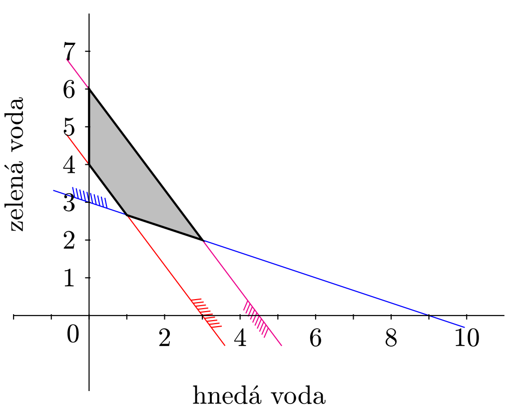
1 Lineárne programovanie

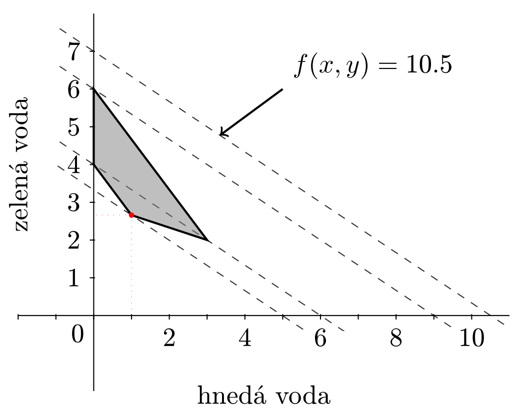
1.1 Pár úvodných slov

Začneme typickým príkladom, ktorý sa v tej či onej modifikácii objaví v každom texte o lineárnom programovaní. Zoberme si študenta, ktorý na prípravu na skúšku potrebuje dostať do tela aspoň 270 mg kofeínu a 120 g cukru. Zároveň nechce prekročiť dávku 180 mg aspartámu. K dispozícii má dva nápoje: hnedú vodu za €1/dl a zelenú vodu za €1,50/dl. Hnedá voda obsahuje 30 mg kofeínu, 40 g cukru a 40 mg aspartámu, kým zelená voda obsahuje 90 mg kofeínu, 30 g cukru a 30 mg aspartámu. Koľko dl ktorej vody si má študent kúpiť, aby čo najlacnejšie uspokojil svoje požiadavky? Ak si študent kúpi x dl hnedej vody a y dl zelenej vody, úlohu (nazývanú *lineárny program*) môžme formulovať takto:

$$
\begin{array}{rcccccl}
 & \multicolumn{3}{c}{\text{dl vody}} & & & \\
 & \text{hnedej} & & \text{zelenej} & & & \\
\text{minimalizovať} & x & + & 1.5y & =: & f(x,y) & \textit{cena} \\
\text{pri obmedzeniach} & 30x & + & 90y & \geq & 270 & \textit{kofeín} \\
 & 40x & + & 30y & \geq & 120 & \textit{cukor} \\
 & 40x & + & 30y & \leq & 180 & \textit{aspartám} \\
 & & & x,y & \geq & 0 &
\end{array}
\qquad (1)
$$

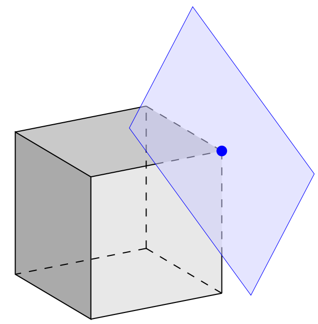
Ľahko vidno, že napr. 4 dl zelenej vody uspokoja všetky požiadavky za cenu €6, pričom kofeínu je aj viac, ako je nutné a aspartámu menej, ako je dovolené. Na druhej strane, ak by študent chcel kupovať iba hnedú vodu, na splnenie potreby kofeínu by jej musel kúpiť 9 dl, čím by ale prekročil prípustný príjem aspartámu (a navyše by to bolo drahé). Pri hľadaní optimálneho riešenia pomôže nasledovná geometrická reprezentácia: ak riešeniu s x dl hnedej a y dl zelenej vody priradíme bod v rovine so súradnicami (x, y), každé z obmedzení určuje polrovinu, v ktorej prípustné riešenie musí ležať. Do úvahy preto prichádzajú iba riešenia v štvoruholníku z ľavého obrázka:





Zároveň vieme, že $f(x,y) = x + 1.5y$ je lineárna funkcia, a preto body s rovnakou hodnotou funkcie f ležia na priamke (viď. pravý obrázok). Preto je zrejmé, že stačí overiť štyri vrcholy štvoruholníka a nájdeme optimálne riešenie, ktoré je v tomto prípade kúpiť 1 dl hnedej vody a $2\frac{2}{3}$ dl zelenej vody za €5.

Z týchto úvah ľahko odvodíme algoritmus na riešenie úlohy s dvoma premennými a obmedzeniami s nerovnosťami: pre každé obmedzenie zostrojíme príslušnú polrovinu, skonštruujeme mnohouholník, ktorý je prienikom všetkých polrovín a vyberieme optimálnu hodnotu spomedzi jeho vrcholov. Pre tri premenné obmedzenia tvaru $a_1x + a_2y + a_3z \geq b$ tvoria polpriestory, ktorých prienikom dostaneme mnohosten. Body s rovnakou hodnotou funkcie $f(x, y, z) = c_1x + c_2y + c_3z$ tvoria rovinu a preto na nájdenie optima stačí overiť všetky vrcholy mnohostena.



S narastajúcim počtom premenných začnú rásť aj problémy a je zrejmé, že priamočiarym zovšeobecňovaním sa ďaleko nedostaneme. Skúsme sa preto pozrieť na geometriu lineárneho programu trochu inak. Najprv si upravme program (1) do ekvivalentnej podoby takto: funkciu $f(x, y)$ vynásobíme -1 a z minimalizačnej úlohy dostaneme maximalizačnú. Potom prvé a druhé obmedzenie vynásobíme -1 a všetky obmedzenia (okrem tých na nezápornosť x, y) budú v tvare "$\leq$". Nakoniec, keďže v každom obmedzení je ľavá strana menšia ako pravá, môžme pridať novú nezápornú premennú, ktorej hodnota bude práve rozdiel ľavej a pravej strany a dostaneme nasledujúci program, ktorý je očividne ekvivalentný s pôvodným.

$$\begin{array}{rrrrrrcl}
\text{maximalizovať} & -x & -1.5y & & & & =: & f(x,y,s_1,s_2,s_3) \\
\text{pri obmedzeniach} & -30x & -90y & +s_1 & & & = & -270 \\
& -40x & -30y & & +s_2 & & = & -120 \\
& 40x & +30y & & & +s_3 & = & 180 \\
& & & & & x,y,s_1,s_2,s_3 & \geq & 0
\end{array} \tag{2}$$

Ak si označíme

$$\mathbf{c} = \begin{pmatrix} -1 \\ -1.5 \\ 0 \\ 0 \\ 0 \end{pmatrix} \qquad A = \begin{pmatrix} -30 & -90 & 1 & 0 & 0 \\ -40 & -30 & 0 & 1 & 0 \\ 30 & 40 & 0 & 0 & 1 \end{pmatrix} \qquad \mathbf{b} = \begin{pmatrix} -270 \\ -120 \\ 200 \end{pmatrix}$$

tak program (2) môžeme skrátene zapísať

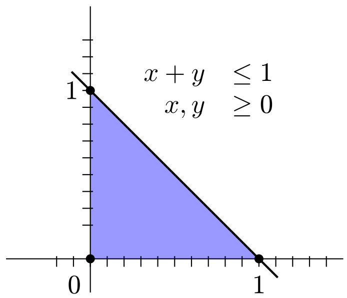
$$\max_{\mathbf{x} \in \mathbb{R}^5} \left\{ \mathbf{c}^{\mathsf{T}}\mathbf{x} \mid A\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq 0 \right\}.$$

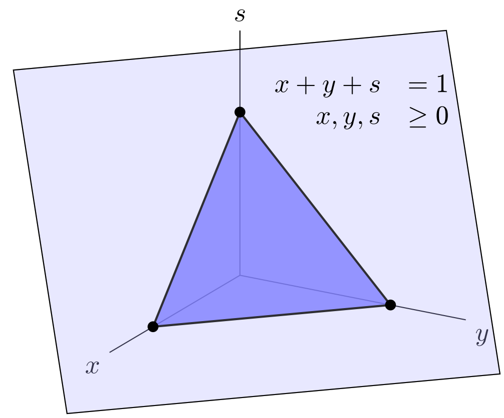
Týmto prepísaním sme zvýšili dimenziu problému z 2 na 5 (a tak sme stratili možnosť "obrázkového" riešenia), ale získali sme krajší tvar množiny prípustných riešení: sú to nezáporné riešenia systému lineárnych rovníc. V našom prípade má matica A hodnosť 3 (riadky sú lineárne nezávislé) a riešenia systému $A\mathbf{x} = \mathbf{b}$ tvoria dvojrozmerný podpriestor $\mathbf{o} + c_1\boldsymbol{\alpha} + c_2\boldsymbol{\beta}$ kde

$$\mathbf{o} = \begin{pmatrix} 0 \\ 0 \\ -270 \\ -120 \\ 180 \end{pmatrix} \qquad \boldsymbol{\alpha} = \begin{pmatrix} 1 \\ 0 \\ 30 \\ 40 \\ -40 \end{pmatrix} \qquad \boldsymbol{\beta} = \begin{pmatrix} 0 \\ 1 \\ 90 \\ 30 \\ -30 \end{pmatrix}.$$

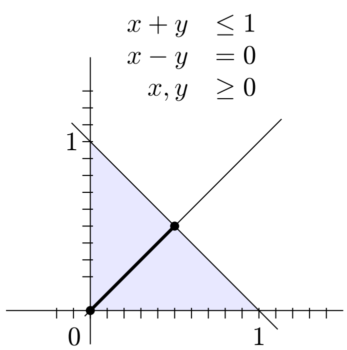
Inými slovami, prípustné riešenia programu (1) tvoria 2-rozmerný útvar (štvoruholník) v dvojrozmernom priestore (rovine), kým prípustné riešenia programu (2) tvoria 2-rozmerný útvar $\mathcal{D}$ v 5-rozmernom priestore, pričom $\mathcal{D}$ je prienikom (2-rozmernej) roviny a kladného ortantu[1]. Pre ilustráciu uvažujme nasledovné jednoduchšie programy (uvádzame iba obmedzenia, na účelovej funkcii v tomto prípade nezáleží):

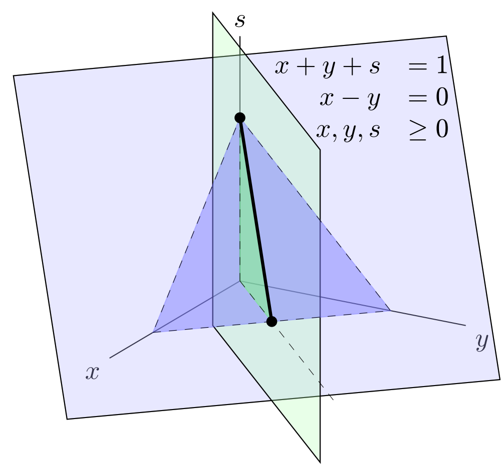
[1]ako je kvadrant v rovine a oktant v 3D priestore, používame slovo ortant v n-rozmernom priestore





Vľavo je program s dvoma premennými a dvojrozmerným priestorom riešení. Po transformácii do troch rozmerov priestor riešení ostal dvojrozmerný útvar a je prienikom roviny a kladného oktantu. V nasledujúcom prípade je v pôvodnom probléme priestor riešení jednorozmerný; po transformácii do trojrozmerného priestoru je priestor riešení stále jednorozmerný a je prienikom priamky a kladného oktantu.





