + 3x_3 + 3x_4 + 5x_5 + 2x_6 + x_7 + 4x_8 \\ & x_1 & & & & & & +x_6 & & & & & & = 1 \\ & x_1 & +x_2 & & & & & & +x_4 & & & & = 1 \\ & & x_2 & +x_3 & & & & & & +x_5 & & & = 1 \\ & & & & x_3 & & & & & & & +x_8 & = 1 \\ & & & & & x_4 & & & +x_6 & +x_7 & & & = 1 \\ & & & & & & x_5 & & & +x_7 & +x_8 & & = 1 \\ & x_i \in \{0, 1\}, & i = 1, \dots, 8. \end{aligned}$$

Je zřejmé, že se jedná dokonce o úlohu binárního CLP. Matice podmínek této úlohy je tvaru

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}.$$

Je zřejmé, že \mathbf{A} je uzlo-hranovou incidenční maticí grafu G . Bohužel, protože ale G je neorientovaný, matice \mathbf{A} obecně není totálně unimodulární (v našem případě stačí např. uvažovat determinant čtvercové podmatice řádu 3 sestavené z řádků 1, 2, 5 a sloupců 1, 4, 6, který je roven dvěma). K řešení tedy nemůžeme použít simplexový algoritmus.

Příklad. Další příklad nám ukáže, že v případě bipartitního grafu bude situace mnohem snazší. Mějme tedy následující bipartitní graf (čísla c_i jsou opět ceny hran).



Protože je náš graf bipartitní, můžeme jeho hrany orientovat od $X = \{1, 2, 3\}$ k $Y = \{1', 2', 3'\}$. Podle vzoru z předchozího příkladu převedeme úlohu nalezení optimálního párování na úlohu CLP. Cenová funkce bude mít stejný tvar, tedy $\sum_{v_i} c_i x_i$, a budeme ji maximalizovat. Vzhledem k orientaci budou mít podmínky pro uzly grafu následující tvar.

$$\begin{array}{cccccccccc} x_1 & +x_2 & +x_3 & & & & & & & & = & 1 \\ & & & x_4 & +x_5 & +x_6 & & & & & = & 1 \\ & & & & & & x_7 & +x_8 & +x_9 & & = & 1 \\ -x_1 & & & -x_4 & & & -x_7 & & & & = & -1 \\ & -x_2 & & & -x_5 & & & -x_8 & & & = & -1 \\ & & -x_3 & & & -x_6 & & & -x_9 & & = & -1 \end{array}$$

$x_i \in \{0, 1\}, i = 1, \dots, 9.$

Matice podmínek je opět uzlo-hranová incidenční matice, tentokrát však orientovaného bipartitního grafu.

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ -1 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & 0 & -1 & 0 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & -1 \end{bmatrix}$$

Podle Věty 4.5 je matice \mathbf{A} totálně unimodulární. Vidíme, že v bipartitním případě je úloha značně jednodušší.

Poznámka. Obdobný postup je možno použít i pro nalezení největšího párování. Podmínky by pak říkaly, že každý uzel musí být incidentní s nejvýše jednou hranou párování – rovnosti v podmínkách bychom tedy nahradili nerovnostmi „nejvýše jedna hrana“ (např. $x_1 + x_2 + x_3 \leq 1$), ceny hran bychom uvažovali rovny jedné. Maximalizací cenové funkce bychom získali počet hran největšího párování.

V následujících oddílech budeme postupně vyšetřovat jednodušší případy obecné úlohy optimálního párování, a to v pořadí:

- největší párování v neohodnoceném bipartitním grafu,
- optimální párování v ohodnoceném bipartitním grafu,
- největší párování v neohodnoceném grafu (obecně nebipartitním).

Uvidíme, že na všechny tyto úlohy existují algoritmy, které naleznou řešení v polynomiálním čase. Případ ohodnocených neorientovaných (obecně nebipartitních) grafů je značně netriviální a nebudeme se jím zabývat, i na něj však existují polynomiální algoritmy, založené na využití grafových vlastností úlohy.

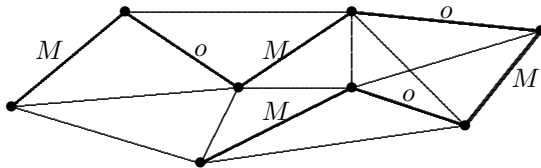
Nejprve ale vybudujeme základní důkazové techniky a pomocí nich dokážeme charakterizační větu největšího párování.

5.2 Rozšiřující cesty a Bergeova věta

Definice 5.2. Necht' G je graf a M párování v G . Řekneme, že párování M pokrývá uzel x , jestliže existuje hrana $h \in H(M)$ taková, že $x \in h$.

Definice 5.3. Necht' G je graf, M párování v G a P cesta v G ($H(M) \neq \emptyset$, $H(P) \neq \emptyset$). Řekneme, že cesta P je M -alternující, jestliže její hrany střídavě jsou a nejsou v párování M .

Příklad.



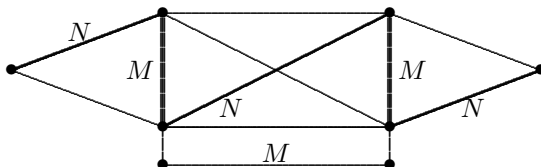
Hrany vyznačené tučnou čarou ukazují M -alternující cestu, hrany této cesty označené M jsou hrany párování, hrany označené o jsou hrany mimo párování.

Definice 5.4. Necht' G je graf, M párování v G a P cesta v G . Řekneme, že cesta P je M -rozšiřující, jestliže P je M -alternující cestou a oba její koncové uzly nejsou M -pokryty.

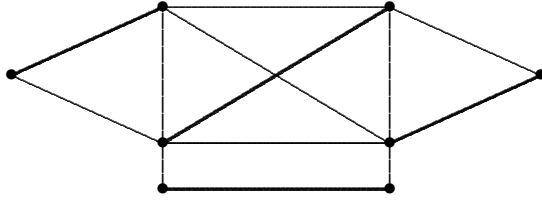
Poznámka. 1. Nejjednodušším případem M -rozšiřující cesty je hrana, neležící v M .

2. Cesta P v minulém příkladu je o -rozšiřující.

Příklad. V grafu na následujícím obrázku je cesta P vyznačena silně, hrany označené M patří do párování, hrany označené N jsou hranami M -rozšiřující cesty P , ale nepatří do párování M .



Jestliže podél M -rozšiřující cesty P prohodíme párování M s množinou hran N , získáme opět párování, které ale má o jednu hranu více.



(Silně vyznačené hrany jsou hranami nového většího párování.)

Tento příklad ukazuje, že pokud v grafu existuje M -rozšiřující cesta, lze po ní párování M zvětšit. Jinak řečeno, neexistence M -rozšiřující cesty v grafu je *nutnou* podmínkou toho, aby párování M bylo největší. Následující věta nám ukáže, že je to i podmínka postačující.

Věta 5.1. Berge (1957) Párování M je největším párováním v grafu G , právě když v G neexistuje M -rozšiřující cesta.

Důkaz. Dokážeme ekvivalentní tvrzení, že v grafu G existuje M -rozšiřující cesta, právě když párování M není největší.

1. Nechť tedy v G existuje M -rozšiřující cesta. Pak lze po této cestě párování M zvětšit, a tedy M není největší.
2. Nechť naopak párování M není největším párováním, tj. nechť existuje nějaké větší párování N . Platí tedy $|H(M)| < |H(N)|$. Nechť G' je podgraf určený těmi hranami, které leží v právě jednom z párování M a N , tj. v symetrické diferenci množin $H(M)$ a $H(N)$. Pak každý uzel podgrafu G' má v G' stupeň 1 nebo 2 (větší stupeň uzlu je vyloučen, neboť každé párování je pravidelný podgraf stupně 1 a dvě párování tedy mohou dát stupeň uzlu nejvýše 2). Odtud plyne, že každá komponenta grafu G' je kružnice sudé délky (neboť hrany kružnice jsou střídaně v M a v N) nebo cesta. Pokud by všechny komponenty G' byly kružnice (jen sudých délek, jak bylo ukázáno), nebo cesty sudých délek, platilo by $|M| = |N|$, což je spor. Tedy nutně musí v G' existovat cesta P liché délky, která začíná i končí v N . Tedy P je M -rozšiřující cestou. □

Poznámka. 1. Symetrická diference dvou množin A, B je množina všech prvků, z $A \cup B$, které leží v právě jedné z množin A, B . Značí se $A \Delta B$ a platí pro ni $A \Delta B = (A \cup B) \setminus (A \cap B) = [(A \setminus B) \cup (B \setminus A)]$.

2. Bergeova věta sice je charakterizační větou, ale není to dobrá charakteristika, protože v případě, že M je největší, svazuje neexistenci většího párování s neexistencí M -rozšiřující cesty.

5.3 Největší párování v bipartitním grafu a maďarská metoda

Množinu uzlů bipartitního grafu lze rozdělit na dvě disjunktní podmnožiny X a Y tak, že $X \cup Y = U(G)$ a pro každou hranu uv je $u \in X$ a $v \in Y$, nebo $u \in Y$ a $v \in X$. Proto u bipartitního grafu budeme stručně psát $G = (X, Y)$.

Definice 5.5. Nechť $G = (X, Y)$ je bipartitní graf, $S \subset X$. Množina uzlů

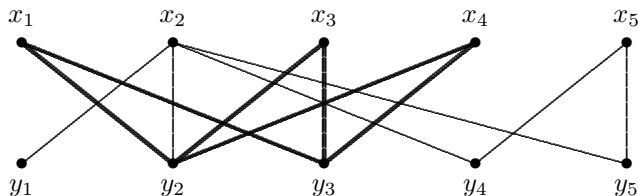
$$N(S) = \{y \in Y \mid \exists x \in S \text{ tak, že } \{x, y\} \in H(G)\}$$

se nazývá okolí množiny uzlů S .

Poznámka. Tedy, pro $S \subset X$ je $N(S) \subset Y$. Symetricky, pro $S \subset Y$ je $N(S) \subset X$.

Je zřejmé, že pokud $|X| \neq |Y|$, pak v bipartitním grafu nemůže existovat perfektní párování. Omezení na grafy s $|X| = |Y|$ je však příliš silné. Z aplikací, uvedených v úvodu této kapitoly, je vidět, že je užitečné ptát se na existenci párování, pokrývajících všechny uzly v X , pokud $|X| \leq |Y|$. Tato triviálně nutná podmínka ale zdaleka není postačující.

Příklad. Najděte párování pokrývajících všechny uzly v X v následujícím grafu B .



V tomto grafu párování pokrývajících X neexistuje, protože pro množinu uzlů $S = \{x_1, x_3, x_4\}$ (viz tučně vyznačený podgraf grafu B) platí $|N(S)| < |S|$. Lze tedy pokrýt nejvýše $|N(S)|$ uzlů z S .

Z předchozího příkladu je vidět, že další nutnou podmínkou existence párování, pokrývajících celou množinu X , je, aby pro každou podmnožinu $S \subset X$ platilo $|N(S)| \geq |S|$. Následující věta říká, že tato podmínka je i postačující.

Věta 5.2. Hall (1935) *Bipartitní graf $G = (X, Y)$ má párování pokrývajících všechny uzly z X , právě když pro každou množinu uzlů $S \subset X$ je $|N(S)| \geq |S|$.*

Důkaz.

- Jestliže pro některou $S \subset X$ je $|N(S)| < |S|$, pak zřejmě v S lze párováním pokrýt nejvýše $|N(S)|$ uzlů. Párování pokrývajících X tedy nemůže existovat.
- Nechť naopak platí $|N(S)| \geq |S|$ pro každou podmnožinu $S \subset X$, nechť M je největší párování v grafu G , a předpokládejme, že M nepokrývá uzel $u \in X$. Označme Z množinu všech uzlů, do nichž vede z uzlu u M -alternující cesta, S množinu $S = Z \cap X$, T množinu $T = Z \cap Y$. Protože párování M je největší, je v množině Z jediný nepokrytý uzel, a to právě uzel u – jinak by totiž podle Bergeho věty nebylo M největším párováním. To znamená, že všechny uzly v množině T jsou pokryty, tedy množina uzlů T je spárována s množinou uzlů $S \setminus \{u\}$. Je zřejmé, že platí $|T| = |S| - 1$, ale množina T je vlastně okolím množiny S , a tedy $|N(S)| = |S| - 1$, tj. $|N(S)| < |S|$, což je spor s předpokladem věty. □

Poznámka. 1. Prakticky věta např. říká, že množinu X zaměstnanců lze zaměstnat právě tehdy, pokud každá skupina $S \subset X$ pracovníků umí alespoň $|S|$ profesí.

2. Hallova věta (na rozdíl od Bergeovy věty) je dobrou charakteristikou:

- existuje-li párování, pokrývajících X , najdeme ho,
- neexistuje-li párování, pokrývajících X , najdeme množinu $S \subset X$, pro kterou je $|N(S)| < |S|$.

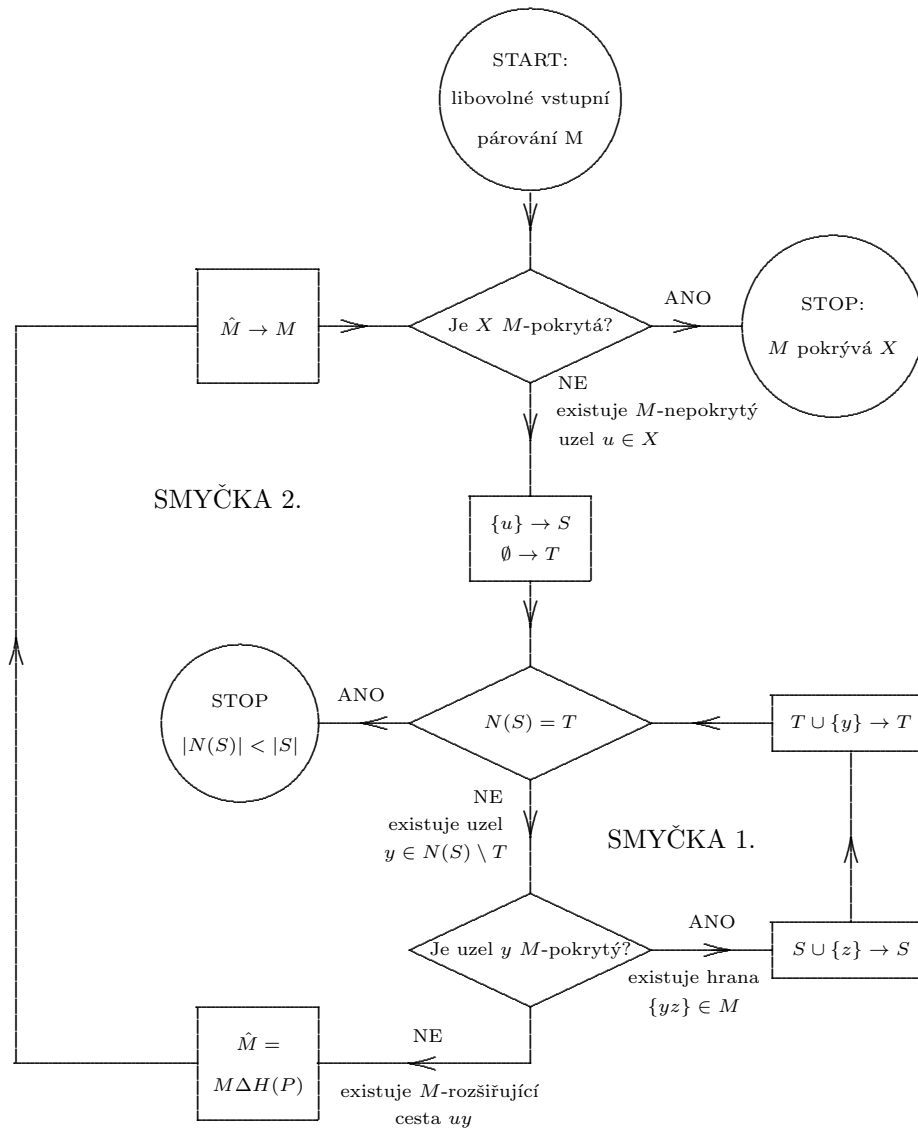
Otázka existence párování, pokrývajících X , tedy je v $\text{NP} \cap \text{co-NP}$.

Myšlenka důkazu Hallovy věty je základem algoritmu na nalezení párování, pokrývajících množinu X . Tento algoritmus spočívá v systematickém vyhledávání M -rozšiřujících cest a je znám jako *mad'arská metoda*.

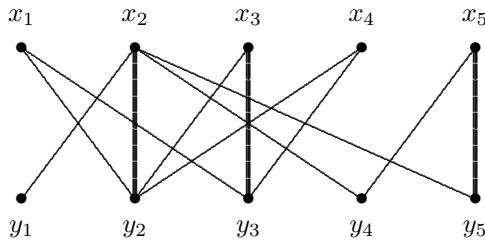
Algoritmus (maďarská metoda)

1. Inicializace - zvol libovolné párování M v G (stačí i jedna hrana).
2. Pokud M pokrývá celou množinu X , KONEC.
Jinak najdi nepokrytý uzel $u \in X$, polož $S := \{u\}$, $T = \emptyset$.
3. Pokud $N(S) = T$, pak $|N(S)| = |T| = |S| - 1$, STOP, hledané párování neexistuje.
Jinak najdi uzel $y \in N(S) \setminus T$.
4. Je-li uzel y M -pokrytý, pak existuje uzel $z \in X$, který je s uzlem y spárovaný.
Polož $S := S \cup \{z\}$, $T := T \cup \{y\}$, jdi na 3.
Jinak najdi M -rozšiřující cestu P z u do y , sestroj párování M' prohozením hran z M podél cesty P , polož $M := M'$, jdi na 2.

Vývojový diagram maďarské metody je uveden na následujícím obrázku.



Příklad. V grafu G najděte párování pokrývající všechny uzly množiny X .



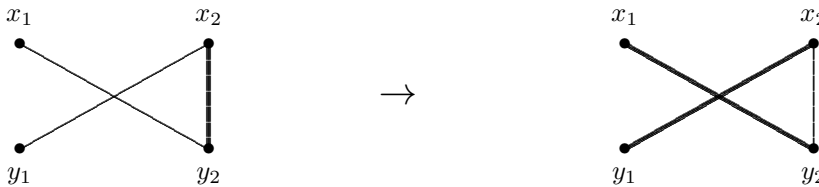
Výchozí párování M pro maďarskou metodu je vyznačeno silně.

Hledáme M -alternující cesty - viz smyčka 1 ve vývojovém diagramu.

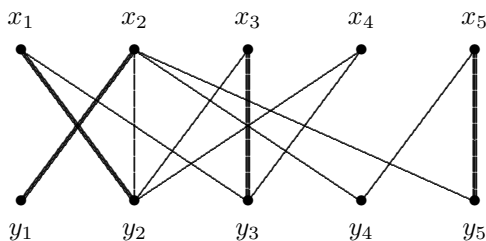
	S	T	y	z
1. krok	x_1	\emptyset	y_2	x_2
2. krok	x_1, x_2	y_2	y_1	-

Pro jednoznačnost popisu běhu algoritmů v příkladech se dohodneme, že v případě, že některé podmínce výběru vyhovuje více uzlů, se vždy řídíme pravidlem „od nejmenších indexů“.

Postupem uvedeným v tabulce jsme získali M -rozšiřující cestu x_1, y_2, x_2, y_1 . Nyní na této cestě prohodíme hrany párování s hranami mimo párování - viz následující obrázek, kde silné hrany patří párování.



Sestrojíme tedy nové párování.



Dále začneme s nepokrytým uzlem x_4 a použití maďarské metody popíšeme v následující tabulce.

S	T	y	z
x_4	\emptyset	y_2	x_1
x_4, x_1	y_2	y_3	x_3
x_4, x_1, x_3	y_2, y_3	-	-

Z tabulky je vidět, že $N(S) = T$. To však znamená, že hledané párování neexistuje, neboť zřejmě platí $|N(S)| < |S|$.

Poznámka. 1. Snadnou úvahou zjistíme, že výpočetní složitost maďarské metody v podobě, v jaké ji zde uvádíme, je $O(n^3)$: smyčka 2 se probíhá nejvýše $O(n)$ -krát, a při každém jejím průběhu má největší složitost, $O(n^2)$, vyhledávání M -rozšiřující cesty (smyčka 1 se probíhá jen nejvýše $O(n)$ -krát). Celkem tedy dostáváme odhad $O(n^3)$.

2. Maďarská metoda v podobě, v jaké ji uvádíme, najde párování pokrývající X nebo řekne, že takové párování neexistuje. Jsou známy analogické postupy, které (se stejnou výpočetní složitostí) v negativním případě naleznou i největší párování. Nejlepší známé takové postupy mají výpočetní složitost $O(m\sqrt{n})$.

3. Alternativní metodou nalezení největšího párování v bipartitním grafu je převod úlohy na úlohu maximálního toku v síti. Podrobnosti tohoto postupu jsou uvedeny v odstavci 1.5. Při použití nejlepšího známého (Dinicova) algoritmu na maximální tok má tento přístup složitost $O(m\sqrt{n})$.

5.4 Optimální párování v ohodnocených bipartitních grafech

Příklad. Připomeňme si motivační příklad 4 z odst. 5.1 na str. 58: je dána množina pracovníků X , množina pracovních míst Y a čísla w_{ij} s významem efektivnosti i -tého pracovníka x_i v j -té profesi y_j ; hledáme přiřazení pracovníků na pracovní místa, které je optimální ve smyslu maximalizace celkové efektivnosti, tj. které maximalizuje sumu $\sum_{\{i,j\} \in H(G)} w_{ij}$. Je zřejmé, že popíšeme-li úlohu ohodnoceným

bipartitním grafem $G = (X, Y)$, v němž w_{ij} je ohodnocení hrany (x_i, y_j) , pak řešením úlohy je to perfektní párování v G , které maximalizuje součet ohodnocení w_{ij} .

Definice 5.6. Necht' $G = (X, Y)$ je ohodnocený bipartitní graf, který má perfektní párování. Potom perfektní párování M v G , pro které je výraz $\sum_{(x_i, y_j) \in M} w_{ij} = \sum_{h_i \in M} w(h_i)$ maximální, se nazývá optimální párování grafu G .

Poznámka. Z předpokladu existence perfektního párování v G vyplývá, že nutně $|X| = |Y|$.

Následující pojem bude klíčem k algoritmu na nalezení optimálního párování.

Definice 5.7. Necht' $\ell : X \cup Y \rightarrow R$ je uzlové ohodnocení bipartitního grafu $G = (X, Y)$. Řekneme, že ℓ je přípustné ohodnocení, jestliže pro všechny hrany $(x_i, y_j) \in H(G)$ platí

$$\ell(x_i) + \ell(y_j) \geq w_{ij}.$$

Pro každé přípustné ohodnocení ℓ sestrojme graf G_ℓ předpisem

$$G_\ell = (X \cup Y, \{(x_i, y_j) \in H(G) \mid \ell(x_i) + \ell(y_j) = w_{ij}\}).$$

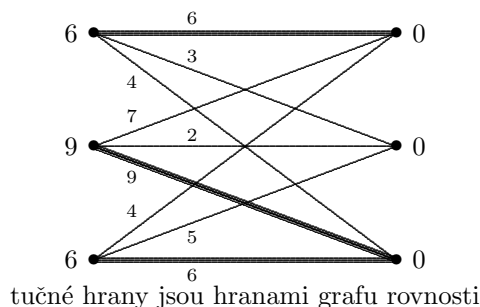
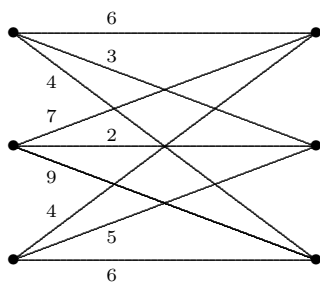
Graf G_ℓ se nazývá graf rovnosti, příslušný ohodnocení ℓ .

Poznámka. 1. Poznamenejme, že $U(G_\ell) = U(G)$, tj. G_ℓ je faktor grafu G .

2. Přípustné ohodnocení vždy existuje a lze ho získat např. předpisem

$$\begin{aligned} \ell(x_i) &= \max_{(x_i, y_j) \in H(G)} w_{ij} && \text{pro } x_i \in X, \\ \ell(y_j) &= 0 && \text{pro } y_j \in Y. \end{aligned}$$

Příklad. Na následujícím obrázku je příklad ohodnoceného bipartitního grafu, v pravé části obrázku je přípustné ohodnocení, sestrojené podle předchozího předpisu, a k němu příslušný graf rovnosti.



Následující věta ukazuje souvislost mezi přípustným ohodnocením a optimálním párováním.

Věta 5.3. *Nechť G je ohodnocený bipartitní graf a ℓ je přípustné ohodnocení uzlů G . Je-li M perfektním párováním v G_ℓ , pak je M optimálním párováním v G .*

Důkaz. Nechť ℓ je přípustné ohodnocení a M je perfektní párování v G_ℓ . Protože graf G_ℓ je faktorem grafu G , je M perfektní párování i v grafu G a platí pro něj

$$w(M) = \sum_{(x_i, y_j) \in M} w_{ij} = \sum_{u \in U(G)} \ell(u),$$

neboť každá hrana M patří do grafu rovnosti G_ℓ a každý uzel $u \in U(G)$ je v právě jedné hraně M . Pro kterékoliv jiné perfektní párování M' ale z přípustnosti ohodnocení ℓ plyne

$$w(M') = \sum_{(x_i, y_j) \in M'} w_{ij} \leq \sum_{u \in U(G)} \ell(u) = w(M),$$

a tedy párování M je optimální. □

Myšlenka důkazu věty 5.3 je základem následujícího algoritmu. Algoritmus je vlastně rozšířením maďarské metody, neboť body 1), 2) a 3) algoritmu jsou aplikací maďarské metody na graf rovnosti G_ℓ . Pokud se ukáže, že v G_ℓ perfektní párování neexistuje, pak se v (oproti maďarské metodě novém) bodu 4) přípustné ohodnocení vhodně modifikuje tak, aby do grafu rovnosti nového přípustného ohodnocení přibyla alespoň jedna hrana, vedoucí z množiny S do $Y \setminus T$, a algoritmus se vrací do maďarské metody v grafu rovnosti nového ohodnocení.

Algoritmus (Kuhn, Munkres)

START: Nalezneme libovolné přípustné ohodnocení ℓ a libovolné párování M v grafu rovnosti G_ℓ .