Výhoda takto zapísaného programu je v tom, že sa nám budú ľahšie hľadať vrcholy telesa prípustných riešení: budú určite ležať v nejakej stene tvaru $x_i = 0$.

Skôr, ako pokročíme v našich úvahách je dobré si uvedomiť, že bez ujmy na všeobecnosti môžme predpokladať, že riadky matice A sú lineárne nezávislé: ak je nejaký riadok lineárnou kombináciou iných riadkov potom buď neexistuje žiadne prípustné riešenie, alebo jeho odstránením nijak nezmeníme priestor prípustných riešení. Naše úvahy môžme zhrnúť a zaviesť *normálny tvar* lineárneho programu takto:

Definícia 1.1. Lineárny program je v **normálnom tvare**, ak je zapísaný ako

$$\max_{\mathbf{x}\in\mathbb{R}^n} \left\{\mathbf{c}^\mathsf{T}\mathbf{x} \mid A\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0}\right\}, \tag{3}$$

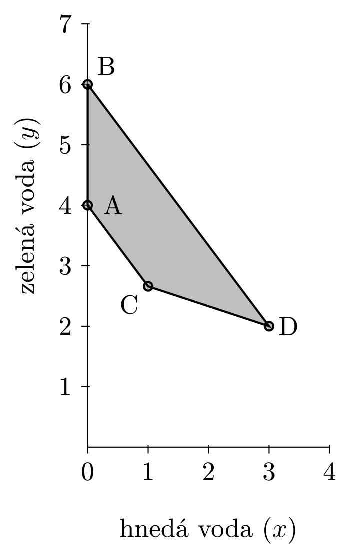
kde $A \in \mathbb{R}^{m\times n}$ má hodnosť m. Množina prípustných hodnôt $\mathcal{D} = \{\mathbf{x} \in \mathbb{R}^n \mid A\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0}\}$.

- **Cieľom je maximalizácia.** Ak bolo pôvodným cieľom minimalizovať lineárnu funkciu $f(\mathbf{x})$, novým cieľom bude maximalizovať lineárnu funkciu $-f(\mathbf{x})$.

- **Všetky premenné sú nezáporné.** Pre každú premennú x, ktorá nemá obmedzenie $x \geq 0$ pridáme dve premenné $p_x, q_x \geq 0$ a každý výskyt x nahradíme $p_x - q_x$.
- **Každé obmedzenie okrem tých tvaru $x \geq 0$ má tvar rovnosti.** Ak bolo pôvodné obmedzenie v tvare $\sum_i a_i x_i \geq b$, najprv ho prenásobením -1 upravíme na tvar $\sum_i -a_i x_i \leq -b$. Keď máme všetky nerovnosti v obmedzeniach otočené rovnakým smerom, pre každé obmedzenie tvaru $\sum_i a_i x_i \leq b$ zavedieme novú premennú s (*rezerva, slack*), ktorá reprezentuje hodnotu, o koľko je pôvodná ľavá strana menšia ako b. Ľahko vidno, že $s \geq 0$ a $s + \sum_i a_i x_i = b$.
- **Matica A má m lineárne nezávislých riadkov.**

Majme teraz program v normálnom tvare. Podobne ako v úvodnom príklade, chceme nájsť množinu vrcholov telesa $\mathcal{D}$ ohraničujúceho prípustné riešenia tak, aby stačilo overiť tieto vrcholy na nájdenie optimálneho riešenia. Keďže $\mathcal{D}$ je prienikom priestoru riešení $A\mathbf{x} = \mathbf{b}$ a kladného ortantu, vrcholy budú mať jednu alebo niekoľko súradníc $x_{i_1}, \ldots, x_{i_k}$ nulových (nadrovina $x_i = 0$ je na hranici ortantu). Navyše, vrcholy sú "špicaté" (na rozdiel od bodov v nejakej stene $x_i = 0$, v ktorej leží napr. celá úsečka). To, že vrchol je "špicatý", formulujeme tak, že ak k obmedzeniam $A\mathbf{x} = \mathbf{b}$ pridáme navyše $x_{i_1} = 0, \ldots, x_{i_k} = 0$, dostaneme jednoznačné riešenie (t.j. vrchol leží na prieniku nejakých stien ortantu a žiaden iný bod v tom istom prieniku neleží). Keďže A má hodnosť m, na to, aby sme dostali jednoznačné riešenie, musí byť $k = n - m$.
Vráťme sa teraz k programu (2). Hodnosť matice A je 3, preto vrcholy dostaneme tak, že k systému $A\mathbf{x} = \mathbf{b}$ pridáme dve obmedzenia $x_i = 0$ a $x_j = 0$. Dostávame 10 rôznych systémov lineárnych rovníc s nasledovnými riešeniami:

obmedzenia	x	y	s_1	s_2	s_3	
$x = y = 0$	0	0	-270	-120	180	
$x = s_1 = 0$	0	3	0	-30	90	
$x = s_2 = 0$	0	4	90	0	60	A
$x = s_3 = 0$	0	6	270	60	0	B
$y = s_1 = 0$	9	0	0	240	-180	
$y = s_2 = 0$	3	0	-180	0	60	
$y = s_3 = 0$	4.5	0	-135	60	0	
$\mathbf{s_1 = s_2 = 0}$	**1**	$\mathbf{\frac{8}{3}}$	**0**	**0**	**60**	C
$s_1 = s_3 = 0$	3	2	0	60	0	D
$s_2 = s_3 = 0$	neexistuje riešenie					



Riešenia systému $A\mathbf{x} = \mathbf{b}$ tvoria rovinu v 5-rozmernom priestore. Premenné x, y zodpovedajú množstvu kúpenej hnedej a zelenej vody, premenné s_1, s_2, s_3 udávajú rezervu, ktorá ostáva k prekročeniu príslušného obmedzenia. Takže napríklad štvrtý riadok tabuľky s obmedzeniami $x = s_3 = 0$ hovorí, že ak študent nekupuje žiadnu hnedú vodu a zároveň chce presne dosiahnuť povolené množstvo aspartámu, musí kúpiť 6 dl zelenej vody, pričom dostane viac kofeínu a cukru ako potrebuje. V poslednom riadku vidno, že obmedzenia sa nedajú pridávať ľubovoľne, ale treba dávať pozor, aby pridaním obmedzení nevznikli lineárne závislé riadky (v našom prípade sú červená a fialová priamka z prvého obrázka rovnobežné, takže neexistuje žiaden bod v ich priesečníku). V reči lineárnych programov sa riadkom tejto tabuľky hovorí *bázové riešenia* a tie z nich, ktoré sú zároveň prípustné (zvýraznené riadky) sú *prípustné bázové riešenia* a zodpovedajú vrcholom $\mathcal{D}$, t.j. prípustné riešenia programu (2) tvoria dvojrozmerný štvoruholník v päťrozmernom priestore. Za povšimnutie stojí, že prípustné bázové riešenia programu (2), t.j. vrcholy štvoruholníka prípustných riešení v päťrozmernom priestore, zodpovedajú vrcholom štvoruholníka prípustných riešení programu (1): každý vrchol štvoruholníka vpravo leží na priesečníku dvoch priamok, z ktorých každá zodpovedá obmedzeniu tvaru $x_i = 0$ (a príslušné bázové riešenie je prípustné). Naopak, každé prípustné bázové riešenie leží na priesečníku dvoch takýchto priamok.

Keď si uvedomíme, že pridaním obmedzenia tvaru $x = 0$ vlastne pri riešení príslušného systému vymažeme stĺpec premennej x z matice dostaneme nasledovnú definíciu.

Označenie. Majme maticu $A \in \mathbb{R}^{m\times n}$ s m riadkami a n stĺpcami. Pre množinu $B \subseteq \{1, 2, \dots, n\}$ označíme A_B podmaticu A, ktorá pozostáva zo stĺpcov indexovaných množinou B. Rovnakú notáciu $\mathbf{x}_B$ budeme používať pre vektory.

Napríklad

$$A = \begin{pmatrix} -30 & -90 & 1 & 0 & 0 \\ -40 & -30 & 0 & 1 & 0 \\ 30 & 40 & 0 & 0 & 1 \end{pmatrix} \qquad A_{\{1,2\}} = \begin{pmatrix} -30 & -90 \\ -40 & -30 \\ 30 & 40 \end{pmatrix} \qquad A_{\{3,4,5\}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Definícia 1.2. Majme lineárny program v normálnom tvare, kde $A \in \mathbb{R}^{m\times n}$. **Bázové riešenie** je vektor $\mathbf{x} \in \mathbb{R}^n$, pre ktorý existuje m-prvková množina $B \subseteq \{1, \dots, n\}$ taká, že

1. matica $A_B \in \mathbb{R}^{m\times m}$ má hodnosť m (t.j. je regulárna)
2. $x_j = 0$ pre všetky $j \notin B$

Teraz ukážeme, že naša predstava bázového riešenia ako vrchola je dobrá v tom, že na nájdenie optima stačí overiť prípustné bázové riešenia.

Veta 1.3. *Majme daný lineárny program v normálnom tvare, pričom hodnota účelovej funkcie je* $\mathbf{c}^\mathsf{T}\mathbf{x}$ *na telese* $\mathcal{D}$ *je zhora ohraničená. Potom pre každé prípustné riešenie* $\mathbf{x_0}$ *existuje prípustné bázové riešenie* $\tilde{\mathbf{x}}$*, pre ktoré* $\mathbf{c}^\mathsf{T}\tilde{\mathbf{x}} \geq \mathbf{c}^\mathsf{T}\mathbf{x}$.

Dôkaz:. Zoberme si ľubovoľné prípustné riešenie $\mathbf{x_0}$ a uvažujme všetky také prípustné riešenia $\mathbf{x}$, pre ktoré $\mathbf{c}^\mathsf{T}\mathbf{x} \geq \mathbf{c}^\mathsf{T}\mathbf{x_0}$. Za $\tilde{\mathbf{x}}$ vyberme také z nich, ktoré má najväčší počet nulových zložiek. Ukážeme, že $\tilde{\mathbf{x}}$ je bázové. Ak $\tilde{\mathbf{x}} = \mathbf{0}$, je zrejme bázové. Nech teda $\tilde{\mathbf{x}}$ má aspoň jednu nenulovú zložku. Označme $K = \{j \in \{1, \dots, n\} \mid \tilde{x}_j > 0\}$ množinu kladných (žiadne prípustné riešenie nemá záporné zložky) zložiek vektora $\tilde{\mathbf{x}}$ a uvažujme dva prípady.