- 1) Jestliže M pokrývá X : KONEC.
Jinak: najdi nepokrytý uzel $u \in X$, polož $S := \{u\}$, $T := \emptyset$.
- 2) Jestliže $N_{G_\ell}(S) = T$, jdi na 4).
Jinak: nalezni uzel $y \in N_{G_\ell}(S) \setminus T$.
- 3) Je-li uzel y M -pokrytý, potom existuje uzel $z \in X$ takový, že $\{y, z\} \in H(M)$;
polož $S := S \cup \{z\}$, $T := T \cup \{y\}$, jdi na 2).
Jinak: najdi M -rozšiřující cestu P z u do y ; sestroj párování M' prohozením podél P ; polož $M := M'$ a jdi na 1).
- 4) Vypočítej číslo

$$\alpha_\ell = \min_{x_i \in S, y_j \notin T} \{\ell(x_i) + \ell(y_j) - w_{ij}\}$$

a nové přípustné ohodnocení

$$\ell'(v) = \begin{cases} \ell(v) - \alpha_\ell & \text{pro } v \in S, \\ \ell(v) + \alpha_\ell & \text{pro } v \in T, \\ \ell(v) & \text{jinak.} \end{cases}$$

Polož $\ell := \ell'$, $G_\ell := G_{\ell'}$, zvol uzel $y \in N_{G_\ell}(S) \setminus T$ a jdi na 3).

Poznámka. V bodu 4) algoritmu je číslo α_ℓ nutně kladné (neboť ℓ je přípustné ohodnocení a pro hrany nepatřící do grafu rovnosti je rozdíl nutně kladný), a snadno ověříme, že ℓ' je také přípustné ohodnocení a $N_{G_{\ell'}}(S) \setminus T \neq \emptyset$.

Příklad. Mějme ohodnocený bipartitní graf G , popsany maticí $W = [w_{ij}]$:

$$W = \begin{bmatrix} 3 & 5 & 5 & 4 & 1 \\ 2 & 2 & 0 & 2 & 2 \\ 2 & 4 & 4 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 2 & 1 & 3 & 3 \end{bmatrix}$$

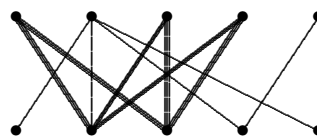
(tj. G je bigrafem matice W a nulové prvky znamenají, že příslušná dvojice uzlů není v grafu G spojena hranou).

Výchozí přípustné ohodnocení ℓ :

$$\begin{aligned} \ell(x_i) &= \max_{y_j \in Y} w_{ij} \quad \text{pro } x_i \in X, \\ \ell(y_j) &= 0 \quad \text{pro } y_j \in Y. \end{aligned}$$

Připíšeme ohodnocení uzlů k matici a nalezneme graf rovnosti (v následujících maticích bude vyznačen tučně).

$\ell(x_i) \backslash \ell(y_j)$	0	0	0	0	0
5	3	5	5	4	1
2	2	2	0	2	2
4	2	4	4	1	0
1	0	1	1	0	0
3	1	2	1	3	3



Nyní pomocí maďarské metody budeme hledat perfektní párování v grafu rovnosti. Je ihned vidět, že v G_ℓ perfektní párování neexistuje a algoritmus skončí při $S = \{x_1, x_3, x_4\}$, $T = \{y_2, y_3\}$. Proto dále určíme hodnotu α_ℓ :

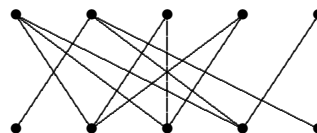
$$\alpha_\ell = \min_{x_i \in S, y_j \notin T} \{\ell(x_i) + \ell(y_j) - w_{ij}\}.$$

Zjistíme, že $\alpha_\ell = 1$. Nové přípustné ohodnocení je

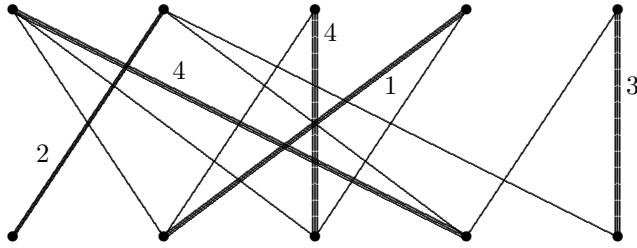
$$\ell'(v) = \begin{cases} \ell(v) - \alpha_\ell & \text{pro } v \in S, \\ \ell(v) + \alpha_\ell & \text{pro } v \in T, \\ \ell(v) & \text{jinak.} \end{cases}$$

Uurčíme nový graf rovnosti $G_{\ell'}$ (opět je vyznačen v matici tučně a je nakreslen na obrázku vpravo).

$\ell'(x_i) \backslash \ell'(y_j)$	0	1	1	0	0
4	3	5	5	4	1
2	2	2	0	2	2
3	2	4	4	1	0
0	0	1	1	0	0
3	1	2	1	3	3



Maďarská metoda v $G_{\ell'}$ nám již nyní dá perfektní párování:



Toto párování M (na obrázku vyznačeno tučně) je optimálním párováním v grafu G a $w(M) = 14$. Poznamenejme, že v G_{ℓ} existuje více perfektních párování, ale všechna mají stejnou cenu a jsou tedy všechna optimální.

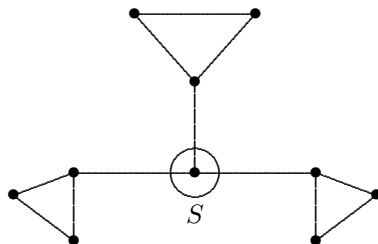
5.5 Perfektní párování v neorientovaném grafu a Tutteova věta

V tomto a následujících dvou odstavcích se budeme zabývat párováním v neorientovaném grafu, který již nemusí být bipartitní. Vyjdeme z Bergeovy věty (věta 5.1), která říká, že párování M je největší, právě když pro něj v G neexistuje rozšiřující cesta. Jak jsme již poznamenali, Bergeova věta není dobrá charakteristika, a právě nalezení dobré charakteristiky největšího párování bude naším prvním cílem.

Příklad. Uvedme si následující příklady a zkusme se zamyslet nad tím, proč tyto grafy nemají perfektní párování.

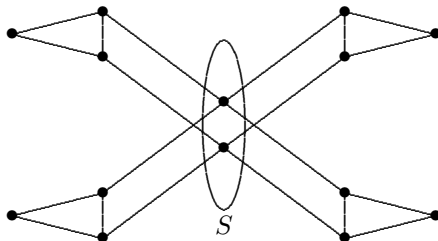
1. Necht' G je libovolný graf s lichým počtem uzlů. Pak zřejmě G nemá perfektní párování.

2.



Nemá perfektní párování

3.



Nemá perfektní párování.

Grafy z příkladů 2 a 3 mají sudý počet uzlů, a přesto se (po chvíli přemýšlení) snadno přesvědčíme o tom, že nemají perfektní párování. Jaký je důvod? Množina uzlů S má následující vlastnost:

$$c_o(G - S) > |S|$$

(zde $c_o(G - S)$ značí počet lichých komponent grafu $G - S$). Taková množina S se nazývá *antifaktorová množina* a její existence v grafu vylučuje existenci perfektního párování. Nutnou podmínkou existence

perfektního párování tedy je neexistence antifaktorové množiny, tj. musí platit $c_o(G - S) \leq |S| \quad \forall S \subset G$. Poznamenejme, že z této podmínky též plyne, že graf G má sudý počet uzlů (dosazením $S = \emptyset$). Tutte ukázal, že tato nutná podmínka je i postačující.

Věta 5.4. (Tutte, 1947) Graf G má perfektní párování, právě když pro každou podmnožinu S množiny uzlů grafu G platí

$$c_o(G - S) \leq |S|$$

(tj. počet lichých komponent grafu $G - S$ je nejvýše roven počtu uzlů množiny S).

Poznámka. Větu lze ekvivalentně formulovat i následujícím způsobem.

Graf G má perfektní párování, právě když v něm neexistuje antifaktorová množina.

Z této formulace je ihned zřejmé, že Tutteova věta je dobrá charakteristika.

Důkaz. 1. Implikace \Rightarrow je zřejmá.

2. Opačnou implikaci \Leftarrow dokážeme sporem. Předpokládejme že graf G je protipříkladem, tj., G je takový, že pro každou množinu $S \subset U(G)$ platí

$$c_o(G - S) \leq |S| \tag{*}$$

(v grafu G neexistuje antifaktorová množina), a G přitom nemá perfektní párování. Jak jsme již poznamenali, z podmínky (*) plyne, že G má sudý počet uzlů.

Všimněme si toho, že splňuje-li G podmínku (*), pak graf, vzniklý z G přidáním libovolné hrany splňuje podmínku (*) také (neboť počet lichých komponent v $G - S$ může přidáním hrany jen klesnout). Přidáním hrany se tedy neporuší neexistence antifaktorové množiny.

Z předchozího pozorování vyplývá, že můžeme předpokládat, že G je maximální protipříklad, tj.

- G neobsahuje antifaktorovou množinu,
- G nemá perfektní párování,
- G přidáním libovolné hrany vznikne z G graf, který již perfektní párování má.

Označme K množinu všech univerzálních uzlů grafu G (uzel x je *univerzální*, jestliže $d_G(x) = n - 1$, tj. x sousedí se všemi ostatními uzly grafu).

Kdyby všechny komponenty grafu $G - K$ byly úplné grafy, tak by platilo:

- v každé sudé komponentě $G - K$ existuje perfektní párování,
- v každé liché komponentě $G - K$ existuje párování, které v ní nechá jen jediný nepokrytý uzel,
- z (*) plyne, že těchto nepokrytých uzlů je méně než $|K|$ a lze je tedy pokrýt hranami do K ,
- díky sudosti $|U(G)|$ je zbylých uzlů v K sudý počet, podle definice množiny K tyto uzly indukují úplný graf a lze je tedy spárovat.

Graf G by tedy měl perfektní párování, což je spor.

Odtud vyplývá, že existuje komponenta G' grafu $G - K$, která není úplným grafem. Ze souvislosti G' plyne, že existují uzly $a, b, c \in U(G')$ takové, že $\{a, b\} \in H(G')$, $\{b, c\} \in H(G')$, ale $\{a, c\} \notin H(G')$. Protože $b \notin K$, existuje uzel d takový, že $\{b, d\} \in H(G)$. Z maximality G plyne, že

- přidáním hrany $\{a, c\}$ získáme graf s perfektním párováním (označíme ho M_1),
- přidáním hrany $\{b, d\}$ získáme graf s perfektním párováním (označíme ho M_2).

Protože M_1 i M_2 jsou perfektní párování, určuje symetrická diference jejich množin hran $H(M_1) \Delta H(M_2)$ graf, v němž má každý uzel $x \notin \{a, b, c, d\}$ stupeň 0 nebo 2. Odtud plyne, že z uzlu d lze budovat alternující cestu P v G hladovým algoritmem, při němž začneme v uzlu d hranou z párování M_1 , dále hranou z M_2 a pokračujeme střídavě, dokud to jde. V každém uzlu $x \notin \{a, b, c, d\}$ lze pokračovat, neboť v symetrické diferenci hran párování M_1 a M_2 má x stupeň 0 nebo 2 a pokud do tohoto uzlu vstoupíme, musíme z něj i vyjít. V uzlu d skončit nemůžeme, neboť párování M_1 již uzel d pokrývá a hrana párování M_2 je hrana $\{b, d\}$ a v grafu G není. Algoritmus tedy skončí v jednom z uzlů a, b, c .

Je-li koncovým uzlem cesty P uzel b , byla poslední hrana cesty P z párování M_1 . Uzly a, c jsou již nutně pokryty párováním M_2 (viz definice perfektního párování M_2). Jestliže nyní podél cesty P mezi uzly d a b prohodíme párování M_1 a M_2 (a jinde je ponecháme tak, jak byla), získáme perfektní párování v G , což je spor.

Zbývá už jen možnost, že skončíme v uzlu a nebo c . Situaci vyšetříme např. pro uzel a (pro c je analogická). Poslední hrana cesty P je nyní z párování M_2 , a uzel c je tímto párováním pokryt (viz definice perfektního párování M_2). Prohodíme-li párování M_1 a M_2 podél cesty C , která vznikla přidáním hrany $\{a, b\}$ k cestě P , získáme z párování M_2 perfektní párování v G , což je opět spor. \square

5.6 Největší párování v neorientovaném grafu, párovací číslo a deficience

V případě, že graf G nemá perfektní párování, je možné se ptát na minimální počet uzlů, které musí zůstat párováním nepokryté. Jinak řečeno, hledáme největší párování, tj. maximalizujeme $|H(M)|$.

Definice 5.8. Počet hran největšího párování v grafu G se nazývá párovací číslo grafu G a značí se $\nu(G)$.

Je zřejmé, že graf má perfektní párování, právě když $\nu(G) = \frac{|U(G)|}{2}$.

Definice 5.9. Číslo $\text{def}(G) = n - 2\nu(G)$ se nazývá deficience grafu G .

Ekvivalentně, deficience $\text{def}(G)$ určuje minimální počet nepokrytých uzlů grafu G .

Věta 5.5. (Berge, 1958)

$$\text{def}(G) = \max\{c_o(G - S) - |S| \mid S \subset U(G)\}.$$

Jinak řečeno, Bergeova věta říká, že počet nutně nepokrytých uzlů grafu G je dán „nejhorší“ antifaktorovou množinou. Poněkud přesněji - nazveme-li číslo $c_o(G - S) - |S|$ defektem antifaktorové množiny S , pak Bergeova věta říká, že deficience grafu G (tj. nejmenší počet nepokrytých uzlů) je rovna maximálnímu defektu antifaktorové množiny v G .

Zdůrazněme, že Bergeova věta je dobrou charakteristikou největšího párování. To plyne z toho, že máme-li rozhodnout, zda dané párování M je největší, pak

- není-li M největší, najdeme větší párování,
- je-li M největší, pak najdeme antifaktorovou množinu S , pro niž je číslo $c_o(G - S) - |S|$ rovno počtu uzlů, nepokrytých párováním M .

Rozhodnutí, zda dané párování M je největší, tedy je v $\text{NP} \cap \text{co-NP}$.

Důkaz. 1. Je-li $S \subset U(G)$ antifaktorová množina, pak evidentně musí zůstat v G alespoň $c_o(G - S) - |S|$ nepokrytých uzlů (po jednom v každé z nadpočetných lichých komponent). Tedy platí

$$\text{def}(G) \geq \max\{c_o(G - S) - |S| \mid S \subset U(G)\}.$$

2. Zbývá ukázat, že této hodnoty se skutečně nabývá, tj. že existuje antifaktorová množina, pro kterou platí rovnost.

Nechť S je antifaktorová množina, která realizuje maximum, tj. je pro ni číslo $k = c_o(G - S) - |S|$ maximální. Chceme ukázat, že $\text{def}(G) \leq k$, tj. že existuje párování, nechávající nejvýše k nepokrytých uzlů. Sestrojíme graf G' následujícím způsobem:

- ke grafu G přidáme množinu K , která obsahuje k nových uzlů,
- z každého uzlu množiny K uděláme přidáním nových hran univerzální uzel v grafu G' .

Ukážeme, že v grafu G' existuje perfektní párování. Kdyby ne, tak v G' existuje antifaktorová množina S' . Protože uzly množiny K jsou univerzální v G , nutně musí být $K \subset S'$ (jinak by totiž zbylé uzly z množiny K způsobily souvislost grafu $G - S'$, což nelze).

Označme $S_1 = S' \setminus K$. Potom platí, že $S_1 \subset U(G)$ a $S' = S_1 \cup K$. Z antifaktorovosti S' v grafu G' plyne, že

$$c_o(G' - S') > |S'| = |S_1| + |K| = |S_1| + c_o(G - S) - |S|,$$

odkud

$$c_o(G' - S') - |S_1| > c_o(G - S) - |S|.$$

Protože však $c_o(G' - S') = c_o(G - S_1)$, dostáváme

$$c_o(G - S_1) - |S_1| > c_o(G - S) - |S|,$$

což je spor s maximalitou S .

Tím je dokázáno, že graf G' má perfektní párování. Odstraněním množiny K a návratem k původnímu grafu G získáme párování, které nechává nepokrytých nejvýše k uzlů. Platí tedy $\text{def}(G) \leq k$. \square

5.7 Edmondsův algoritmus

Základní myšlenka algoritmu na nalezení největšího párování je zdánlivě jednoduchá:

- 1) zvolíme výchozí párování $M = (\emptyset, \emptyset)$,
- 2) hledáme nepokrytý uzel v ,
- 3) hledáme rozšiřující cestu začínající v uzlu v .

Pokud rozšiřující cesta existuje, rozšíříme párování M a zopakujeme krok 2); pokud rozšiřující cesta neexistuje, je párování M největším a algoritmus končí.

Hlavním problémem tohoto postupu je vyhledávání rozšiřujících cest. Uvedeme (dnes již klasický) postup, pocházející od Edmondse (1965). K překonání klíčového kroku bude potřebné jedno pomocné tvrzení. Nejprve si ale zavedeme jedno označení.

Je-li G graf a $F \subset G$ jeho podgraf, pak $G|_F$ označíme graf definovaný následovně:

$$U(G|_F) = U(G) \setminus U(F) \cup \{x\}, \text{ kde } x \notin U(G),$$

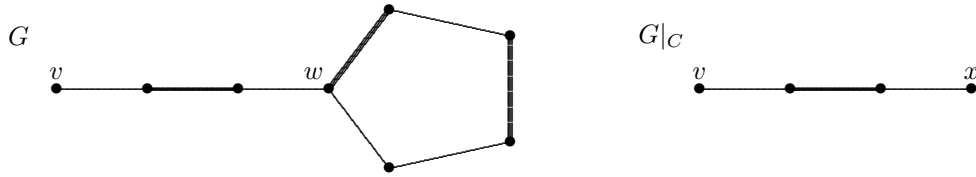
$$H(G|_F) = \{ \{u, v\} \in H(G) \mid u, v \in U(G) \setminus U(F) \} \cup \{ \{v, x\} \mid v \in U(G) \setminus U(F) \text{ a } \{v, y\} \in H(G) \text{ pro některý } y \in U(F) \}.$$

Řekneme, že $G|_F$ vznikl *stažením* F do uzlu x .

Lemma 5.1. *Nechť G je graf, M je párování v G a C je kružnice v G délky $2k + 1$, obsahující k hran párování M a 1 uzel nepokrytý párováním M . Pak M je největší párování v G , právě když párování $M' = M \setminus H(C)$ (tj. párování vzniklé odstraněním z M těch hran, které patří do C) je největší párování v $G|_C$.*

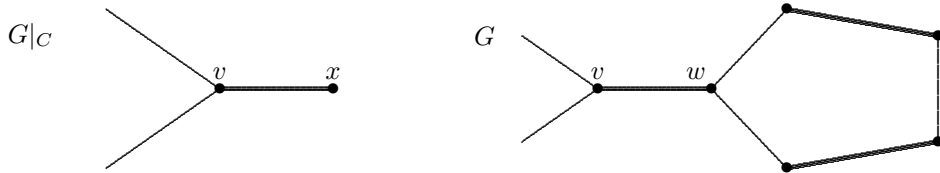
Důkaz. Dokážeme ekvivalenci negací obou výroků.

1. Necht' M není největší v G . Potom existuje M -rozšiřující cesta P v G . Je-li $U(P) \cap U(C) = \emptyset$, je P rozšiřující v $G|_C$ a jsme hotovi. Můžeme tedy předpokládat, že P a C mají společný uzel. Pak nutně jeden koncový uzel cesty P (označme ho v) je mimo C . Označme w první uzel cesty P , který je na kružnici C . Pak cesta vPw odpovídá v grafu $G|_C$ M' -rozšiřující cestě, a tedy M' není největší v $G|_C$.



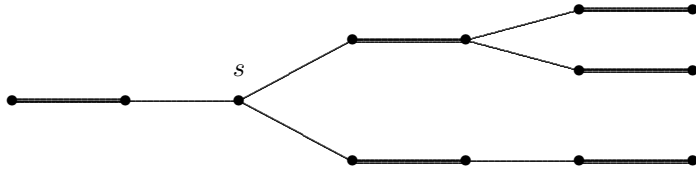
2. Necht' naopak M' není největší párování v grafu $G|_C$, necht' párování N' je větší. Sestrojíme příslušné párování N v grafu G následujícím způsobem:

- všechny hrany N' neobsahující uzel x budou v N ,
- pokud párování N' pokrývalo uzel x hranou vx , má uzel v v grafu G souseda w na kružnici C a do párování N přidáme hranu vw ,
- dodáme k hran kružnice C .



Pak platí, že párování N je v grafu G větší než párování M . □

Vlastní Edmondsův algoritmus je založen na tom, že se z nepokrytých uzlů budují stromy obsahující alternující cesty (viz následující obrázek).



Definice 5.10. Necht' M je párování v grafu G a $X \neq \emptyset$ je množina uzlů nepokrytých párováním M . Les F v grafu G takový, že

- každý uzel z X leží v právě jedné komponentě F ,
- každá komponenta F_i lesa F obsahuje právě jeden uzel $x_i \in X$, nazývaný střed komponenty F_i ,
- každá hrana F v liché vzdálenosti od středu komponenty je v M ,
- koncové uzly F mají sudou vzdálenost od středu komponenty,

se nazývá M -alternující les v G .

Uzly lesa F v sudé (liché) vzdálenosti od středu komponenty se nazývají sudé (liché) uzly (střed komponenty budeme považovat za sudý uzel).

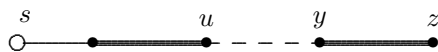
Povšimněme si toho, že je-li F M -alternující les, pak pro každý lichý uzel lesa F je jeho stupeň v F roven dvěma (to plyne z toho, že při větším stupni by podle 3. podmínky vycházely z lichého uzlu alespoň dvě hrany, které jsou v M , ale M by pak nebylo párováním, stupeň 1 není možný podle podmínky 4). Naopak, sudé uzly mohou mít stupeň i větší.

Nechť nyní M je párování a F je M -alternující les. Budeme vyšetřovat sousedy sudých uzlů lesa F . Jsou následující čtyři možnosti:

- 1) některý sudý uzel u má souseda $y \notin U(F)$,
- 2) některé dva sudé uzly u_1, u_2 v různých komponentách F_1, F_2 lesa F jsou v grafu G sousední,
- 3) některé dva sudé uzly u_1, u_2 též komponenty F_i lesa F jsou sousední v grafu G ,
- 4) nenastane ani jedna z předchozích možností, tj. každý sudý uzel má za sousedy jen liché uzly.

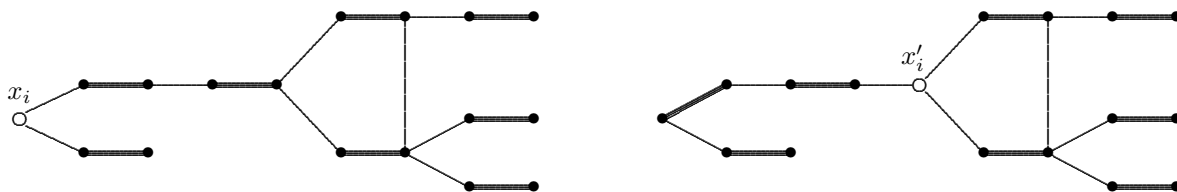
Pro každou z těchto možností si ukážeme, jak postupovat.

1. Jestliže některý sudý uzel u má souseda $y \notin U(F)$, pak platí, že uzel y je M -pokrytý, neboť množina X nepokrytých uzlů je celá v lese F . Jeho soused z v párování M není obsažen v lese F (podle 4. podmínky definice lesa F). Přidáním hran uy a yz do lesa F *zvětšíme* F .



2. Jsou-li některé dva sudé uzly u_1, u_2 v různých komponentách F_1, F_2 lesa F sousední v grafu G , pak středy těchto komponent jsou spojeny rozšiřující cestou. Prohozením hran párování podél této cesty *zvětšíme* M .
3. Předpokládejme, že některé dva sudé uzly u_1, u_2 též komponenty F_i lesa F jsou sousední v grafu G . Pak hrana $\{u_1, u_2\}$ spolu s cestou, spojující u_1 a u_2 v F_i , určují v této komponentě lichou kružnici C .
 - a) Je-li střed x_i komponenty F_i na kružnici C , pak kružnice C splňuje předpoklady lemmatu 5.1.
 - b) Není-li střed x_i na kružnici C , pak prohozením párování M podél cesty P , která spojuje kružnici C a střed x_i , dostaneme párování M' s vlastností $|M'| = |M|$, a takové, že nový střed komponenty F_i již leží na kružnici C , čímž jsme převedli tuto variantu na předchozí.

(Převod případu ad b) na ad a) je ukázán na následujícím obrázku, kde vlevo je původní komponenta a vpravo je komponenta s prohozeným párováním a novým středem.)



Nyní podle lemmatu 5.1 lze stáhnout kružnici C do jednoho uzlu, čímž *redukujeme velikost grafu* G .

4. Předpokládejme, že neplatí ani jedna z předchozích možností, tj. každý sudý uzel má za sousedy jen liché uzly. Pak tvrdíme, že

M je největší párování.

To dokážeme takto. Označme S množinu všech sudých uzlů (včetně středů komponent), L množinu všech lichých uzlů, $s = |S|$ a $\ell = |L|$. Pak $s - \ell = |X|$ (kde X je množina všech středů komponent).

Je zřejmé, že graf $G - L$ nemá žádnou hranu. Tento graf má tedy s lichých (jednouzlových) komponent, tedy $c_o(G - L) = s$. To znamená, že $c_o(G - L) - |L| = |X|$. Množina L je tedy antifaktorovou množinou, a pro deficienci $\text{def}(G)$ grafu G tedy platí $\text{def}(G) \geq s - \ell = |X|$. Naše párování M ale nechává nepokrytých právě $|X|$ uzlů, proto $\text{def}(G) \leq |X|$. Platí tedy rovnost, a tedy podle Věty 5.5 je párování M největší (nalezli jsme antifaktorovou množinu L a párování M tak, že defekt množiny L je roven počtu uzlů nepokrytých párováním M).

Ukázali jsme, že v každém kroku algoritmu je možno udělat jednu z následujících věcí:

- 1) zvětšit alternující les F ,
- 2) zvětšit párování M ,

- 3) redukovat velikost grafu G ,
- 4) skončit s největším párováním M .