• *Stĺpce matice A_K sú lineárne nezávislé.* Zjavne $|K| \leq m$ (matica A má m riadkov). Ak $|K| = m$, $\tilde{\mathbf{x}}$ v zhode s Definíciou 1.2 má $\tilde{x}_j = 0$ pre všetky $j \notin K$ a matica A_K je regulárna. Ak $|K| < m$, môžeme $|K|$ stĺpcov matice A_K doplniť $m-k$ stĺpcami z A tak, aby boli lineárne nezávislé[2]. Takže dostaneme množinu K' tak, že $|K'| = m$, $A_{K'}$ je regulárna a $\tilde{x}_j = 0$ pre všetky $j \notin K' \supseteq K$.

• *Stĺpce matice A_K sú lineárne závislé,* to znamená, že existuje vektor $\boldsymbol{\vartheta} \in \mathbb{R}^{|K|}$ taký, že $A_K\boldsymbol{\vartheta} = \mathbf{0}$ ($\boldsymbol{\vartheta}$ určuje lineárnu kombináciu stĺpcov A_K, ktorej výsledkom je nulový vektor). Doplňme $\boldsymbol{\vartheta}$ na n-rozmerný vektor $\mathbf{w}$ tak, že na miesta mimo množiny K dosadíme 0, takže $\mathbf{w}_K = \boldsymbol{\vartheta}$ a $A\mathbf{w} = 0$. Pre ľubovoľné reálne $t \geq 0$ označme $\mathbf{x}(t) = \tilde{\mathbf{x}} + t\mathbf{w}$. Keďže $\tilde{\mathbf{x}}$ je prípustné riešenie, platí $A\tilde{\mathbf{x}} = \mathbf{b}$. Zároveň platí $A\mathbf{w} = \mathbf{0}$ a teda aj $A\mathbf{x}(t) = \mathbf{b}$.
Prv, než budeme pokračovať v dôkaze, upravíme vektor $\mathbf{w}$ tak, aby $\mathbf{c}^\mathsf{T}\mathbf{w} \geq 0$ a zároveň $w_j < 0$ pre nejaké $j \in K$. Ak $\mathbf{c}^\mathsf{T}\mathbf{w} = 0$ a pre všetky $j \in K$ platí $w_j > 0$, stačí $\mathbf{w}$ prenásobiť -1 a máme ho v požadovanom tvare. Nech teda $\mathbf{c}^\mathsf{T}\mathbf{w} \neq 0$. Ak $\mathbf{c}^\mathsf{T}\mathbf{w} < 0$, môžme opäť $\mathbf{w}$ prenásobiť -1, takže bez ujmy na všeobecnosti nech $\mathbf{c}^\mathsf{T}\mathbf{w} > 0$. Ukážeme, že teraz musí také existovať $j \in K$, že $w_j < 0$. Ak by to tak nebolo, t.j. ak pre všetky $j \in K$ je $w_j > 0$, zjavne $\mathbf{w} \geq \mathbf{0}$ (zložky $i \notin K$ sme doplnili nulami). Potom ale $\mathbf{x}(t) = \tilde{\mathbf{x}} + t\mathbf{w} \geq 0$ pre všetky $t \geq 0$, takže $\mathbf{x}(t)$ je prípustné riešenie. Hodnota účelovej funkcie je $\mathbf{c}^\mathsf{T}\mathbf{x}(t) = \mathbf{c}^\mathsf{T}\tilde{\mathbf{x}} + t\mathbf{c}^\mathsf{T}\mathbf{w}$. Keďže $\mathbf{c}^\mathsf{T}\mathbf{w} > 0$, pre $t \mapsto \infty$ je $\mathbf{c}^\mathsf{T}\mathbf{x}(t) \mapsto \infty$, a teda lineárny program nebol ohraničený.

[2] Toto tvrdenie je súčasťou základného kurzu algebry.

Majme teraz vektor $\mathbf{w}$ upravený tak, že spĺňa $\mathbf{c}^\mathsf{T}\mathbf{w} \geq 0$ a zároveň $w_j < 0$ pre nejaké $j \in K$. Ukážeme, že pre nejaké $t_1 > 0$ je vektor $\mathbf{x}(t_1)$ prípustné riešenie s viacerými nulovými zložkami ako $\tilde{\mathbf{x}}$. To bude ale v spore s tým, že $\tilde{\mathbf{x}}$ má najviac nulových zložiek spomedzi všetkých prípustných riešení $\mathbf{x}$, pre ktoré $\mathbf{c}^\mathsf{T}\mathbf{x} \geq \mathbf{c}^\mathsf{T}\mathbf{x_0}$, pretože $\mathbf{c}^\mathsf{T}\mathbf{x}(t_1) = \mathbf{c}^\mathsf{T}\tilde{\mathbf{x}} + t_1\mathbf{c}^\mathsf{T}\mathbf{w} \geq \mathbf{c}^\mathsf{T}\mathbf{x_0}$ (lebo $\mathbf{c}^\mathsf{T}\tilde{\mathbf{x}} \geq \mathbf{c}^\mathsf{T}\mathbf{x_0}$ a $\mathbf{c}^\mathsf{T}\mathbf{w} \geq 0$).
Vektor $\mathbf{x}(t_0) = \tilde{\mathbf{x}}$ je prípustné riešenie a má zložky $j \in K$ (ostro) kladné a zvyšné zložky nulové. Zároveň vieme, že existuje aspoň jedno $j \in K$, kde $w_j < 0$. Keďže j-ta zložka $\mathbf{x}(t)$ je $x(t)_j = \tilde{x}_j + tw_j$, s rastúcim t klesajú hodnoty $x(t)_j$ pre všetky j, kde $w_j < 0$. Zvoľme za t_1 také t, keď prvá z hodnôt $x(t)_j$ dosiahne 0. Zjavne $\mathbf{x}(t_1)$ je prípustné riešenie a má viac nulových zložiek ako $\tilde{\mathbf{x}}$. $\square$

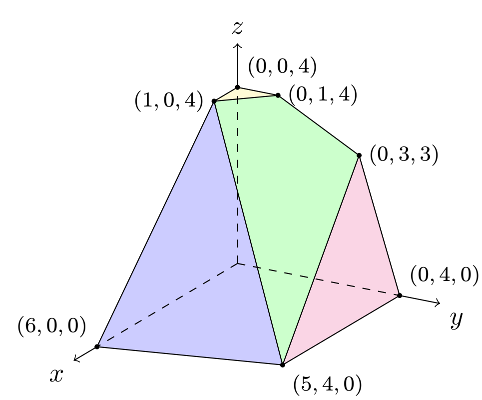
Dôsledkom tejto vety je, že na nájdenie optimálneho riešenia pre lineárne programy, ktoré majú konečné optimum, stačí prehľadať všetky prípustné bázové riešenia. Toto prehľadávanie je zovšeobecnením prístupu v dvoch rozmeroch z úvodného príkladu, kde stačilo prehľadať vrcholy vhodného mnohouholníka. Ako sa dajú bázové riešenia nájsť? Stačí si uvedomiť, že pre danú množinu $B \subseteq \{1, \ldots, n\}$ existuje najviac jedno bázové riešenie[3]: ak by boli dve bázové riešenia $\mathbf{y}, \mathbf{z}$ s tou istou množinou B, musí platiť $A\mathbf{y} = A\mathbf{z} = \mathbf{b}$ a teda $A_B\mathbf{y} = A_B\mathbf{z} = \mathbf{b}$. Keďže A je regulárna štvorcová matica, systém $A_B\mathbf{x} = \mathbf{b}$ má jednoznačné riešenie a preto $\mathbf{y} = \mathbf{z}$. Preto stačí vyskúšať všetky množiny B, overiť, či príslušná A_B je regulárna (napr. Gaussovou elimináciou), overiť, či je získané riešenie $A_B\mathbf{x} = \mathbf{b}$ prípustné a spomedzi všetkých takto získaných riešení $\mathbf{x}$ vybrať to najlepšie. Problém s týmto algoritmom je, že pri n premenných a m obmedzeniach môže byť potenciálne $\binom{n}{m}$ rôznych bázových riešení, a teda vo všeobecnosti nie je polynomiálny[4]. V nasledujúcej časti si ukážeme, ako úlohu lineárneho programovania riešiť efektívnejšie pomocou simplexovej metódy.

1.2 Simplexová metóda

Ako simplexový algoritmus sa označuje každý taký greedy algoritmus, ktorý prechádza bázové riešenia (vrcholy $\mathcal{D}$) tak, že vždy sa presunie po hrane smerom, v ktorom rastie (pri maximalizácii) hodnota účelovej funkcie. Začnime opäť príkladom. Uvažujme nasledovný lineárny program:

$$\begin{array}{rrrrl}
\text{maximalizovať} & x & +y & +z & =: f(x,y,z) \\
\text{pri obmedzeniach} & x & +y & +2z & \leq 9 \\
 & 4x & +y & +5z & \leq 24 \\
 & & 3y & +z & \leq 12 \\
 & & & z & \leq 4 \\
 & & & x,y,z & \geq 0
\end{array} \tag{4}$$

Obmedzenia tvoria polpriestory v trojrozmernom priestore. Hraničné roviny (v poradí zelená, modrá, červená a žltá) vymedzujú mnohosten $\mathcal{D}$. Preskúšaním všetkých vrcholov $\mathcal{D}$ zistíme, že maximum sa dosahuje v bode $(5, 4, 0)$.



[3]Naopak to neplatí, to isté bázové riešenie $\mathbf{x}$ je možné dostať z rôznych množín B, B'. Ak napríklad vektor $\mathbf{0}$ je prípustné riešenie, potom je aj prípustné bázové riešenie pre ľubovoľnú množinu B.

[4]Napríklad pre $m = n/2$ sa zo Stirlingovej aproximácie $n! \approx \sqrt{2\pi n}\left(\frac{n}{e}\right)^n(1 + o(n))$ ľahko ukáže, že $\binom{n}{\frac{n}{2}} = \frac{n!}{[(\frac{n}{2})!]^2} \geq 2^n/n^2$.

Podobne ako v predchádzajúcej časti si zavedieme rezervné (*slack*) premenné $s_1, \dots, s_4$; ak si premenné očíslujeme v poradí $x, y, z, s_1, \dots, s_4$, dostaneme ekvivalentný program v normálnom tvare

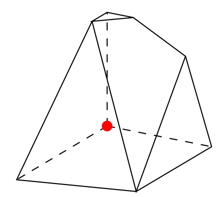
$$\max_{\mathbf{x} \in \mathbb{R}^7} \left\{ \mathbf{c}^\mathsf{T} \mathbf{x} \mid A\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq 0 \right\}$$

kde

$$\mathbf{c} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \qquad A = \begin{pmatrix} 1 & 1 & 2 & 1 & 0 & 0 & 0 \\ 4 & 1 & 5 & 0 & 1 & 0 & 0 \\ 0 & 3 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \qquad \mathbf{b} = \begin{pmatrix} 9 \\ 24 \\ 12 \\ 4 \end{pmatrix}$$

Máme 7 premenných a matica A má hodnosť 4, teda riešenia systému $A\mathbf{x} = \mathbf{b}$ tvoria trojrozmerný podpriestor v sedemrozmernom priestore. Špeciálne, môžme si vybrať ľubovoľné tri premenné ako parametre a ostatné premenné (aj hodnotu funkcie) vyjadriť pomocou nich. V našom prípade je matica $A_{4,5,6,7}$ diagonálna, takže ľahko vidno, že program (4) je ekvivalentne zápisateľný takto:

$$\begin{array}{rllll} f = & & x & +\,y & +\,z \\ \hline s_1 = & 9 & -\,x & -\,y & -\,2z \\ s_2 = & 24 & -\,4x & -\,y & -\,5z \\ s_3 = & 12 & & -\,3y & -\,z \\ s_4 = & 4 & & & -\,z \end{array} \tag{5}$$

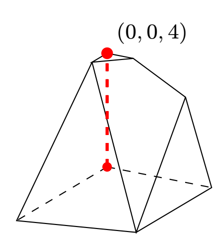


Zápisu (5) budeme hovoriť *tablo* a znamená toto: hľadáme parametre x, y, z tak, aby hodnota f bola maximálna a pritom parametre $s_1, \dots, s_4$ boli nezáporné. V našom prípade vidno, že pri voľbe $x = y = z = 0$ bude nezápornosť $s_1, \dots, s_4$ splnená. Tablo (5) preto bude reprezentovať bázové riešenie $(0, 0, 0, 9, 24, 12, 4)$ pre bázu $\{s_1, s_2, s_3, s_4\}$ s hodnotou funkcie $f = 0$. Bázové premenné sú v riadkoch a nebázové nulové premenné sú parametre v stĺpcoch.