Je zřejmé, že takový postup dává *polynomiální algoritmus*. Praktické implementace tohoto algoritmu pracují v čase $O(n^4)$. Lepší známé postupy pracují v čase $O(nm)$, nejrychlejší (a rovněž nejsložitější) dokonce řádově v čase $O(n^{\frac{5}{2}})$. Podrobnosti lze nalézt např. v knížce [8], hlubší teoretické souvislosti například v monografii [9].

6 Hranové grafy

V této a následující kapitole poznáme příklady dvou speciálních tříd grafů, které jsou zajímavé tím, že pro grafy z těchto tříd se zjednodušuje výpočetní složitost některých problémů.

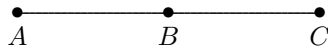
6.1 Základní pojmy

Začneme nejprve motivačním příkladem.

Příklad. Je dán systém množin $\mathcal{A} = \{A, B, C\}$, kde

$$\begin{aligned} A &= \{1, 5, 6, 9\}, \\ B &= \{1, 4, 7\}, \\ C &= \{2, 3, 7, 8\}. \end{aligned}$$

Průnikový graf systému \mathcal{A} je graf, jehož uzly tvoří jednotlivé množiny A, B, C , a hranami spojíme takové dvojice uzlů, jejichž příslušné množiny mají neprázdný průnik. V našem případě tedy $A \cap B \neq \emptyset$, $B \cap C \neq \emptyset$, ale $A \cap C = \emptyset$. Průnikový graf tohoto systému množin tedy vypadá následovně:



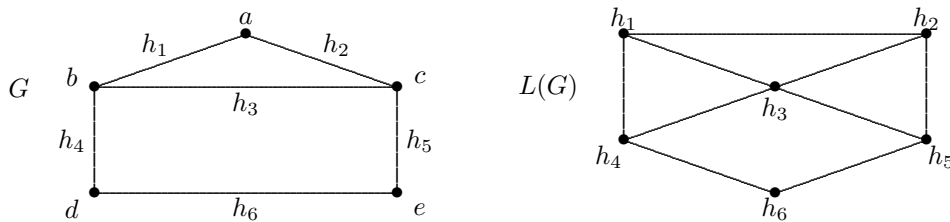
Definice 6.1. Necht' G je neorientovaný graf. Graf $L(G)$, definovaný předpisem

$$\begin{aligned} U(L(G)) &= H(G), \\ H(L(G)) &= \{ \{h_i, h_j\} \mid \text{hrany } h_i, h_j \text{ mají společný uzel} \}, \end{aligned}$$

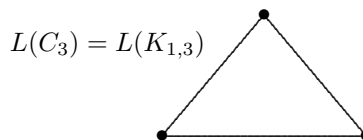
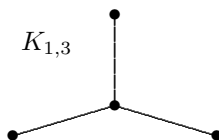
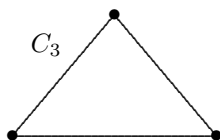
se nazývá *hranový graf* (anglicky „line graph“) grafu G .

Poznámka. Množina hran neorientovaného grafu G je množinou dvouprvkových podmnožin množiny $U(G)$, a tedy lze na ni pohlížet jako na množinový systém. Hranový graf $L(G)$ je pak průnikový graf tohoto systému množin.

Příklad. Na následujícím obrázku je vlevo graf G a vpravo jeho hranový graf $L(G)$.



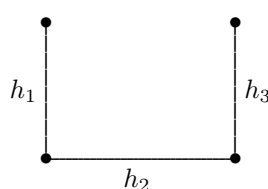
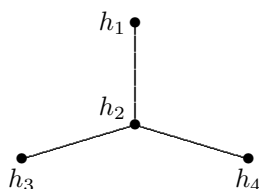
Příklad. Necht' G_1 je kružnice délky 3 a G_2 je úplný bipartitní graf $K_{1,3}$. Pak G_1 a G_2 jsou neizomorfní, ale jejich hranové grafy jsou izomorfní.



Lze však dokázat, že grafy C_3 a $K_{1,3}$ jsou jediné dva neizomorfní grafy, které mají stejný hranový graf. Kromě tohoto vyjímecného případu platí, že každé dva neizomorfní souvislé grafy mají neizomorfní hranové grafy.

Nyní se pokusíme vyšetřit opačnou otázku, tj. k danému hranovému grafu G hledat jeho „vzor“ H , tj. graf H takový, že $L(H) = G$. První otázkou je, zda vůbec pro každý G takový H existuje. Následující příklad nám dá odpověď.

Příklad. Mějme dán graf $K_{1,3}$ a pokusíme se zjistit, zda je hranovým grafem nějakého grafu. Označíme jeho uzly postupně h_1, \dots, h_4 . V původním grafu tyto uzly samozřejmě představují hrany (viz definice hranového grafu). Budeme postupně vytvářet graf, pro nějž je graf $K_{1,3}$ hranový.



Zde se dostáváme do problémů. Máme přidat hranu h_4 , která má společný uzel jen s hranou h_2 a žádnou jinou. To ale nejde, proto pro graf $K_{1,3}$ nelze vytvořit vzor, graf $K_{1,3}$ tedy není hranový.

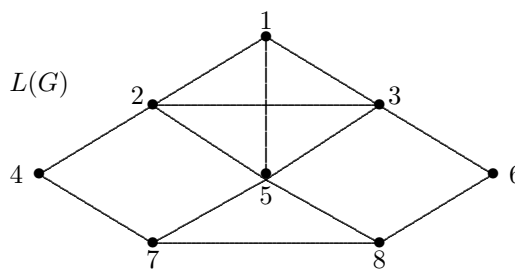
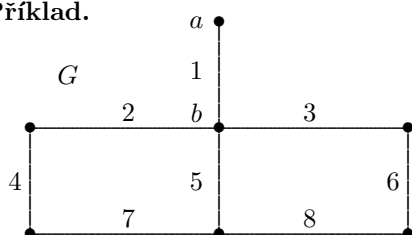
Definice 6.2. Řekneme, že G je hranový graf, jestliže existuje graf H takový, že platí $L(H) = G$.

Poznámka. Z předchozího příkladu vyplývá, že pokud graf G obsahuje indukovaný podgraf izomorfní s $K_{1,3}$, pak G není hranovým grafem.

6.2 Charakterizační věty

Následující příklad ukazuje klíčovou myšlenku charakterizace hranových grafů.

Příklad.



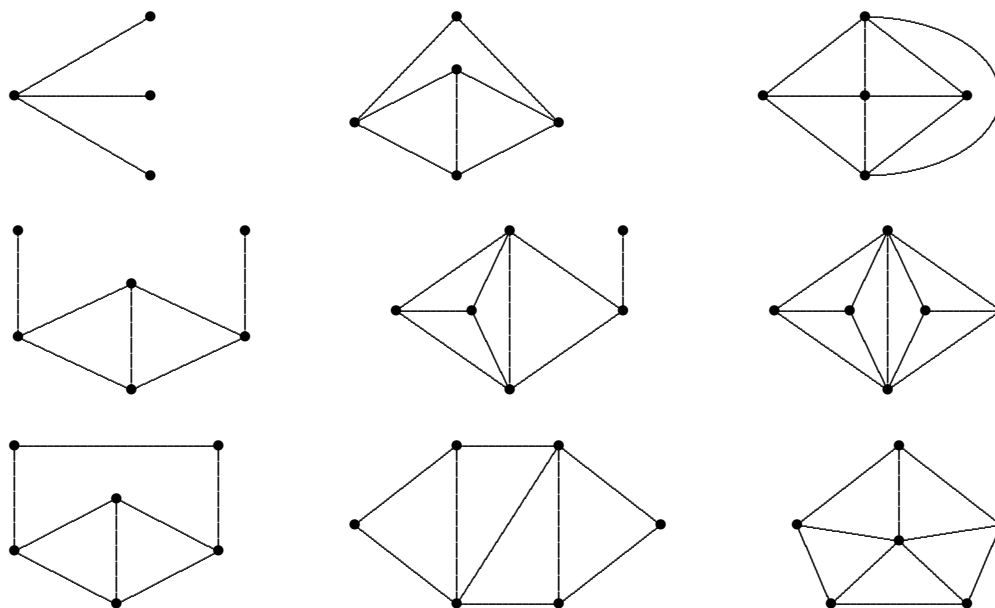
Po krátkém zamyšlení nad oběma grafy zjistíme, že uzlu stupně i v G odpovídá klika velikosti i v hranovém grafu $L(G)$. Konkrétně, jednozlová klika s uzlem 1 v $L(G)$ odpovídá uzlu a grafu G , klika indukovaná na uzlech 1, 2, 3, 5 v grafu $L(G)$ odpovídá uzlu b v grafu G atd. Tento příklad ukazuje myšlenku důkazu následující věty.

Věta 6.1. (Krausz) Graf G je hranovým grafem, právě když v grafu G existuje takový systém klik \mathcal{K} , že každý uzel leží právě ve dvou klikách a každá hrana leží v právě jedné klice systému \mathcal{K} .

Poznámka. Důkaz této věty nebudeme provádět. Poznamenejme jen, že podmínka, že každý uzel grafu $G = L(H)$ leží v právě dvou klikách znamená, že v grafu H má každá hrana dva konce, a podmínka, že každá hrana grafu $G = L(H)$ náleží právě jedné klice znamená, že dvě hrany grafu H , které mají společný uzel, mohou mít společný jen jeden uzel.

Věta 6.1 evidentně není dobrou charakteristikou. Tuto příjemnou vlastnost má až následující věta (kterou uvádíme také bez důkazu).

Věta 6.2. (Beineke, 1969) Graf G je hranovým grafem, právě když neobsahuje jako indukovaný podgraf žádný z následujících devíti grafů.



Důsledek 6.1.

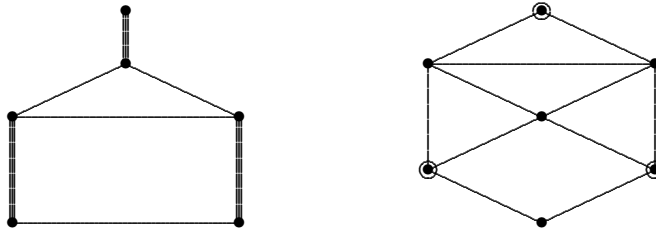
- (i) Vlastnost „býti hranovým grafem“ je dědičná na indukované podgrafy.
- (ii) Otázku „je daný graf hranovým grafem?“ lze řešit v polynomiálním čase.

Důkaz.

- (i) To je zřejmé, neboť je-li graf G_1 indukovaným podgrafem grafu G , pak i každý indukovaný podgraf grafu G_1 je indukovaným podgrafem grafu G .
- (ii) Grafy z Beinekeho věty 6.2 mají nejvýše šest uzlů, a tedy je lze určitě poznat v čase $O(n^6)$. □

6.3 Hranové grafy a výpočetní složitost

Příklad. Uvažujme graf G a jeho hranový graf $L(G)$ na následujícím obrázku.



Vidíme, že silně vyznačeným hranám párování v grafu G odpovídají zakroužkované uzly v grafu $L(G)$, které zřejmě tvoří nezávislou množinu.

Obdobným pozorováním snadno zjistíme i korespondence mezi dalšími grafovými vlastnostmi grafu H a jeho hranového grafu $G = L(H)$. Tato pozorování přehledně shrneme do následující tabulky.

H	$G = L(H)$
hrana	uzel
uzel	klika
stupeň uzlu	počet uzlů kliky
maximální stupeň $\Delta(H)$	klikovost $\omega(G)$ ⁴
párování	nezávislá množina
párovací číslo $\nu(H)$	nezávislost $\alpha(G)$
hranové obarvení	uzlové obarvení
chromatický index $\chi'(H)$	chromatické číslo $\chi(G)$

Následující tvrzení, vyplývající ihned z uvedených korespondencí, ukazuje důležitost zkoumání speciálních tříd grafů v souvislosti s algoritmy a výpočetní složitostí.

Tvrzení 6.1. *Je-li G hranovým grafem, pak je možno určit jeho nezávislost $\alpha(G)$ a klikovost $\omega(G)$ v polynomiálním čase.*

Důkaz. Úlohu nalezení největší nezávislé množiny v $L(G)$ (která je v obecných grafech NP-těžká) převedeme na úlohu nalezení největšího párování v grafu G . Tato úloha je řešitelná v polynomiálním čase a převod $L(G) \rightarrow G$, resp. $G \rightarrow L(G)$ je rovněž polynomiální.

Úlohu nalezení klikovosti v $L(G)$ (která je rovněž v obecnosti NP-těžká) převedeme na úlohu hledání maximálního stupně uzlu v grafu G , která je zřejmě polynomiální. □

7 Rovinné grafy

V této kapitole se budeme zabývat pouze neorientovanými grafy. Úvodem je třeba říci, že některé pojmy budou definovány poněkud intuitivně (k přesné formulaci by bylo zapotřebí poměrně hlubokých poznatků z topologie, přesahující rámec předmětu). Pro naše účely bude intuitivní definice plně postačující.

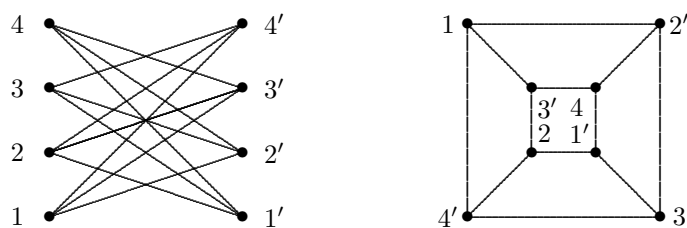
7.1 Uložení grafu na plochu

Definice 7.1. *Nechť S je plocha v E_3 . Řekneme, že graf G lze uložit na plochu S , jestliže jej lze znázornit pomocí bodů (uzlů) a oblouků (hran) na S tak, že žádné dva oblouky nemají společný vnitřní bod.*

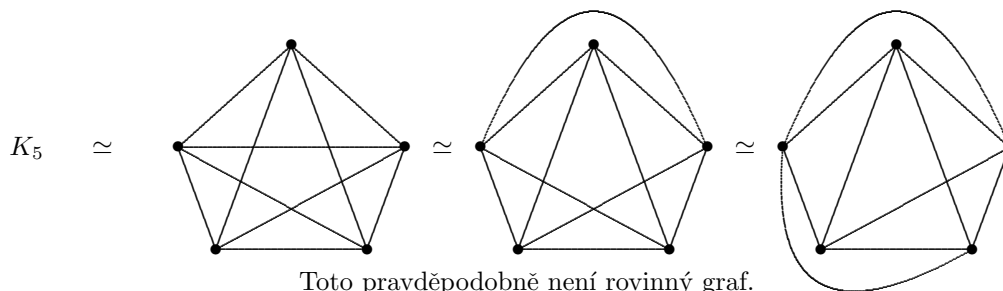
⁴Korespondence $\Delta(H) = \omega(L(H))$ platí až na jednu výjimku, kterou už známe: je-li $H = C_3$, pak $L(H) = C_3$ a je $\Delta(H) = 2$, ale $\omega(L(H)) = 3$.

Definice 7.2. Řekneme, že G je rovinný graf, jestliže jej lze uložit do roviny.

Příklad.



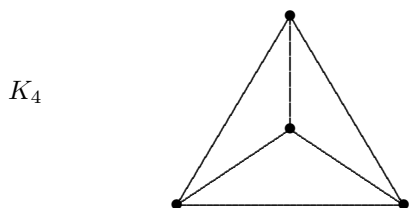
Tento graf je zřejmě rovinný.



Toto pravděpodobně není rovinný graf.

Definice 7.3. Je-li graf G uložen v rovině, pak oblasti (části) roviny, které vymezuje, se nazývají stěny grafu G .

Příklad.



3 omezené stěny a jedna (vnější) stěna.

Poznámka. Každý strom má zřejmě jedinou stěnu, která není ohraničena žádnou kružnicí a je tedy vnější.

7.2 Eulerova věta a její důsledky

Věta 7.1. (Euler) Necht' souvislý rovinný graf G má $n \geq 1$ uzlů, m hran a s stěn. Pak platí

$$n - m + s = 2.$$

Poznámka. Těto rovnosti se velmi často říká *Eulerův vzorec*.

Důkaz. Důkaz provedeme indukcí podle počtu hran m .

1) Pro $m = 1$ je zřejmě $n = 2$ a $s = 1$, věta tedy platí.

2) Necht' graf G má m hran, n uzlů a s stěn a necht' věta platí pro každý souvislý rovinný graf s $m - 1$ hranami.

a) Je-li $s = 1$, tj. graf G je strom, vytvoříme graf G' odstraněním koncové hrany i s příslušným uzlem stupně 1. Potom graf G' má $m - 1$ hran, $n - 1$ uzlů a opět jednu stěnu. Platí pro něj tedy

$$(n - 1) - (m - 1) + s = 2 \Rightarrow n - m + s = 2.$$

b) Je-li $s \geq 2$, vytvoříme graf G' odstraněním hrany v G , která sousedí se dvěma různými stěnami. Graf G' tedy má n uzlů, $m - 1$ hran a $s - 1$ stěn a platí pro něj

$$n - (m - 1) + (s - 1) = 2 \Rightarrow n - m + s = 2.$$

□

Důsledek 7.1. Pro každý rovinný graf G s alespoň 3 uzly platí

$$|H(G)| \leq 3|U(G)| - 6.$$

Důkaz. Nejvíce hran má graf, v němž každá stěna je trojúhelník (tzv. triangulace). Potom platí $3s = 2m$, tedy $s = \frac{2}{3}m$. Po dosazení do Eulerova vzorce obdržíme rovnost

$$n - m + \frac{2}{3}m = 2 \Rightarrow m = 3n - 6.$$

Pro každý graf, který není maximální (tj. není triangulací) bude platit nerovnost. □

Důsledek 7.1 ukazuje, že rovinné grafy nemohou být příliš husté: počet hran rovinného grafu je nejvýše lineární funkcí počtu uzlů (zatímco v obecnosti může počet hran růst s počtem uzlů kvadraticky). To má význam pro složitost některých algoritmů. Například složitost prohledávání do hloubky (backtrackingu) je $O(m + n)$, což v obecnosti může být $O(n^2)$, ale v rovinných grafech díky důsledku 7.1 jen $O(n)$.

V grafech bez trojúhelníků lze tento horní odhad dále vylepšit.

Důsledek 7.2. Pro každý rovinný graf bez trojúhelníků s alespoň 4 uzly platí

$$|H(G)| \leq 2|U(G)| - 4.$$

Důkaz. Maximální rovinný graf bez trojúhelníků má stěny buď C_4 nebo C_5 (ve stěně C_6 lze totiž přidat hranu bez vzniku trojúhelníku a získali bychom spor s maximalitou). Počet stěn C_4 označíme s_4 a počet stěn C_5 označíme s_5 . Potom platí

$$2m = 4s_4 + 5s_5 \Rightarrow s_4 + \frac{5}{4}s_5 = \frac{m}{2}.$$

Celkový počet stěn grafu je $s = s_4 + s_5$, a tedy

$$s = s_4 + s_5 \leq s_4 + \frac{5}{4}s_5 = \frac{m}{2}.$$

Z Eulerova vzorce dále plyne

$$m = n + s - 2 \leq n + \frac{m}{2} - 2 \Rightarrow m \leq 2n - 4.$$

□

Důsledek 7.3. Grafy K_5 a $K_{3,3}$ nejsou rovinné.

Důkaz.

1. Pro K_5 je $n = 5$, $m = 10$. Kdyby to byl rovinný graf, muselo by podle důsledku 7.1 být $m \leq 3n - 6$, a tedy $10 \leq 3 \cdot 5 - 6 = 9$.
2. $K_{3,3}$ nemá trojúhelníky a je v něm 6 uzlů a 9 hran, tedy $n = 6$, $m = 9$. Kdyby byl rovinný, pak by podle důsledku 7.2 platilo $m \leq 2n - 4$, a tedy $9 \leq 2 \cdot 6 - 4 = 8$.

□

Tento důsledek ukazuje, že pokud graf obsahuje jako podgraf jeden z grafů K_5 nebo $K_{3,3}$, nutně musí být nerovinný. Zanedlouho uvidíme, že tato nutná podmínka je (v jistém smyslu) i postačující. Nejprve ale uvedeme ještě jeden důsledek Eulerovy věty.

Důsledek 7.4. Každý souvislý rovinný graf s alespoň čtyřmi uzly má alespoň tři uzly stupně nejvýše 5.

Důkaz. Kdyby měl nejvýše dva takové uzly, platilo by

$$2|H(G)| = \sum d_G(u) \geq 6(|U(G)| - 2) + 1 \cdot 2 = 6|U(G)| - 10.$$

(předpokládáme, že ony dva uzly jsou stupně 1 a ostatní stupně 6). Odtud však plyne

$$|H(G)| \geq 3|U(G)| - 5,$$

podle Eulerovy věty tedy takový graf není rovinný.

□

Poznámka. Lze dokonce dokázat (s využitím dalších grafových vlastností), že uzly stupně ≤ 5 musí být alespoň čtyři.

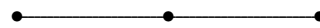
7.3 Kuratowského věta a další věty o rovinných grafech

Poznámka. Přidáme-li na některou hranu uzel 2. stupně, určitě to nebude mít vliv na rovinnost. Takovou operaci nazveme *půlení hrany*.

Příklad. Operace půlení hrany:



původní graf



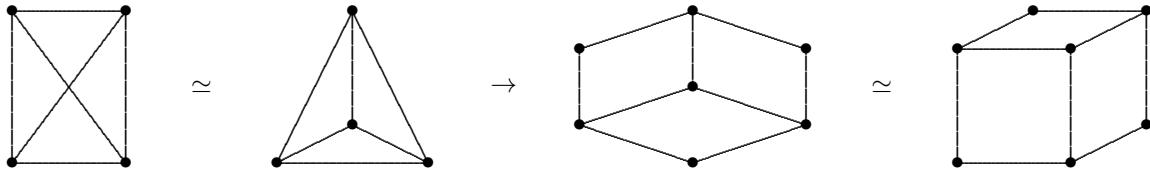
graf po operaci půlení hrany

Definice 7.4. Řekneme, že dva grafy G_1, G_2 jsou homeomorfní, pokud je možno konečným počtem půlení hran dosáhnout toho, že vzniklé grafy jsou izomorfní.

(Konečným počtem samozřejmě rozumíme i nulu, tedy každé dva izomorfní grafy jsou homeomorfní.)

Příklad.

1. Každé dvě kružnice nebo cesty jsou homeomorfní.
2. Příklad vzájemně homeomorfních grafů:



Věta 7.2. (Kuratowski) *Graf je rovinný, právě když neobsahuje podgraf homeomorfní s K_5 nebo s $K_{3,3}$.*

Důkaz. Podle 3. důsledku Eulerovy věty je tato podmínka nutná; že je postačující nebudeme dokazovat. \square

Poznámka. Poznamenejme, že Kuratowského věta je dobrou charakteristikou.

Věta 7.2 je elegantní, ale ne příliš praktická. Jsou známy algoritmy, které umí nalézt rovinné uložení grafu dokonce v čase $O(n)$, jsou však dosti složité. Jejich základní princip spočívá v tom, že naleznou kružnici, zakreslí ji do roviny a pak k ní přidávají další části. Jejich hledání, nebo důkaz neexistence uložení, jsou založeny na backtrackingu.

V důsledku Kuratowského věty pro klikovost $\omega(G)$ rovinného grafu G platí $\omega(G) \leq 4$. Určení $\omega(G)$ je tedy ve třídě rovinných grafů polynomiální (například hrubou silou - probírkou všech čtveřic uzlů, se složitostí $O(n^4)$).

Věta 7.3. *Graf je rovinný, právě když jej lze uložit na kulovou plochu (sféru).*

Důkaz. Opírá se o stereografickou projekci. \square

Mnohostěnový graf je graf vzniklý uložním mnohostěnu do roviny.

Důsledek 7.5. *Každý mnohostěnový graf je rovinný.*

Důsledek 7.6. *Pro každou hranu h 2-souvislého grafu G existuje takové uložení do roviny, při němž hrana h sousedí s vnější stěnou.*

Důkaz. Zobrazíme graf G na sféru, pootočíme tak, aby stěna s hranou h byla nahoře, a zobrazíme jej zpět do roviny. \square

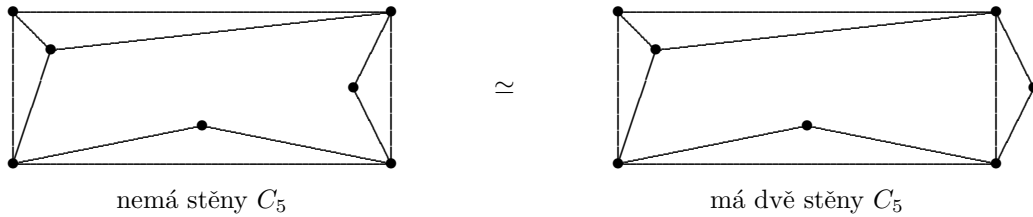
Vraťme se ještě k maximálním rovinným grafům. Již víme, že každý maximální rovinný graf má jen trojúhelníkové stěny. Následující věta ukazuje, že platí ještě více.

Věta 7.4. *Každý maximální rovinný graf s alespoň čtyřmi uzly je 3-souvislý.*

Důkaz. Kdyby G měl dvouprvkový uzlový řez x, y , tak by, po případném přesunutí hrany $\{xy\}$ do vnější stěny, v omezené stěně obsahující oba uzly x, y musely být další uzly a, b (jinak by uzly x, y byly spojeny dvojicí paralelních hran). Tyto dva uzly a, b lze spojit při zachování rovinnosti, což je spor s maximalitou. \square

3-souvislost je však důležitá i jinak.

Příklad. Pokud není graf 3-souvislý, ale např. jen 2-souvislý, může se stát následující.



Přitom se však jedná o dvě různá uložení téhož grafu. Pojem stěny proto nelze vztahovat ke grafu, nýbrž jen k jeho konkrétnímu uložení. Následující věta ukazuje, že pro 3-souvislé grafy tato potíž odpadá.

Věta 7.5. (Whitney) Každý 3-souvislý graf má jednoznačné uložení na sféru.

Důkaz. Že je podmínka 3-souvislosti nutná, plyne z předchozího příkladu, že je postačující, nebudeme dokazovat. □

Následující věta, charakterizující mnohostěnové grafy, ukazuje další situaci, kdy je 3-souvislost důležitá. Každý mnohostěnový graf musí být uložitelný na sféru, tj. musí být rovinný. Názorná je i nutnost 3-souvislosti pro „tuhost prostorové konstrukce“. Není lehké dokázat, že jsou tyto podmínky i postačující.