Obrázok vpravo graficky reprezentuje bázové riešenie z tabla (5). Síce program, ktorý riešime, má 7 rozmerov, ale, podobne ako v predchádzajúcej časti, ľahko vidno, že každému prípustnému bázovému riešeniu jednoznačne prislúcha jeden vrchol mnohostena v trojrozmernom priestore (x, y, z). Naším cieľom je nájsť najlepšie prípustné bázové riešenie, resp. jeho bázu. Chceme teda nájsť nejaké tri nebázové premenné (parametre), ktoré sa nastavia na 0 a z vyjadrenia ostatných premenných (a účelovej funkcie) pomocou týchto parametrov vieme určiť ich hodnoty. Vyskúšanie všetkých trojíc parametrov by preto zodpovedalo prehľadaniu všetkých bázových riešení (vrcholov mnohostenu). Tomuto sa ale snažíme vyhnúť.

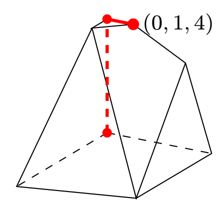
Ako môžme lokálne zväčšiť hodnotu funkcie f? Vidíme, že napr. premenná z je v prvom riadku s kladným koeficientom, takže ak zvýšime hodnotu z, vzrastie aj f. Ako veľa môžme z zväčšiť? Všetky premenné $s_1, \dots, s_4$ musia zostať nezáporné, takže každý riadok nám dáva limit na maximálnu hodnotu z v poradí $\frac{9}{2}$, $\frac{24}{5}$, 12, 4. Môžme teda nastaviť $z = 4$, čím dostaneme, že $s_4 = 0$. Naše riešenie sa teda zmenilo tak, že nebázové premenné (parametre) budú x, y, s_4 namiesto x, y, z. Prispôsobíme tomu aj náš zápis tak, že z rovnice pre s_4 vyjadríme z a dosadíme do ostatných rovníc. Dostaneme tak zápis

$$\begin{array}{rllll} f = & 4 & +\,x & +\,y & -\,s_4 \\ \hline z = & 4 & & & -\,s_4 \\ s_1 = & 1 & -\,x & -\,y & +\,2s_4 \\ s_2 = & 4 & -\,4x & -\,y & +\,5s_4 \\ s_3 = & 8 & & -\,3y & +\,s_4 \end{array} \tag{6}$$



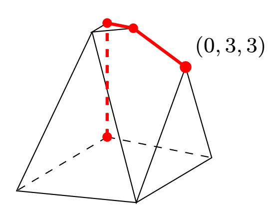
Tablo (6) zodpovedá bázovému riešeniu $(0,0,4,1,4,8,0)$ pre bázu $\{z, s_1, s_2, s_3\}$ s hodnotou funkcie $f = 4$. Opäť máme zápis, kde v riadkoch sú bázové premenné a parametre stĺpcov sú nebázové premenné. Cieľom je nájsť parametre x, y, s_4 tak aby sa maximalizovala hodnota f a zároveň s_1, s_2, s_3, z ostali nezáporné. Je kľúčové si uvedomiť, že sme urobili iba ekvivalentnú úpravu systému lineárnych rovníc a teda riešenia (5) a (6) sú rovnaké. Krok z (5) do (6) zodpovedá prejdeniu po jednej hrane mnohostena riešení. Budeme ho volať *pivotný krok*: jedna nebázová premenná, *pivot*, v našom prípade z, sa presunula do bázy a jedna bázová premenná (v našom prípade s_4) sa stala nebázovou. Tento postup môžme opakovať. Vidíme napr., že y je v prvom riadku s kladným koeficientom a teda jeho zväčšením zväčšíme f. Jednotlivé riadky nám dávajú obmedzenia na maximálnu hodnotu y v poradí $1, 4, \frac{8}{3}$; v poslednej rovnosti y nevystupuje a preto naň nekladie ani žiadne obmedzenia. Zvolíme $y = 1$ a urobíme pivotný krok s pivotom y, pri ktorom y vystrieda v báze s_1. Vyjadríme $y = 1 - x - s_1 + 2s_4$ a po dosadení dostaneme

$$
\begin{array}{rrrrr}
f = & 5 & & -\, s_1 & +\, s_4 \\
\hline
y = & 1 & -\, x & -\, s_1 & +\, 2s_4 \\
z = & 4 & & & -\, s_4 \\
s_2 = & 3 & -\, 3x & +\, s_1 & +\, 3s_4 \\
s_3 = & 5 & +\, 3x & +\, 3s_1 & -\, 5s_4
\end{array}
\qquad (7)
$$



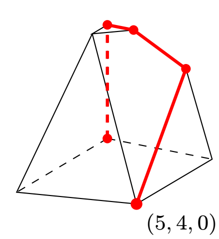
Máme tablo pre bázové riešenie $(0,1,4,0,3,5,0)$ pre bázu $\{y, z, s_2, s_3\}$ s hodnotou funkcie $f = 5$. Pokračujeme ďalej; jediná možnosť, ako zvýšiť f je zvoliť pivota s_4 a nahradiť ním v báze s_3. Dostaneme

$$
\begin{array}{rrrrr}
f = & 6 & +\, \frac{3}{5}x & -\, \frac{2}{5}s_1 & -\, \frac{1}{5}s_3 \\
\hline
y = & 3 & +\, \frac{1}{5}x & +\, \frac{1}{5}s_1 & -\, \frac{2}{5}s_3 \\
z = & 3 & -\, \frac{3}{5}x & -\frac{3}{5}s_1 & +\, \frac{1}{5}s_3 \\
s_2 = & 6 & -\, \frac{6}{5}x & +\, \frac{14}{5}s_1 & -\, \frac{3}{5}s_3 \\
s_4 = & 1 & +\, \frac{3}{5}x & +\, \frac{3}{5}s_1 & -\, \frac{1}{5}s_3
\end{array}
\qquad (8)
$$



Opäť jediná možnosť, ako spraviť pivotný krok, je zobrať do bázy x. Pri nastavení $x = 5$ sa ale vynulujú z aj s_2 a môžme si vybrať, ktoré z nich v báze ponecháme. Nech x vystrieda v báze z, dostaneme

$$
\begin{array}{rrrrr}
f = & 9 & -\, z & -\, s_1 & \\
\hline
x = & 5 & -\, \frac{5}{3}z & -\, s_1 & +\, \frac{1}{3}s_3 \\
y = & 4 & -\, \frac{1}{3}z & & -\, \frac{1}{3}s_3 \\
s_2 = & & 2z & +\, 4s_1 & -\, s_3 \\
s_4 = & 4 & -\, z & &
\end{array}
\qquad (9)
$$



Dostali sme sa do situácie, keď nie je možné urobiť žiaden pivotný krok. Vieme ale, že sme robili iba ekvivalentné úpravy, a teda riešenia programov (4) a (9) sú rovnaké (programy majú rovnakú množinu prípustných riešení a rovnaké hodnoty účelovej funkcie). Lenže z (9) jasne vidno, že pre ľubovoľné nezáporné z a s_1, hodnota f je vždy nanajvýš 9, takže nájdené riešenie je optimálne.

Tento príklad môžme zovšeobecniť. Formálne si tablo prislúchajúce prípustnému bázovému riešeniu môžme zadefinovať takto:

Definícia 1.4. Majme program v normálnom tvare

$$\max_{\mathbf{x} \in \mathbb{R}^n} \left\{ \mathbf{c}^\mathsf{T}\mathbf{x} \mid A\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq 0 \right\}$$

kde $A \in \mathbb{R}^{m \times n}$, a jeho prípustné bázové riešenie prislúchajúce báze B. **Tablo** $\mathcal{T}(B)$ prislúchajúce báze B je systém $m+1$ lineárnych rovníc v premenných $x_1, \ldots, x_n, f$, ktorý má rovnakú množinu riešení ako systém $A\mathbf{x} = \mathbf{b}, f = \mathbf{c}^\mathsf{T}\mathbf{x}$ a v maticovom zápise vyzerá

$$\begin{array}{cccccc} f & = & f_0 & + & \mathbf{r}^\mathsf{T}\mathbf{x}_N \\ \hline \mathbf{x}_B & = & \mathbf{p} & + & Q\,\mathbf{x}_N \end{array}$$

kde $\mathbf{x}_B$ je vektor bázových premenných, $N = \{1, \ldots, n\} - B$, $\mathbf{x}_N$ je vektor nebázových premenných, $\mathbf{r} \in \mathbb{R}^{n-m}$, $\mathbf{p} \in \mathbb{R}^m$ a $Q \in \mathbb{R}^{m \times n-m}$.

Keďže sme v definícii tabla vychádzali z prípustného bázového riešenia B, zrejme $\mathbf{p} \geq \mathbf{0}$. Ľahko sa presvedčíme, že takáto definícia tabla je korektná. Stačí si uvedomiť, že $\mathbf{b} = A\mathbf{x} = A_B\mathbf{x}_B + A_N\mathbf{x}_N$ a A_B je regulárna, takže existuje inverzná matica A_B^{-1}. Preto platí $\mathbf{x}_B = A_B^{-1}\mathbf{b} - A_B^{-1}A_N\mathbf{x}_N$ a dostávame nasledujúcu lemu, ktorej podrobný dôkaz prenechávame na čitateľa:

Lema 1.5. *Každému prípustnému bázovému riešeniu B programu z definície 1.4 prislúcha práve jedno tablo $\mathcal{T}(B)$ a platí*

$$\mathbf{p} = A_B^{-1}\mathbf{b} \qquad Q = -A_B^{-1}A_N \qquad f_0 = \mathbf{c}_B^\mathsf{T}A_B^{-1}\mathbf{b} \qquad \mathbf{r} = \mathbf{c}_N - (\mathbf{c}_B^\mathsf{T}A_B^{-1}A_N)^\mathsf{T}.$$

Z rovnakých úvah, ako sme robili v úvodnom príklade vyplýva

Tvrdenie 1.6. Nech B je báza prípustného riešenia a nech v $\mathcal{T}(B)$ je $\mathbf{r} \leq \mathbf{0}$. Potom f_0 je maximálna hodnota daného programu.