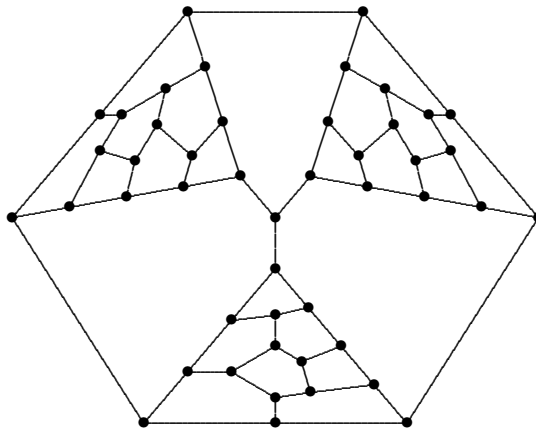
Věta 7.6. (Steinitz, Rademacher) Graf je grafem mnohostěnu, právě když je rovinný a 3-souvislý.

Uveďme ještě některé další zajímavé výsledky o rovinných grafech vyšších stupňů souvislosti.

Věta 7.7. (Tutte, 1956) Každý 4-souvislý rovinný graf je hamiltonovský.

Následující příklad ukazuje, že pro 3-souvislé grafy již taková věta neplatí.

Příklad. Ukažte, že tento graf není hamiltonovský.



Závěrem tohoto odstavce si uveďme příklad jednoho klasického otevřeného problému.

Hypotéza 7.1. (Barnette) Každý 3-souvislý bipartitní rovinný pravidelný graf 3. stupně je hamiltonovský.

Je známo, že hypotéza platí pro grafy s nejvýše 26 uzly. Pro grafy s více uzly je otázka otevřená.

8 Problém obchodního cestujícího

Problém obchodního cestujícího (anglicky "Traveling Salesman Problem, TSP) je vlastně úlohou nalézt v ohodnoceném neorientovaném grafu hamiltonovskou kružnici s nejmenším součtem ohodnocení hran. V této kapitole si ukážeme dvě různé heuristiky pro řešení této úlohy. Tyto heuristiky uvádíme jako příklady dvou různých typů heuristik.

První z nich, metoda penalizací, je typickým příkladem algoritmu, který nedá odpověď vždy, ale pokud umí řešení najít, pak je najde přesně (tj. buď to dá přesnou odpověď, nebo odpoví „nevím“).

Naproti tomu algoritmus z odstavce 8.2 „funguje“ vždy, ale odpověď, kterou dá, je pouze aproximací přesného řešení. Pokud hranové ohodnocení grafu splňuje trojúhelníkovou nerovnost, je možno odvodit horní odhad chyby (nejvýše dvojnásobek optima).

8.1 Metoda penalizací

Východiskem našich úvah bude následující zřejmé tvrzení.

Lemma 8.1. *Podgraf H grafu G s n uzly je hamiltonovskou kružnicí v G , právě když platí, že H má n uzlů, n hran, je souvislý a je pravidelným grafem stupně 2.*

Pokud z předchozího lemmatu vynecháme požadavek pravidelnosti stupně 2, obdržíme následující pojem.

Definice 8.1. *Nechť G je souvislý graf. Souvislý faktor grafu G , který má stejný počet hran i uzlů, se nazývá 1-strom grafu G .*

Je-li graf G hranově ohodnocený, pak minimální 1-strom je takový 1-strom, pro nějž je součet ohodnocení hran minimální.

Víme, že souvislý graf na n uzlech je stromem, právě když má $n - 1$ hran. 1-strom tedy vznikne přidáním jedné hrany ke stromu. Následující tvrzení ukazuje, že minimální 1-strom pak lze vytvořit z minimální kostry grafu přidáním té hrany, která nepatří do kostry a má ze všech takových hran nejmenší ohodnocení.

Tvrzení 8.1. *Nechť G je ohodnocený graf, K je jeho minimální kostra a $h \in H(G) \setminus H(K)$ je taková hrana, pro kterou $w(h) = \min\{w(h') \mid h' \in H(G) \setminus H(K)\}$. Pak graf $T = K + h$ je minimální 1-strom grafu G .*

Důkaz. Tvrzení dokážeme sporem. Nechť tedy 1-strom T není minimální, nechť 1-strom T' má menší součet ohodnocení. 1-strom T' má právě jednu kružnici C' (což plyne přímo z definice 1-stromu). Tato kružnice C' obsahuje alespoň jednu hranu, která není v kostře K . Tuto hranu označíme h' . Označme K' kostru $K' = T' - h'$. Podle předpokladu je $w(T') < w(T)$. Podle konstrukce 1-stromu T platí, že $w(h) \leq w(h')$. Odtud ale plyne, že

$$w(K') = w(T' - h') < w(T - h) = w(K),$$

a tedy kostra K nebyla minimální, což je spor. □

Poznamenejme, že Tvrzení 8.1 umožňuje najít minimální 1-strom v polynomiálním čase. Hamiltonovská kružnice je zřejmě speciálním případem 1-stromu, a proto je součet ohodnocení minimálního 1-stromu dolním odhadem velikosti řešení příslušné úlohy obchodního cestujícího.

Následující lemma dává základní ideu heuristiky pro řešení problému obchodního cestujícího, které budeme říkat *metoda penalizací*.

Lemma 8.2. *Nechť G je ohodnocený graf s hranovým ohodnocením c_{ij} a nechť t_1, \dots, t_n je ohodnocení uzlů grafu G . Změníme-li ohodnocení grafu G tak, že ohodnocení hrany $\{v_i, v_j\}$ se změní z čísla c_{ij} na $c_{ij} + t_i + t_j$, pak se nezmění řešení problému obchodního cestujícího, ale mohou se změnit minimální 1-stromy.*

Důkaz. Každý uzel je incidentní se dvěma hranami hamiltonovské kružnice. Změna ohodnocení hran proto vede ke změně ohodnocení všech kružnic o tutéž hodnotu, která je rovna dvojnásobku součtu čísel t_i . Pro 1-stromy ale nic podobného obecně neplatí. \square

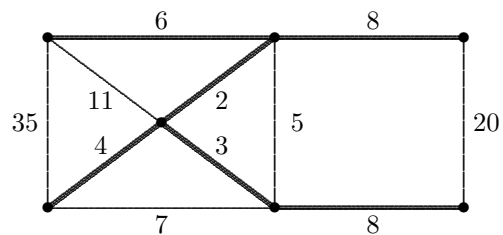
Čísla t_i budeme nazývat *penalizace uzlů*. Touto penalizací se nezmění řešení problému obchodního cestujícího, ale je možné se pokoušet vhodným penalizováním měnit tvar určeného minimálního 1-stromu tak, aby se stal kružnicí.

Za míru odlišnosti 1-stromu T od kružnice budeme považovat číslo $D(T) = \sum_{U(T)} |\deg_T(v_i) - 2|$, neboť podle lemmatu 8.1 je 1-strom T kružnicí, právě když $D(T) = 0$.

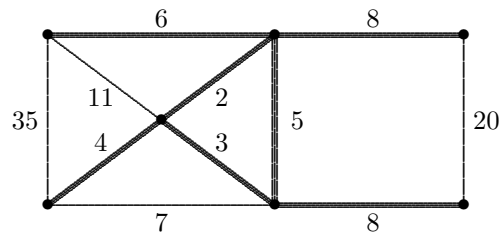
Jestliže pro uzel v_i je $\deg_T(v_i) > 2$, a zvolíme-li penalizaci uzlu v_i kladnou, pak se ohodnocení hran vycházejících z uzlu v_i zvětší, takže je možné očekávat, že některé z nich v důsledku penalizace z minimálního 1-stromu vypadnou, což způsobí pokles čísla $\deg_T(v_i) - 2$. Je-li $\deg_T(v_j) = 1$ (jiný menší než 2 být nemůže), volíme naopak penalizaci uzlu v_j zápornou, abychom stupeň uzlu zvětšili. I když penalizace může ovlivnit stupně uzlů v 1-stromu zcela jinak, než bylo zamýšleno, jsou praktické zkušenosti s touto metodou dobré.

Příklad. Ukážeme si, jak metoda penalizací postupně mění minimální 1-strom. Zde byla použita penalizace $t_i = 10(\deg_T v_i - 2)$.

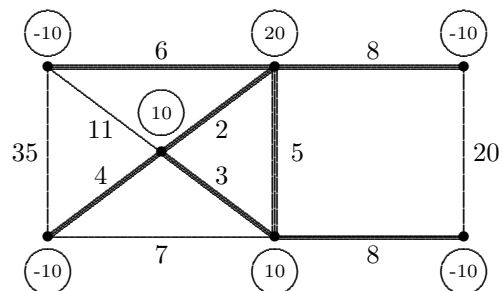
Mějme dán následující graf s hranovým ohodnocením, ve kterém jsme určili minimální kostru (je vyznačena silně).



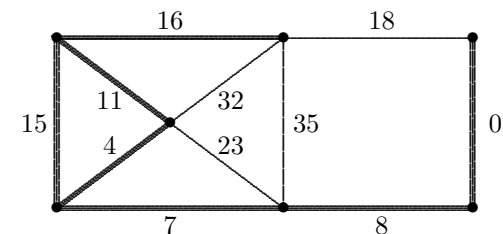
Tuto minimální kostru doplníme na minimální 1-strom.



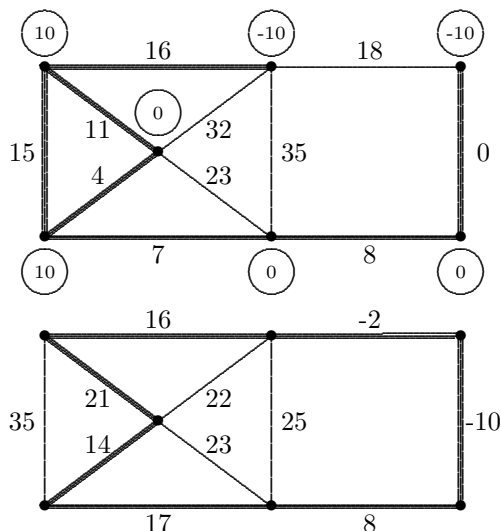
Nalezený minimální 1-strom není kružnicí, budeme tedy penalizovat výše uvedeným způsobem.



Určíme nové ohodnocení hran a nalezneme nový minimální 1-strom.



Minimální 1-strom opět není kružnicí, je tedy nutné znovu penalizovat.



Opět určíme nové ohodnocení hran a nalezneme nový minimální 1-strom.

Tento minimální 1-strom je kružnicí, našli jsme tedy řešení problému obchodního cestujícího.

Připomeňme, že metoda penalizací je příkladem optimalizační heuristiky, která buďto nalezne přesné řešení, nebo řekne „nevím“.

Algoritmus (schéma metody penalizací)

Vstupní data - graf G s hranami ohodnocenými čísly c_{ij} . Úkolem je nalézt v grafu G minimální hamiltonovskou kružnici. V následujícím schématu budeme potřebovat pomocné proměnné t_i, z_i .

1. *Inicializace.* Pro $i := 1, \dots, n$ položme $t_i = 0$.
2. *Určení minimálního 1-stromu.* V grafu G , ve kterém se změnil ohodnocení hran z hodnot c_{ij} na hodnoty $c_{ij} + t_i + t_j$, určíme minimální 1-strom T .
3. *Test ukončení.* Je-li T kružnice, výpočet končí, T je hledaná kružnice (KONEC).
4. *Změna penalizace.* Pokud T není kružnice, pro každé $i := 1, \dots, n$ zvolíme číslo z_i a položíme $t_i = t_i + z_i$. Jdeme do bodu 2.

Může se stát, že takto popsaný postup opakuje cyklus bodů 2, 3, 4 neustále, aniž by našel řešení, ať je tomu tak kvůli zadání úlohy, nebo nevhodnými změnami penalizace. Proto je třeba do bodu 3 přidat mechanismus, který v takovém případě výpočet zastaví.

8.2 Kružnice malé váhy v úplném ohodnoceném grafu

Uvažujme univerzální obráběcí stroj, který umí dělat n různých operací (vrtání, řezání, broušení atd.). K vyrobení výrobku je třeba provést všechny operace a na jejich pořadí nezáleží. Přestavení stroje z i -té na j -tou operaci trvá w_{ij} časových jednotek. Úkolem je nalézt takové pořadí operací, při němž se minimalizuje celková doba prostojů, způsobených přestavováním stroje. Tato úloha se nazývá „job sequencing problem“ (česky snad „problém řazení operací“) a lze ji zřejmým způsobem převést na úlohu obchodního cestujícího v ohodnoceném úplném grafu, v němž uzly jsou operace a ohodnocení hran je dáno čísly w_{ij} . Je rozumné předpokládat, že hranové ohodnocení splňuje trojúhelníkovou nerovnost

$$w_{ij} + w_{jk} \geq w_{ik}, \quad \forall i, j, k \in U(G).$$

Algoritmus, který si ukážeme, sice dá (nějaké) řešení vždy, ale hamiltonovská kružnice, kterou nalezneme, nemusí být nutně optimální.

Algoritmus (hamiltonovská kružnice malé váhy v úplném ohodnoceném grafu s trojúhelníkovou nerovností)

Vstup: Úplný graf G s ohodnocením $w : H(G) \rightarrow (0, \infty)$, splňujícím trojúhelníkovou nerovnost.

1. Najdeme minimální kostru T grafu G .
2. Zvolíme libovolný uzel $v \in V(G)$ a očíslováme uzly grafu G v pořadí prohlédávání stromu T do hloubky od uzlu v . Nechť H je výsledná posloupnost.
3. Položíme $C = H, v$.

Výstup: Kružnice C .

Příklad. Mějme ohodnocený úplný graf G , popsany maticí $W(G)$:

$$W(G) = \begin{pmatrix} 0 & 3 & 3 & 2 & 7 & 3 \\ 3 & 0 & 3 & 4 & 5 & 5 \\ 3 & 3 & 0 & 1 & 4 & 4 \\ 2 & 4 & 1 & 0 & 5 & 5 \\ 7 & 5 & 4 & 5 & 0 & 4 \\ 3 & 5 & 4 & 5 & 4 & 0 \end{pmatrix}$$

Jedna z minimálních koster grafu G má množinu hran $H(T_1) = \{\{u_1, u_2\}, \{u_1, u_4\}, \{u_1, u_6\}, \{u_3, u_4\}, \{u_5, u_6\}\}$. Při jejím prohlédávání do hloubky dostaneme posloupnost uzlů $H_1 = u_1, u_2, u_4, u_3, u_6, u_5$ a odtud hamiltonovskou kružnici $C_1 = u_1 u_2 u_4 u_3 u_6 u_5 u_1$ s váhou $w(C_1) = 23$.

Jiná minimální kostra tohoto grafu má množinu hran $H(T_2) = \{\{u_1, u_4\}, \{u_1, u_6\}, \{u_2, u_3\}, \{u_3, u_4\}, \{u_3, u_5\}\}$; její prohlédávání do hloubky dává posloupnost uzlů $H_2 = u_1, u_4, u_3, u_2, u_5, u_6$ a hamiltonovskou kružnici $C_2 = u_1 u_4 u_3 u_2 u_5 u_6 u_1$ s váhou $w(C_2) = 18$.

Poznamenejme, že kružnice C_2 je optimálním řešením dané úlohy.

Následující tvrzení ukazuje, jak kvalitní odhad optimálního řešení tento algoritmus dává.

Věta 8.1. *Nechť G je ohodnocený úplný graf s ohodnocením, splňujícím trojúhelníkovou nerovnost, C' je hamiltonovská kružnice malé váhy v G , vytvořená naším algoritmem, a C je minimální hamiltonovská kružnice v G (tj. řešení úlohy obchodního cestujícího v G). Pak platí*

$$w(C') \leq 2w(C).$$

Důkaz. Nechť C je optimální hamiltonovská kružnice v grafu G , a C' je hamiltonovská kružnice, vytvořená naším algoritmem. Je-li T minimální kostra grafu G , pak jistě platí

$$w(T) = \sum_{h \in H(T)} w(h) \leq w(C),$$

neboť vyjmutím jedné hrany z C dostaneme kostru G , ale T je minimální kostra.

Nechť W je uzavřený sled, určený prohlédáváním kostry T do hloubky. Pak W prochází každou hranou kostry T právě dvakrát. Tedy

$$w(W) = \sum_{h \in H(W)} w(h) = 2w(T) \leq 2w(C).$$

Povšimneme si nyní toho, že kružnici C' lze z T sestrojít tak, že při zpětném chodu backtrackingu opakovaně místo každé dvojice následujících hran použijeme příslušnou chordu. Při každé náhradě dvojice hran $\{x, y\}, \{y, z\}$ jedinou hranou $\{x, z\}$ ale platí $w(x, z) \leq w(x, y) + w(y, z)$, a tedy suma ohodnocení se nezvětšuje. Proto $w(C') \leq w(W)$, odkud

$$w(C') \leq w(W) \leq 2w(C). \quad \square$$

Poznámka. Uvažujme nějaký optimalizační problém. O polynomiálním algoritmu, který pro každá přípustná vstupní data poskytne přibližné řešení, jehož váha je nejvýše k -násobkem váhy optimálního řešení, říkáme, že je *polynomiální k -aproximací* daného problému. Věta 8.1 tedy říká, že náš algoritmus je polynomiální 2-aproximací pro úlohu obchodního cestujícího v úplném grafu s trojúhelníkovou nerovností. Následující tvrzení ukazuje, že bez předpokladu trojúhelníkové nerovnosti nic takového nelze očekávat.

Věta 8.2. *Je-li $P \neq NP$, pak pro žádné přirozené číslo k neexistuje polynomiální k -aproximace úlohy obchodního cestujícího v úplném ohodnoceném grafu.*

Důkaz. Nejprve zavedeme následující označení.

k -TSP (k -aproximační problém TSP)

Vstup: úplný graf G s hranovým ohodnocením $w : H(G) \rightarrow (0, \infty)$ a přirozené číslo $k \geq 2$.

Úkol: najít kružnici C' v G takovou, že $w(C') \leq k \cdot w(C)$, kde C je optimální řešení úlohy obchodního cestujícího v grafu G .

Výstup: kružnice C' .

Je zřejmé, že polynomiální k -aproximace úlohy obchodního cestujícího v úplném ohodnoceném grafu by dávala polynomiální algoritmus pro úlohu k -TSP. K důkazu věty tedy postačí najít polynomiální převod

$$\text{HAM} \triangleleft k\text{-TSP},$$

tj. ukázat, že problém k -TSP je NP-těžký pro každé přirozené k .

Nechť je tedy dán graf G na $n = |U(G)|$ uzlech (o němž máme rozhodnout, zda je hamiltonovský), a libovolné přirozené číslo k . Grafu G přiřadíme úplný ohodnocený graf G'_k následujícím předpisem:

$$U(G'_k) = U(G),$$

$$H(G'_k) = \binom{U(G)}{2},$$

$$\text{pro } h \in H(G'_k) \text{ položíme } w(h) = \begin{cases} 1 & \text{pokud } h \in H(G), \\ kn + 1 & \text{jinak.} \end{cases}$$

Nyní je zřejmé, že

- je-li graf G hamiltonovský, pak v G'_k existuje kružnice váhy n , a tedy každá k -aproximace optima úlohy obchodního cestujícího v G'_k má váhu nejvýše kn ,
- jestliže G není hamiltonovský, pak každá kružnice v G'_k obsahuje alespoň jednu hranu s váhou $kn + 1$, a tedy má váhu alespoň $(n - 1) \cdot 1 + 1 \cdot (kn + 1) = (k + 1)n$.

Pomocí výstupu z problému k -TSP na grafu G'_k tedy můžeme rozhodnout, zda je graf G hamiltonovský: je-li výstupem k -TSP na G'_k kružnice C' , pak G je hamiltonovský, právě když $w(C') \leq kn$.

Zbývá ověřit, že převod $\text{HAM} \triangleleft k\text{-TSP}$ je polynomiální, ale to je zřejmé. \square

Na závěr tohoto odstavce si uvedeme ještě jeden (ještě jednodušší) algoritmus, o němž lze také dokázat, že za předpokladu trojúhelníkové nerovnosti poskytuje polynomiální 2-aproximaci úlohy obchodního cestujícího. Důkaz lze nalézt např. v knize [7].

Algoritmus (hamiltonovská kružnice malé váhy v úplném ohodnoceném grafu s trojúhelníkovou nerovností II)

1. $i := 1$.
2. Zvolme libovolný uzel u_1 a položme $C_1 = (\{u_1\}, \emptyset)$.

3. Je-li $i < n$, nalezneme uzly $v_i \in U(G) \setminus U(C_i)$ a $u_i \in U(C_i)$ takové, že $w(u_i, v_i)$ je minimální, jinak KONEC.
4. Sestrojíme C_{i+1} vložením uzlu v_i do kružnice C_i před uzel u_i .
5. $i := i + 1$, zpět na 3.

Poznámka. Uvedený algoritmus je v podstatě postupem z „rodiny“ hladových algoritmů.

Příklad. Mějme ohodnocený úplný graf G , popsany maticí $W(G)$:

$$W(G) = \begin{pmatrix} 0 & 3 & 3 & 2 & 7 & 3 \\ 3 & 0 & 3 & 4 & 5 & 5 \\ 3 & 3 & 0 & 1 & 4 & 4 \\ 2 & 4 & 1 & 0 & 5 & 5 \\ 7 & 5 & 4 & 5 & 0 & 4 \\ 3 & 5 & 4 & 5 & 4 & 0 \end{pmatrix}$$

Zvolme výchozí uzel, např. ten, který je reprezentován prvním sloupcem (nebo řádkem, matice je symetrická). Vytvořili jsme jednoduzlovou kružnici $C_1 = u_1$. Ze všech nenulových čísel v příslušném řádku nebo sloupci vybereme to nejmenší - nalezneme uzel u_4 - viz krok 3.

$$W(G) = \begin{pmatrix} 0 & 3 & 3 & \mathbf{2} & 7 & 3 \\ 3 & 0 & 3 & 4 & 5 & 5 \\ 3 & 3 & 0 & 1 & 4 & 4 \\ 2 & 4 & 1 & 0 & 5 & 5 \\ 7 & 5 & 4 & 5 & 0 & 4 \\ 3 & 5 & 4 & 5 & 4 & 0 \end{pmatrix}$$

Vytvořili jsme novou kružnici $C_2 = u_1 u_4 u_1$. Nyní hledáme minimum v řádcích 1 a 4 (vynecháme přitom sloupce 1 a 4, neboť každým uzlem projdeme právě jednou). Tak nalezneme uzel u_3 , který je danou hranou spojen s uzlem, v jehož řádku jsme tento prvek našli. Je to čtvrtý řádek, tedy $w(u_3, u_4) = 1$.

$$W(G) = \begin{pmatrix} 0 & 3 & 3 & 2 & 7 & 3 \\ 3 & 0 & 3 & 4 & 5 & 5 \\ 3 & 3 & 0 & 1 & 4 & 4 \\ 2 & 4 & \mathbf{1} & 0 & 5 & 5 \\ 7 & 5 & 4 & 5 & 0 & 4 \\ 3 & 5 & 4 & 5 & 4 & 0 \end{pmatrix}$$

Získáme kružnici $C_3 = u_1 u_3 u_4 u_1$. Podle kroku 4 v našem algoritmu jsme dali nový uzel před ten, s nímž je spojen danou hranou s minimálním ohodnocením. Opět hledáme minimální prvek, tentokrát v řádcích 1, 3, 4 s vynecháním stejných sloupců. Nalezneme uzel u_2 v 1. řádku. Zde jsme měli několik možností, vybrali jsme tu první.

$$W(G) = \begin{pmatrix} 0 & \mathbf{3} & 3 & 2 & 7 & 3 \\ 3 & 0 & 3 & 4 & 5 & 5 \\ 3 & 3 & 0 & 1 & 4 & 4 \\ 2 & 4 & 1 & 0 & 5 & 5 \\ 7 & 5 & 4 & 5 & 0 & 4 \\ 3 & 5 & 4 & 5 & 4 & 0 \end{pmatrix}$$

Opět vytvoříme novou kružnici $C_4 = u_1 u_3 u_4 u_2 u_1$. V dalších krocích bychom přidali oba zbylé uzly a vytvořili tak kružnici $C_6 = u_1 u_5 u_3 u_4 u_2 u_6 u_1$. Cena této kružnice $w(C_6) = 24$. Není to však optimální řešení, to má cenu $w(C) = 18$ a příslušná kružnice je $C = u_1 u_4 u_3 u_2 u_5 u_6 u_1$. V matici $W(G)$ se jedná o silně vyznačené prvky:

$$W(G) = \begin{pmatrix} 0 & 3 & 3 & \mathbf{2} & 7 & 3 \\ 3 & 0 & 3 & 4 & \mathbf{5} & 5 \\ 3 & \mathbf{3} & 0 & 1 & 4 & 4 \\ 2 & 4 & \mathbf{1} & 0 & 5 & 5 \\ 7 & 5 & 4 & 5 & 0 & \mathbf{4} \\ \mathbf{3} & 5 & 4 & 5 & 4 & 0 \end{pmatrix}$$

Reference

- [1] W.J. Cook, W.H. Cunningham, W.R. Pulleyblank, A. Schrijver: Combinatorial Optimization. Wiley - Interscience, New York, 1998.
- [2] R. Čada, T. Kaiser, Z. Ryjáček: Diskrétní matematika. Skripta ZČU Plzeň, 2004.
- [3] M.R. Garey, D.S. Johnson: Computers and Intractability – a Guide to the Theory of NP-Completeness. Bell Laboratories 1979.
- [4] J. Holenda, Z. Ryjáček: Lineární algebra II – Úvod do diskrétní matematiky. Skripta ZČU Plzeň, 1995.
- [5] J. Hromkovič: Algorithmics for Hard Problems. Springer, 2003.
- [6] B. Korte, J. Vygen: J. Combinatorial Optimization - Theory and Algorithms. Springer, 2003.
- [7] G. Chartrand, O.R. Oellerman: Applied and Algorithmic Graph Theory. Mc Graw-Hill, 1993.
- [8] L. Kučera: Kombinatorické algoritmy. Matematický seminář SNTL, Praha 1989.
- [9] L. Lovász, M.D. Plummer: Matching Theory. Akadémiai Kiadó, Budapest, 1986, a Annals of Discrete Math. 29 (1986).
- [10] C.H. Papadimitriou, K. Steiglitz: Combinatorial Optimization – Algorithms and Complexity. Prentice-Hall, 1982.
- [11] J. Plesník: Grafové algoritmy. Veda, Bratislava, 1983.
- [12] J. Plesník, J. Dupačová, M. Vlach: Lineárne programovanie. Alfa, Bratislava, 1990.