Na to, aby sme dokončili opis simplexového algoritmu, potrebujeme definovať pivotný krok: vybrať nejakú nebázovú premennú, ktorá je v $\mathbf{r}^\mathsf{T}\mathbf{x}_N$ s kladným koeficientom, zvýšiť ju koľko sa dá tak, aby bázové premenné ostali nezáporné a zmeniť bázu. Označme si $B = \{\beta_1, \ldots, \beta_m\}$ tak, že $\beta_1 < \beta_2 < \cdots < \beta_m$ a podobne $N = \{\mu_1, \ldots, \mu_{n-m}\}$, kde $\mu_1 < \mu_2 < \cdots < \mu_{n-m}$. V tomto označení môžme tablo rozpísať ako

$$\begin{array}{ccccc} f & = & f_0 & + & \sum\limits_{j=1}^{n-m} r_j x_{\mu_j} \\ \hline x_{\beta_1} & = & p_1 & + & \sum\limits_{j=1}^{n-m} q_{1,j} x_{\mu_j} \\ \vdots & & \vdots & & \\ x_{\beta_m} & = & p_m & + & \sum\limits_{j=1}^{n-m} q_{m,j} x_{\mu_j} \end{array} \tag{10}$$

Za pivota môžme zobrať hocijakú premennú x_{μ_j} takú, že $r_j > 0$. Ak je $q_{i,j} > 0$, i-ty riadok nekladie žiadne obmedzenia, inak musí platiť $p_i + q_{i,j}x_{\mu_j} > 0$. Dostávame sa tak k nasledovnej definícii:

Definícia 1.7. Majme daný lineárny program v normálnom tvare a bázu B prislúchajúcu prípustnému riešeniu. Nech $\mathcal{T}(B)$ je zapísané ako v (10) a nech $r_e > 0$ pre nejaké e. Označme

$$s := \min_{i=1,\ldots,m} \left\{ -\frac{p_i}{q_{i,e}} \mid q_{i,e} < 0 \right\}.$$

> **Pivotný krok** podľa premennej x_{μ_e} zmení bázu B na bázu
>
> $$B' := (B - \{\beta_\ell\}) \cup \{\mu_e\},$$
>
> kde β_ℓ je ľubovoľný index, pre ktorý platí $q_{\ell,e} < 0$ a $-\frac{p_\ell}{q_{\ell,e}} = s$.

Čitateľ sa ľahko presvedčí, že B' je opäť báza prípustného riešenia. Simplexový algoritmus začína z nejakého prípustného riešenia s bázou B_0 a aplikuje pivotné kroky, kým sa dá. Podľa tvrdenia 1.6, ak algoritmus nájde bázu B, pre ktorú je $\mathbf{r} \leq \mathbf{0}$, tak našiel optimálne riešenie. Aby sme ukázali korektnosť simplexovej metódy, potrebujeme vyriešiť tri problémy: jednak ukázať, ako nájsť B_0, dvak rozhodnúť, čo robiť, keď sa nedá urobiť žiaden pivotný krok a napokon ukázať, že algoritmus v konečnom čase skončí.

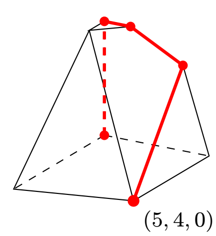
Čo sa stane, keď sa nedá vybrať pivot

Definícia pivotného kroku (Definícia 1.7) vyžaduje, aby pre pivota x_{μ_e} platilo $r_e > 0$. Ak neexistuje x_{μ_e}, pre ktoré $r_e > 0$, podľa Tvrdenia 1.6 máme optimálne riešenie. Ďalej musí platiť, že pivot nahradí v báze premennú x_{β_ℓ}, pre ktorú $q_{\ell,e} < 0$. Ak také ℓ neexistuje, t.j. ak $q_{\ell,e} \geq 0$ pre všetky ℓ, znamená to, že žiaden riadok tabla nekladie limit na zväčšovanie premennej x_{μ_e}. S rastúcim x_{μ_e} rastie aj hodnota f, a preto daný program nemá konečné maximum.

Ako sa nezacykliť

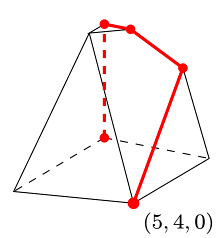
V úvodnom príklade sme v každom pivotnom kroku zväčšili hodnotu f. Ak by sme to vedeli zaručiť vždy, ľahko vidno, že algoritmus v konečnom čase skončí: je totiž iba konečne veľa bázových riešení. Čo by sa ale stalo, keby sme sa v kroku (8) rozhodli, že x nevystrieda v báze z ale s_2? Namiesto tabla (9) dostaneme tablo

$$\begin{array}{rrrrr}
f = & 9 & + s_1 & - \frac{1}{2}s_2 & - \frac{1}{2}s_3 \\
\hline
x = & 5 & + \frac{7}{3}s_1 & - \frac{5}{6}s_2 & - \frac{1}{2}s_3 \\
y = & 4 & + \frac{2}{3}s_1 & - \frac{1}{6}s_2 & - \frac{1}{2}s_3 \\
z = & & - 2s_1 & + \frac{1}{2}s_2 & + \frac{1}{2}s_3 \\
s_4 = & 4 & + 2s_1 & - \frac{1}{2}s_2 & - \frac{1}{2}s_3
\end{array} \qquad (11)$$



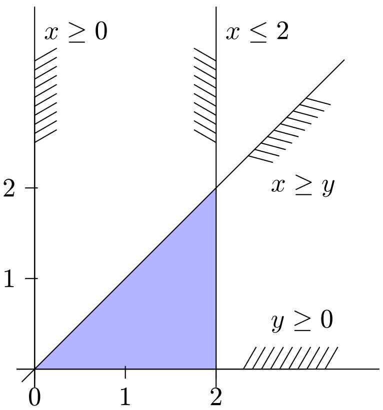
Tablo (9) reprezentuje bázu $\{x, y, s_2, s_4\}$ a tablo (11) bázu $\{x, y, z, s_4\}$, pričom obidvom bázam prislúcha rovnaké riešenie $(5, 4, 0, 0, 0, 0, 4)$. Zo zápisu (11) ale nevidno, že sme už našli optimum, a preto treba urobiť ešte jeden pivotný krok s pivotom s_1. Pri tomto kroku ale zistíme, že premenná z nedovolí zväčšiť s_1, a tak sme nútení urobiť krok "naprázdno" a vymeniť v báze s_1 za z. Dostaneme tablo

$$\begin{array}{rrrrr}
f = & 9 & - \frac{1}{2}z & - \frac{1}{4}s_2 & - \frac{1}{4}s_3 \\
\hline
x = & 5 & - \frac{7}{6}z & - \frac{1}{4}s_2 & + \frac{1}{12}s_3 \\
y = & 4 & - \frac{1}{3}z & & - \frac{1}{3}s_3 \\
s_1 = & & - \frac{1}{2}z & + \frac{1}{4}s_2 & + \frac{1}{4}s_3 \\
s_4 = & 4 & - z & &
\end{array} \qquad (12)$$



z ktorého už vidno optimalitu. Tento degenerovaný krok ale nezvýšil hodnotu f, čím rozbil náš pôvodný argument o zastavení: nevieme totiž zaručiť, že hodnota f sa v každom kroku zväčší. Nutnosť spraviť degenerovaný pivotný krok nemusí nastať iba na konci, keď už máme optimálne riešenie, ako ukazuje jednoduchý príklad (podľa [8]):

$$\begin{array}{rrrl} \text{maximalizovať} & & y & =: f(x,y) \\ \text{pri obmedzeniach} & -x & +y & \leq 0 \\ & x & & \leq 2 \\ & & x,y & \geq 0 \end{array}$$



Keď zavedieme rezervné premenné s_1, s_2, dostaneme tablo

$$\begin{array}{rrrr} f = & & & y \\ \hline s_1 = & & x & -y \\ s_2 = & 2 & -x & \end{array}$$

pre bázu $\{s_1, s_2\}$ s hodnotou riešenia $f = 0$. Jediná možnosť, ako pokračovať (a dostať sa k optimálnemu riešeniu s hodnotou 2), je spraviť degenerovaný pivotný krok, v ktorom y vystrieda v báze s_1:

$$\begin{array}{rrrr} f = & & x & -y \\ \hline y = & & x & -s_1 \\ s_2 = & 2 & -x & \end{array}$$

Podobná situácia je sa vyskytuje pomerne často.

Definícia 1.8. *Degenerovaný krok* simplexového algoritmu je taký pivotný krok, pri ktorom sa báza B transformuje na bázu B' s rovnakým bázovým riešením.

Degenerovaným krokom sa nevieme vyhnúť a navyše nasledovný príklad z [3] ukazuje, že ak nie sme dosť opatrní, môžme sa zacykliť. Uvažujme nasledovné tablo:

$$\begin{array}{rrrrrr} f = & & 10x_1 & -57x_2 & -9x_3 & -24x_4 \\ \hline x_5 = & & -0.5x_1 & +5.5x_2 & +2.5x_3 & -9x_4 \\ x_6 = & & -0.5x_1 & +1.5x_2 & +0.5x_3 & -x_4 \\ x_7 = & 1 & -x_1 & & & \end{array} \tag{13}$$

pre bázu $\{x_5, x_6, x_7\}$. Predpokladajme, že konkrétny simplexový algoritmus vždy vyberie ako pivota nebázovú premennú x_{μ_e} s maximálnou hodnotou r_e. V prípade, že pivotný krok vynuluje viacero bázových premenných, vyberie sa premenná s minimálnym indexom. Nechávame ako cvičenie pre čitateľa overiť, že algoritmus v nasledujúcich iteráciách prejde cez bázy $\{x_1, x_6, x_7\}$, $\{x_1, x_2, x_7\}$, $\{x_2, x_3, x_7\}$, $\{x_3, x_4, x_7\}$, $\{x_4, x_5, x_7\}$ a napokon sa dostane naspäť do $\{x_5, x_6, x_7\}$. Keďže vznikol cyklus z degenerovaných pivotných krokov, algoritmus sa zacyklí.

Vidíme, že nemôžeme dúfať, že dokážeme termináciu simplexovej metódy pre ľubovoľnú voľbu pivota, ale musíme zafixovať nejaký konkrétny algoritmus pre pivotný krok. Existuje veľa alternatívnych pravidiel na výber pivota, s rôznymi prístupmi k problému zacyklenia. My na dôkaz terminácie použijeme *pravidlo najmenšieho indexu* pôvodne z [2]

Definícia 1.9. *Pravidlo najmenšieho indexu* vyberie za pivota premennú x_{μ_e}, kde μ_e je najmenšie také, že $r_e > 0$. Ak pivotný krok vynuluje viacero bázových premenných, z bázy sa vyhodí premenná x_{β_ℓ} s najmenším indexom β_ℓ.

Veta 1.10. *Simplexový algoritmus, ktorý používa pravidlo najmenšieho indexu, vždy skončí a nájde optimálne riešenie.*

***Dôkaz:*.** Vieme, že ak algoritmus skončí, nájde optimálne riešenie. Najprv si uvedomíme, že jediný spôsob, ako algoritmus môže neskončiť je, že sa dostane do cyklu, ktorý pozostáva zo samých degenerovaných krokov. Vskutku, ak algoritmus neskončí, musí nekonečne veľakrát spracovávať nejakú bázu B. Keďže k B prislúcha práve jedno (prípustné) bázové riešenie, vždy, keď algoritmus spracováva B, je hodnota f rovnaká. Každý nedegenerovaný krok ale hodnotu f zväčší a degenerované kroky ju nemenia. Preto musí všetky kroky k ďalšiemu výskytu B musia byť degenerované. Na to, aby sme dokázali tvrdenie vety, nám teda stačí ukázať, že pri použití pravidla najmenšieho indexu nemôže nastať cyklus zo samých degenerovaných krokov.

Budeme postupovať sporom. Predpokladajme, že algoritmus má tablo s bázou B_0 a postupne vytvára tablá pre bázy $B_1, \dots, B_k = B_0$, pričom všetky pivotné kroky sú degenerované (t.j. všetky bázy $B_1, \dots, B_k$ majú rovnaké bázové riešenie). Premennú x_i nazveme *nestála*, ak sa vyskytuje v niektorej báze B_j, ale nevyskytuje v inej $B_{j'}$. Nech t je najväčšie také, že x_t je nestála. Keďže x_t je nestála, existuje pivotný krok, v ktorom x_t vypadne z bázy, t.j. pre nejaké j je $x_t \in B_j$ a $x_t \notin B_{j+1}$. Takže musí existovať nejaká iná (nestála) premenná x_s, ktorá x_t nahradí v báze: $x_s \notin B_j$ a $x_s \in B_{j+1}$. Zároveň, ak skúmame postupnosť báz $B_j, \dots, B_k, B_1, B_2, \dots B_{j+1}$, tak x_t sa zasa musí nejak do bázy vrátiť, t.j. musí existovať báza $B_{j^\star}$, že $x_t \notin B_{j^\star}$ a $x_t \in B_{j^\star+1}$. Nech $\mathcal{T}(B_j)$ vyzerá takto:

$$
\begin{array}{lclcl}
f & = & f_0 & + & \sum\limits_{k \notin B_j} r_k x_k \\
\hline
x_{\beta_1} & = & p_{\beta_1} & + & \sum\limits_{k \in B_j} q_{\beta_1,k} x_k \\
\vdots & & & \vdots & \\
x_{\beta_m} & = & p_{\beta_m} & + & \sum\limits_{k \in B_j} q_{\beta_m,k} x_k
\end{array}
\tag{14}
$$

kde $B_j = \{\beta_1, \dots, \beta_m\}$. Keďže predpokladáme, že všetky kroky cyklu sú degenerované, bázy B_j a $B_{j^\star}$ majú rovnaké bázové riešenie (t.j. hodnoty všetkých premenných aj f sú rovnaké). Preto môžme napísať

$$
f = f_0 + \sum_{k=1}^{n} r_k^\star x_k \tag{15}
$$

kde

$$
r_k^\star = \begin{cases} 0 & \text{ak } k \in B_{j^\star} \\ \text{koeficient } r \text{ pri } x_k \text{ v table } \mathcal{T}(B_{j^\star}) & \text{inak} \end{cases}
$$

Pretože $\mathcal{T}(B_{j^\star})$, a špeciálne rovnicu (15), sme dostali ekvivalentnými úpravami systému (14), všetky riešenia systému (14) spĺňajú (15). Vyrobme si teraz nejaké (nie bázové, ani prípustné) riešenie systému (14): zvoľme ľubovoľné y a položme

$$
x_i = \begin{cases} y & \text{ak } i = s \\ 0 & \text{ak } i \neq s \text{ a } i \notin B_j \\ p_i + q_{i,s} y & \text{ak } i \in B_j \end{cases}
$$

Ľahko vidno, že takto zvolené $\mathbf{x}$ je riešením systému (14)[5]. Keďže naše $\mathbf{x}$ spĺňa (14) aj (15), z vyjadrenia f v obidvoch dostaneme

$$f_0 + r_s y = f_0 + \sum_{k=1}^{n} r_k^\star x_k = f_0 + r_s^\star y + \sum_{k\in B_j} r_k^\star (p_k + q_{k,s} y)$$

a po úprave

$$\left(r_s - r_s^\star - \sum_{k\in B_j} r_k^\star q_{k,s} \right) y = \sum_{k\in B_j} r_k^\star b_k.$$

Tento vzťah platí pre každé y, a nakoľko pravá strana od y nezávisí, dostávame, že

$$r_s - r_s^\star - \sum_{k\in B_j} r_k^\star q_{k,s} = 0.$$

Pretože premenná x_s bola v báze B_j vybratá ako pivot, musí byť $r_s > 0$. V $B_{j^\star}$ bola ako pivot vybratá premenná x_t, a keďže $t > s$, musí byť $r_s^\star \leq 0$. Keďže $r_s - r_s^\star > 0$, musí existovať nejaké $z \in B_j$, pre ktoré

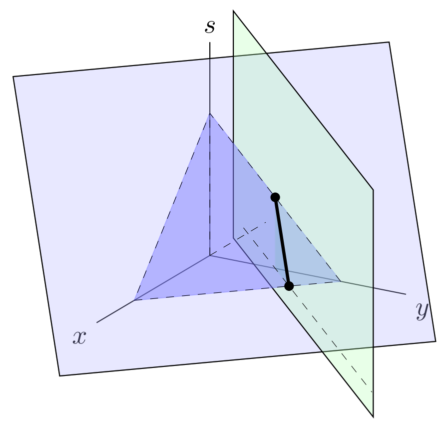
$$r_z^\star q_{z,s} > 0.$$

Premenná x_z je bázová v B_j, ale zároveň $r_z^\star \neq 0$, preto z definície $r^\star$ vyplýva, že $z \notin B_{j^\star}$; x_z je teda nestála premenná a z definície t platí $z \leq t$. Zároveň $z \neq t$: pretože x_t bolo vyhodené z bázy B_j pri pivotnom kroku, musí byť $q_{t,s} < 0$ a aj $r_t^\star q_{t,s} < 0$ (lebo x_t bol pivot pri $B_{j^\star}$). Teraz vieme, že $z < t$. Lenže x_z nebol v $B_{j^\star}$ pivot, preto musí byť $r_z^\star \leq 0$. Keďže $r_z^\star q_z, s > 0$, musí byť $q_{z,s} < 0$. Keďže všetky bázové riešenia v degenerovanom cykle sú rovnaké a $z \notin B_{j^\star}$, je $x_z = 0$ v bázovom riešení B_j aj $B_{j^\star}$. Pretože $z \in B_j$, musí byť $p_z = 0$. To ale znamená, že x_z sa dalo vyhodiť z bázy B_j, ale namiesto neho sa vyhodilo x_t, čo je v spore s pravidlom minimálneho indexu. $\square$

Ako začať

Posledný detail, ktorý potrebujeme vyriešiť, je otázka, ako simplexový algoritmus naštartovať. Doteraz sme totiž predpokladali, že začíname z bázy B_0, ktorá má prípustné bázové riešenie. V úvodnom príklade sme za štartovaciu bázu B_0 v (5) zvolili rezervné premenné $s_1, \ldots, s_4$. Tento prístup zjavne funguje pre programy tvaru $\max_{\mathbf{x}\in\mathbb{R}^7} \left\{ \mathbf{c}^\top \mathbf{x} \mid A\mathbf{x} \leq \mathbf{b},\ \mathbf{x} \geq 0 \right\}$ s pridanými rezervnými premennými, ak $\mathbf{b} \geq \mathbf{0}$. Čo ale s inými programami? Uvažujme nasledovný program:

$$\begin{array}{rlllll}
\text{maximalizovať} & 4x & & -\ z & =: f(x,y,z) \\
\text{pri obmedzeniach} & x & +\ y & +\ z & = 4 \\
 & x & -\ y & & = -2 \\
 & & & x,y,z & \geq 0
\end{array} \qquad (16)$$



Prípustné riešenia tvoria úsečku $(1,3,0) - (0,2,2)$. Na to, aby sme mohli spustiť simplexový algoritmus, potrebujeme nájsť nejaké prípustné riešenie. Pre bod (x,y,z) si označme $p_1 := 4-x-y-z$;

[5] Je to ako keby sme robili pivotný krok s premennou x_s o y, pričom sa nestaráme o to, aby premenné ostali nezáporné.

p_1 nám hovorí, ako veľmi je porušená prvá rovnosť[6]. Podobne nech $p_2 := 2 + x - y$ (všimnite si, že sme rovnicu upravili tak, aby absolútny člen bol nezáporný). Nájsť prípustné riešenie znamená nájsť taký bod (x, y, z), že $p_1 = p_2 = 0$, a teda $p_1, p_2 \geq 0$ a $p_1 + p_2 = 0$. Ľahko vidno, že program (16) má prípustné riešenie práve vtedy, ak program

$$\begin{array}{rlllllll} \text{maximalizovať} & -p_1 & -p_2 & & & & & \\ \text{pri obmedzeniach} & p_1 & & +x & +y & +z & =4 & \\ & & p_2 & -x & +y & & =2 & \\ & & & & x,y,z,p_1,p_2 & & \geq 0 & \end{array} \tag{17}$$

má riešenie s hodnotou 0. V tomto programe ľahko vidno, že $\{p_1, p_2\}$ je báza prípustného riešenia. Môžme teda použiť simplexový algoritmus na nájdenie optimálneho riešenia a toto použiť ako počiatočné prípustné riešenie pôvodného programu.
Tento postup môžme uplatniť vždy. Majme lineárny program v normálnom tvare

$$\max_{\mathbf{x} \in \mathbb{R}^n} \left\{ \mathbf{c}^\mathsf{T} \mathbf{x} \mid A\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0} \right\}$$

Najprv zabezpečíme, aby $\mathbf{b} \geq 0$: ak $b_i < 0$ pre nejaké i, tak príslušnú rovnicu prenásobíme -1. Zavedieme nové premenné $x_{n_1}, \ldots, x_{n+m}$ a zostavíme pomocný program

$$\max_{\tilde{\mathbf{x}} \in \mathbb{R}^{n+m}} \left\{ -x_{n+1} - \ldots - x_{n+m} \mid \tilde{A}\tilde{\mathbf{x}} = \mathbf{b},\ \tilde{\mathbf{x}} \geq \mathbf{0} \right\}$$

kde $\tilde{A} = (A \mid I_m)$ dostaneme z A pripojením identickej matice rozmerov $m \times m$. Pretože $\mathbf{b} \geq 0$, $\{x_{n+1}, \ldots, x_{n+m}\}$ tvoria bázu prípustného riešenia a môžme použiť simplexový algoritmus na získanie optimálneho riešenia. Ak je optimálne riešenie 0, máme prípustné riešenie pôvodného programu. Naopak, pre každé prípustné riešenie pôvodného programu existuje riešenie pomocného programu s hodnotou 0, takže ak je optimum pomocného programu záporné, vieme, že pôvodný program nemal žiadne prípustné riešenie.

Cvičenie. Naprogramujte simplexový algoritmus s pravidlom najmenšieho indexu.

1.3 Zložitosť simplexového algoritmu

V predchádzajúcej kapitole sme sa si priblížili simplexovú metódu, ktorá umožňuje riešiť úlohy lineárneho programovania efektívnejšie ako prehľadávaním všetkých vrcholov telesa prípustných riešení. Ukázali sme, že simplexová metóda s pravidlom najmenšieho indexu vždy skončí. Otázkou teraz je, či je naozaj efektívna. Odpoveď je prekvapivá. Napriek tomu, že v praxi sa simplexový algoritmus ukazuje ako veľmi rýchly, jeho zložitosť v najhoršom prípade je exponenciálna, ako o chvíľu ukážeme.
Najprv je však namieste zopakovať niekoľko základných faktov, keďže v tomto prípade záleží na subtílnych detailoch. Keď analyzujeme zložitosť nejakého algoritmu, robíme tak v závislosti od parametra, ktorý je, v princípe, súčasťou definície problému. Keď napríklad povieme, že nejaký algoritmus má v najhoršom prípade zložitosť $O(n^2)$, myslíme tým, že existuje konštanta c a n_0 taká, že pre ľubovoľný vstup, ktorého parameter $n > n_0$, je čas algoritmu $\leq cn^2$. Prirodzeným parametrom, ktorý sa dá použiť univerzálne, je dĺžka vstupu: súčasťou definície problému je vždy aj spôsob kódovania stupu do reťazca a počet bitov, potrebných na zápis daného vstupného reťazca je dobrý zložitostný parameter. Niekedy (a vlastne dosť často) sa ale používajú iné parametre, ktoré sú pre daný problém prirodzenejšie: keď sa napríklad analyzuje zložitosť triedenia postupnosti prirodzených čísel, je zväčša parametrom počet triedených čísel n, aj keď dĺžka vstupu závisí od veľkosti triedených čísel. Podobným príkladom sú grafové algoritmy, ktoré sa niekedy analyzujú vzhľadom na počet vrcholov, aj keď na zápis n-vrcholových grafov je treba vo všeobecnosti až $\Omega(n^2)$ bitov[7].

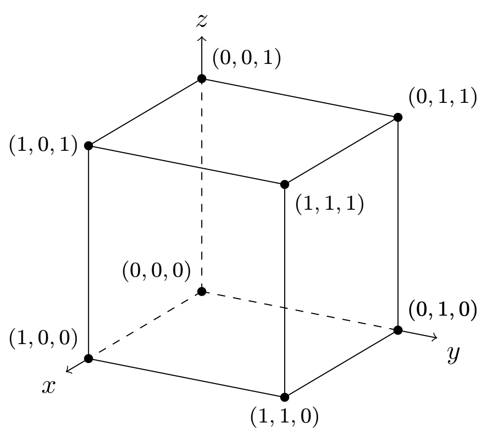
[6] *nie je* to vzdialenosť bodu (x, y, z) od roviny $x + y + z = 4$

[7] Stačí si uvedomiť, že v očíslovanom grafe je $\binom{n}{2}$ potenciálnych hrán a každá v ňom môže byť alebo nebyť prítomná, t.j. je $2^{\binom{n}{2}}$ grafov a na identifikáciu každého z nich je z Dirichletovho princípu treba aspoň $\binom{n}{2}$ bitov.

Vstupom lineárneho programu s n premennými a m obmedzeniami sú dva vektory $\mathbf{c}$ a $\mathbf{b}$ reálnych čísel a matica $A \in \mathbb{R}^{m \times n}$. Prirodzenými parametrami sú teda m, n a dĺžka vstupu, pričom v poslednom prípade treba brať do úvahy aj spôsob kódovania reálnych čísel a zmieriť sa s faktom, že ak chceme mať konečné vstupy, tak nemôžeme zapísať všetky reálne čísla.
Tieto aspekty je dobré mať na pamäti, aj keď nás momentálne nemusia príliš trápiť: zostrojíme vstup s n premennými a $2n$ obmedzeniami, na ktorom je čas simplexového algoritmu $\Omega(2^n)$. Navyše pri tom použijeme čísla s krátkym zápisom (stačia nám čísla $\{\pm 1, \pm\frac{1}{4}, 0\}$), takže ukážeme, že algoritmus je exponenciálny od hociktorého zo spomenutých parametrov.
Budeme uvažovať simplexový algoritmus, ktorý používa pravidlo najmenšieho indexu (pre veľa iných pravidiel existujú podobné kontrapríklady) a pre každé n skonštruujeme vstup s n premennými a $2n$ obmedzeniami tak, že teleso prípustných riešení má 2^n vrcholov a simplexový algoritmus ich všetky prehľadá. V nami konštruovanom zadaní bude cieľom maximalizovať x_n a obmedzenia budú tvoriť "pokrivenú" kocku. Začnime s tým, že pomocou $2n$ obmedzení vyrobíme n-rozmernú kocku:

$$\begin{array}{rcl} 0 \leq & x_1 & \leq 1 \\ 0 \leq & x_2 & \leq 1 \\ & \cdots & \\ 0 \leq & x_n & \leq 1 \end{array}$$

V troch rozmeroch teleso prípustných riešení je kocka:



Našim cieľom bude posunúť vrcholy kocky tak, aby vznikla dlhá rastúca "špirála". Zvoľme si nejaké $\varepsilon < \frac{1}{2}$ a definujme obmedzenia

$$\begin{array}{rcl} \varepsilon \leq & x_1 & \leq 1 \\ \varepsilon x_1 \leq & x_2 & \leq 1 - \varepsilon x_1 \\ & \cdots & \\ \varepsilon x_{n-1} \leq & x_n & \leq 1 - \varepsilon x_{n-1} \end{array}$$

Program prepíšeme do normálneho tvaru tak, že zavedieme rezervné premenné r_i, s_i a obmedzenia budú mať formu rovností. Dostávame program:

$$\begin{array}{lrl} \text{maximalizovať} & x_n & \\ \text{pri obmedzeniach} & x_1 - r_1 & = \varepsilon \\ & x_1 + s_1 & = 1 \\ & x_2 - \varepsilon x_1 - r_2 & = 0 \\ & x_2 + \varepsilon x_1 + s_2 & = 1 \\ & \cdots & \cdots \\ & x_n - \varepsilon x_{n-1} - r_n & = 0 \\ & x_n + \varepsilon x_{n-1} + s_n & = 1 \end{array} \tag{18}$$

kde všetky premenné sú nezáporné. Ako vyzerajú bázové riešenia? Kvôli pridaným premenným sú jednotlivé obmedzenia nezávislé (každé obmedzenie obsahuje jednu premennú, ktorá sa nevyskytuje nikde inde), preto báza má $2n$ prvkov. Zároveň vidno, že $r_1 + s_1 = 1 - \varepsilon$ a pre každú dvojicu premenných r_i, s_i, kde $i > 1$, platí $r_i + s_i = 1 - 2\varepsilon x_{i-1} > 0$. Preto nemôže platiť $r_i = s_i = 0$, a teda každá báza musí obsahovať aspoň jednu z premenných r_i, s_i. Navyše všetky $x_i > 0$ a teda sú v každej báze. Každá báza B sa preto dá jednoznačne charakterizovať množinou $R_B \subseteq \{1, \dots, d\}$: premenné bázy B sú potom

$$\{x_1, \dots, x_n\} \cup \bigcup_{i \in R_B} \{r_i\} \cup \bigcup_{i \notin R_B} \{s_i\}$$

Zároveň je zrejmé nasledovné tvrdenie:

Tvrdenie 1.11. Každý pivotný krok je jednoznačne charakterizovaný indexom i, pričom vymení príslušnosť do bázy pre premenné r_i a s_i.

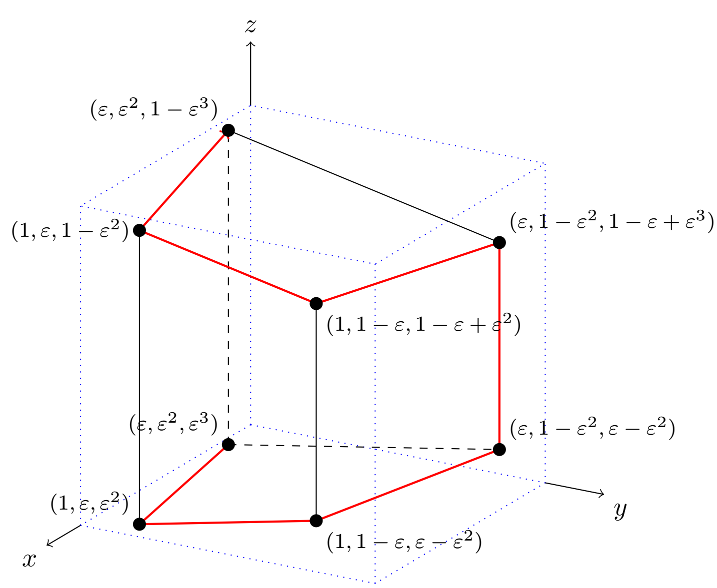
Pre ilustráciu, nech $n = 3$. V maticovom zápise máme program

$$\max\{x_3 \mid A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq 0\}$$

kde

$$A = \begin{pmatrix} 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -\varepsilon & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ \varepsilon & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & -\varepsilon & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & \varepsilon & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \qquad \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ r_1 \\ s_1 \\ r_2 \\ s_2 \\ r_3 \\ s_3 \end{pmatrix} \qquad \mathbf{b} = \begin{pmatrix} \varepsilon \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}$$

Matica A má hodnosť 6 a $R_B \subseteq \{1, 2, 3\}$. Máme teda 8 bázových riešení, ktoré tvoria pokrivenú kocku:



Červeným je vyznačená rastúca cesta, ktorá začína v riešení s množinou $R_{B_0} = \emptyset$ a prejde všetky vrcholy, pričom vždy sa posunie po prvej možnej dimenzii, v ktorej účelová funkcia rastie. Aby sme mohli tento príklad zovšeobecniť na n dimenzií, potrebujeme vedieť argumentovať o pivotných krokoch algoritmu s pravidlom najmenšieho indexu. K tomu nám pomôže tvrdenie 1.11 a nasledovná lema:

Lema 1.12. *Majme bázu B programu (18) a k nej tablo $\mathcal{T}(B)$. Nech je účelová funkcia v $\mathcal{T}(B)$ vyjadrená pomocou nebázových premenných ako $x_n = c_0 + c_1v_1 + c_2v_2 + \cdots + c_nv_n$, kde v_i je r_i alebo s_i a c_i je koeficient. Potom c_i je kladný práve vtedy, ak počet bázových premenných r_j pre $j \geq i$ je párny, t.j.*

$$|\{j \mid j \in R_B,\ j \geq i\}| \equiv 0 \pmod 2$$

Dôkaz:. Dôkaz urobíme indukciou na rozmer problému n. Pre $n = 1$, ak r_1 je v báze, máme $x_1 = 1 - s_1$ a c_1 je záporný, ak r_1 nie je v báze, máme $x_1 = \varepsilon + r_1$ a c_1 je kladný.
Nech tvrdenie platí pre $n - 1$. Ak $n \in R_B$, tak v zápise x_n musí figurovať s_n a teda musí byť tvaru $x_n = 1 - s_n - \varepsilon(c'_0 + c'_1v'_1 + \cdots + c'_{n-1}v'_{n-1})$, kde $x_{n-1} = c'_0 + c'_1v'_1 + \cdots + c'_{n-1}v'_{n-1}$ je zápis x_{n-1} pomocou nebázových premenných $v_1, \ldots, v_{n-1}$. Roznásobením a použitím indukčného predpokladu dostaneme výsledok. Ak $n \notin R_B$, postup je analogický s použitím vzťahu $x_n = r_n + \varepsilon x_{n-1}$. □

Teraz môžeme ukázať, že simplexový algoritmus navštívi všetky vrcholy:

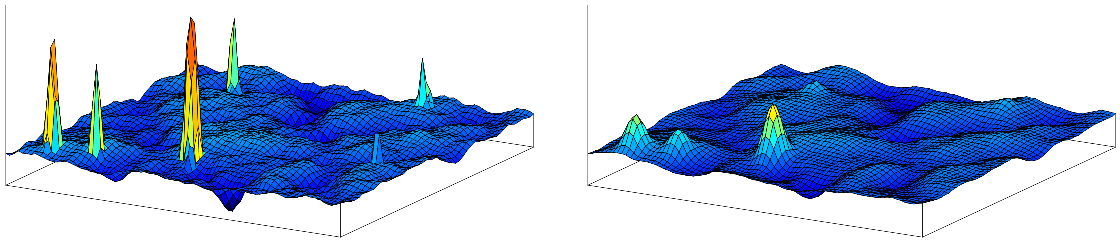
Veta 1.13. *Nech $i \in \{1, \ldots, n\}$ a $R \subseteq \{i + 1, \ldots, n\}$. Ak simplexový algoritmus, ktorý používa pravidlo najmenšieho indexu, začína z bázy B_0, kde $R_{B_0} = R$ (resp. $R_{B_0} = \{i\} \cup R$) a $|R|$ je párne (resp. $|R|$ je nepárne), tak prejde cez všetky bázy tvaru $R' \cup R$ kde $R' \subseteq \{1, \ldots, i\}$ a skončí v báze B_1, kde $R_{B_1} = \{i\} \cup R$ (resp. $R_{B_1} = R$).*

Dôkaz:. Indukciou na i. Ak $i = 1$, potom v obidvoch prípadoch ($R_{B_0} = R$, $|R|$ je párne, aj $R_{B_0} = \{1\} \cup R$, $|R|$ je nepárne) je podľa lemy 1.12 koeficient pri v_1 kladný a algoritmus urobí pivotný krok s indexom 1.
Nech teraz tvrdenie platí pre $i - 1$. Máme dva prípady. Nech najprv $R_{B_0} = R$ a $|R|$ je párne. Keďže $R \subseteq \{i, \ldots, n\}$, môzme použiť indukčný predpoklad: algoritmus prejde všetky bázy tvaru $R' \cup R$, kde $R' \subseteq \{1, \ldots, i - 1\}$ a skončí v $\{i - 1\} \cup R$. Pretože $|R|$ je párne, podľa lemy 1.12 algoritmus prejde to bázy $\{i - 1, i\} \cup R$. Použitím indukčného predpokladu pre $i - 1$ a nepárnu množinu $\{i\} \cup R$ dostávame výsledok.
Druhý prípad, keď $R_{B_0} = \{i\} \cup R$ a $|R|$ je nepárne je analogický a prenechávame ho na čitateľa. □

Dôsledok 1.14. *Simplexový algoritmus s pravidlom najmenšieho indexu urobí exponenciálne veľa iterácií na programe (18).*

Vidíme teda, že simplexový algoritmus je v najhoršom prípade exponenciálny, nech už za parameter zoberieme počet premenných, počet obmedzení, alebo dĺžku vstupu. Ako si ale vysvetliť, že v praxi funguje ozaj dobre? Možným smerom by bolo analyzovať priemerný prípad. Hneď ale narážame na problém, ako priemerný prípad definovať. Vskutku, existujú výsledky, ktoré hovoria, že simplexový algoritmus urobí v "priemernom" prípade polynomiálny počet krokov, kde "priemerný prípad" znamená očakávanú hodnotu, ak matica A aj vektory **c**, **b** sú vybrané náhodne z daného pravdepodobnostného rozdelenia. Toto ale stále nie je zďaleka uspokojivá odpoveď: priemerný prípad je totiž veľmi ďaleko od "typického", v praxi sa vyskytujúceho, prípadu; program, ktorého matica by bola náhodná by bol v skutočnosti veľmi podivná výnimka. Vysvetlenie priniesol pojem *vyhladenej zložitosti*[8], ktorý je kombináciou najhoršieho a priemerného prípadu: uvažujeme najhoršiu možnú inštanciu, ale pre každú inštanciu neuvažujeme iba čas potrebný na jej vyriešenie, ale priemerný čas potrebný na vyriešenie inštancií z jej blízkeho okolia. Okolie inštancie dostaneme tak, že každé číslo, vyskytujúce sa vo vstupe, posunieme o malú náhodnú hodnotu. Spielman a Teng [9] ukázali, že vyhladená zložitosť simplexového algoritmu je pre každú inštanciu polynomiálna. Ak si intuitívne predstavíme priestor všetkých vstupov ako rovinu, zložitosť simplexového algoritmu je ako na obrázku vľavo: väčšinou je polynomiálna a má iba riedko rozmiestnené jednotlivé "zlé" inštancie.

[8] *smoothed complexity*



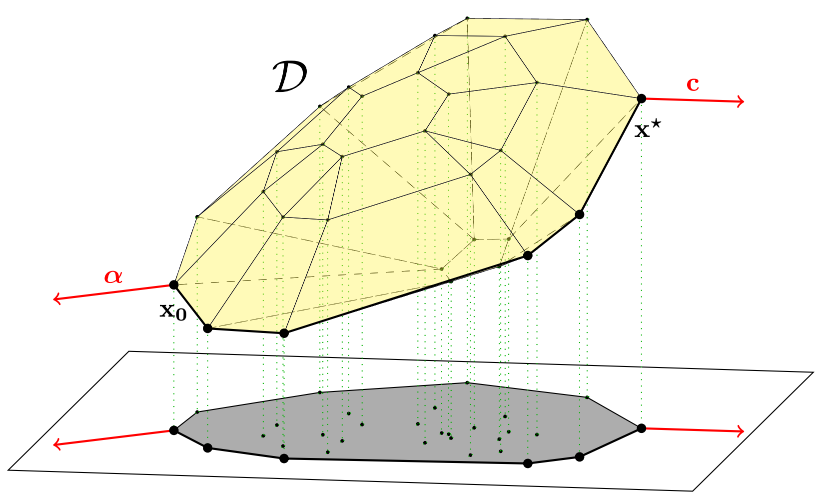
Po spriemerovaní cez malé okolie sa zlé inštancie vyhladia ako na obrázku vpravo. Toto vysvetľuje, prečo sa simplexová metóda dá považovať za efektívnu metódu riešenia lineárnych programov: existujú síce exponenciálne zlé vstupy, ale na to, aby sme na taký natrafili, musia byť všetky vstupné čísla nastavené veľmi presne – stačí, ak vstupné hodnoty obsahujú malý náhodný šum a v očakávanom prípade je každá inštancia polynomiálna.

Výsledok Spielmana a Tenga sa považuje za prelomový a na prvý (aj na druhý a tretí) pohľad môže vyzerať úplne nepochopiteľne. Nie je v možnostiach tohoto textu ukázať kompletný dôkaz, ktorý je pomerne náročný, ničmenej na záver tejto kapitoly by sme chceli ukázať aspoň jednoduchú vizuálnu predstavu, ktorá by zvedavému čitateľovi naznačila, že nejde o žiadnu mágiu.

V predchádzajúcom texte sme predstavili simplexový algoritmus s pravidlom najmenšieho indexu, ale pre teraz sa pre naše účely bude viac hodiť iné pravidlo, ktoré sa volá *pravidlo sledovania tieňa*. Majme lineárny program v tvare

$$\max_{\mathbf{x}\in\mathbb{R}^n}\{\mathbf{c}^{\mathsf{T}}\mathbf{x} \mid A\mathbf{x} \le \mathbf{b}, \mathbf{x} \ge 0\}$$

Prípustné riešenia tvoria mnohosten $\mathcal{D}$ v n-rozmernom priestore a nájsť optimálne bázové riešenie znamená nájsť vrchol $\mathcal{D}$, ktorý je najďalej v smere vektora $\mathbf{c}$. Keďže vieme, že v dvoch rozmeroch je tento problém ľahký, môžme urobiť nasledovnú úvahu: zoberme si štartovacie bázové riešenie $\mathbf{x_0}$; keďže je to vrchol $\mathcal{D}$, existuje vektor $\boldsymbol{\alpha}$, že vrchol $\mathbf{x_0}$ maximalizuje hodnotu $\boldsymbol{\alpha}^{\mathsf{T}}\mathbf{x}$ ($\mathbf{x_0}$ je najďalej v smere $\boldsymbol{\alpha}$). Zoberme si rovinu danú vektormi $\boldsymbol{\alpha}$ a $\mathbf{c}$ a premietnime každý vrchol $\mathcal{D}$ do nej – dostaneme množinu bodov v rovine a ich konexný obal bude *tieň*, ktorý $\mathcal{D}$ vrhá do roviny.



Nie je ťažké vidieť, že $\mathbf{x_0}$ aj optimálne riešenie $\mathbf{x}^\star$ ležia na hranici tieňa. Simplexový algoritmus s pravidlom sledovania teiňa postupuje v princípe takto: keď sa v nejakom bázovom riešení treba

rozhodnúť, ako zmeniť bázu, vyberie sa bázové riešenie, ktoré leží na hranici tieňa (to sa dá otestovať napr. tak, že zostrojíme všetky susedné bázové riešenia a porovnáme, kde ležia ich priemety). Počet iterácií simplexového algoritmu s týmto pravidlom je zjavne nanajvýš počet bodov na hranici tieňa. Dôležitý medzikrok v dôkaze je, že sa riešený program prevedie do tvaru, kde obmedzenia majú tvar $\mathbf{a_i}^\mathsf{T}\mathbf{x} \leq 1$; v tomto prípade sa dá ukázať, že počet vrcholov tieňa je nanajvýš počet vrcholov mnohouholníka $\mathcal{M}$, ktorý dostaneme ako prienik konvexného obalu bodov $\mathbf{a_1}, \dots, \mathbf{a_n}$ a roviny definovanej vektormi $\boldsymbol{\alpha}$, $\mathbf{c}$. Jadrom dôkazu je geometrické tvrdenie: ak máme n bodov v d-rozmernom priestore, každý z nich posunieme o náhodný vektor s normálnym rozdelením a disperziou σ^2, tak v očakávanom prípade sú uhly medzi susednými úsečkami $\mathcal{M}$ dosť ostré, a preto $\mathcal{M}$ nemôže mať príliš veľa vrcholov, konkrétne nanajvýš nejaký polynóm $poly(n, d, \frac{1}{\sigma})$.
Od týchto letmých úvah je k dôkazu ešte veľmi dlhá cesta; našim cieľom však bolo iba naznačiť smer, akým sa dôkaz tohto typu môže uberať. Záujemcov o detaily odkazujeme na články [Spielman,Teng] a prednášky `http://www.cs.yale.edu/homes/spielman/BAP/`