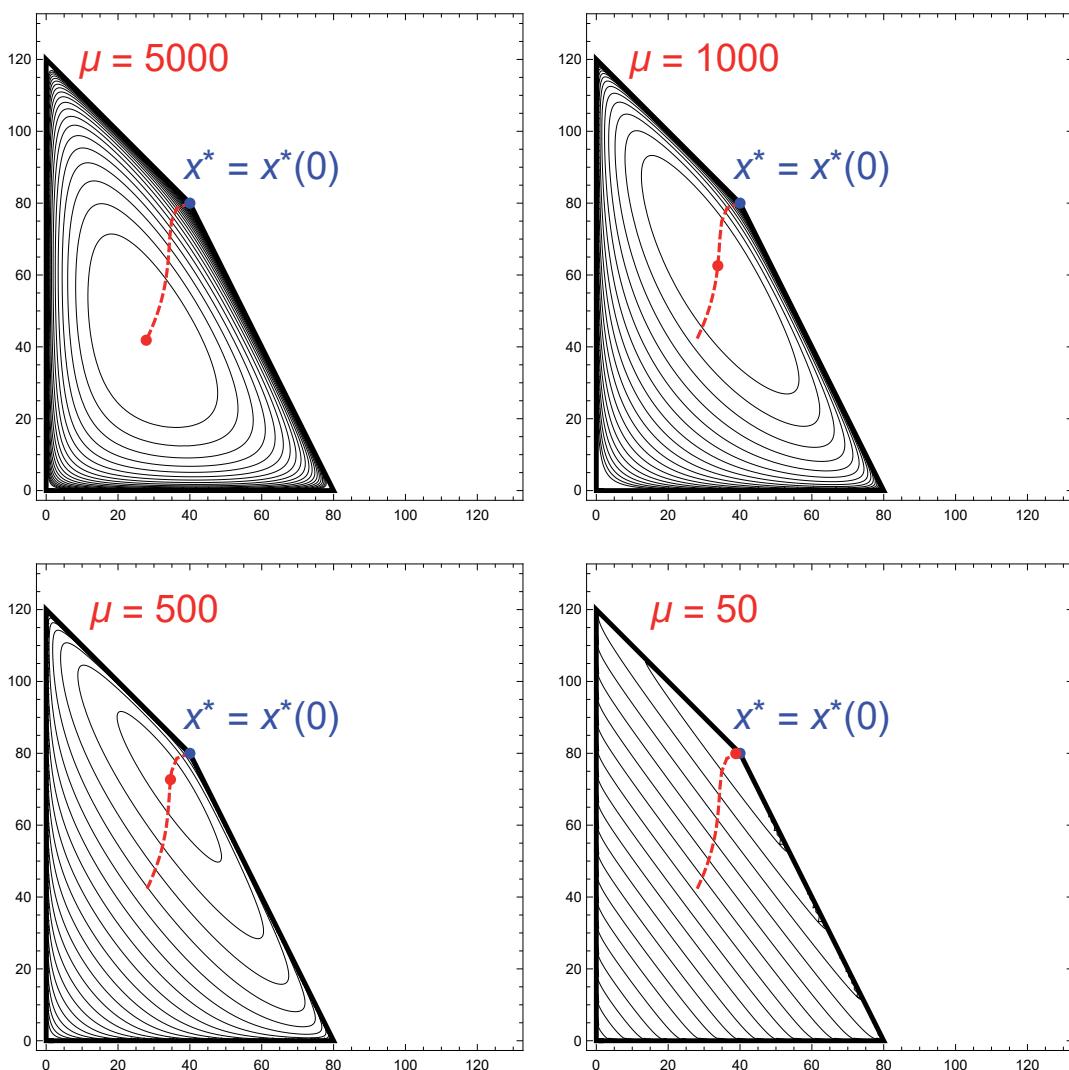


OPTIMALIZÁCIA 2

Martin Knor, Lukáš Tomek



OPTIMALIZÁCIA 2

Martin Knor, Lukáš Tomek

Všetky práva vyhradené. Nijaká časť textu nesmie byť použitá na ďalšie šírenie akoukoľvek formou bez predchádzajúceho súhlasu autorov alebo vydavateľstva.

© prof. RNDr. Martin Knor, PhD., Ing. Mgr. Lukáš Tomek, PhD.

Recenzenti: RNDr. Igor Fabrici, Dr. rer. nat.
doc. RNDr. Róbert Jajcay, DrSc.
doc. Mgr. Petr Kolman, Ph.D.

Schválila Edičná rada Stavebnej fakulty STU v Bratislave.

ISBN 978-80-227-4918-3

Predhovor

Tento učebný text je určený študentom štvrtého ročníka stavebnej fakulty Slovenskej technickej univerzity študujúcim v študijnom programe matematicko-počítačové modelovanie. Predstavuje spisané a v niektorých častiach mierne rozšírené prednášky z predmetu optimalizácia 2.

Táto publikácia voľne nadväzuje na skriptá Lineárna a nelineárna optimalizácia [10], ktoré sú základným textom pre predmet optimalizácia. Tento predmet sa v súčasnosti vyučuje v druhom ročníku v študijnom programe matematicko-počítačové modelovanie. Znamená to, že predpokladáme základné vedomosti z matematickej optimalizácie. Napriek tomu sa pri definíciiach neodvolávame na [10], ale všetky potrebné pojmy zopakujeme.

Štruktúra učebnice je upravená tak, že každá kapitola tvorí jednu prednášku. Čitateľovi predkladáme 11 kapitol. Po obsahovej stránke možno učebnicu rozdeliť na tri celky. V kapitolách 1 až 3 sa zaobráme lineárnym programovaním. Zopakujeme si simplexový algoritmus a opíšeme štyri algoritmy založené na metóde vnútorného bodu. Kapitoly 4 až 9 sú venované matematickému (zväčša konvexnému) programovaniu. Zavedieme Karushove-Kuhnove-Tuckerove podmienky, dualitu a ukážeme si niekoľko algoritmov. Okrem algoritmov využívajúcich Newtonovu metódu si objasníme metódu projekcie gradientu. Posledné dve kapitoly sú venované stochastickým algoritmom. Opíšeme metódu Monte-Carlo, Horolezecký algoritmus, Simulované žíhanie a Genetické algoritmy.

Z uvedeného vyplýva, že učebnica obsahuje viaceré zaujímavé algoritmy. Tieto algoritmy sa na prednáške preberajú teoreticky a na cvičeniach ich študenti programujú. Využíva sa najmä softvér Wolfram Mathematica a na jednom cvičení aj softvér Matlab. Na cvičeniach sa predpokladá znalosť softvéru Mathematica v rozsahu príslušných bakalárskych predmetov. Znalosť softvéru Matlab sa nepredpokladá. K cvičeniam existujú predpripravené súbory a časti programov, ktoré sa nachádzajú na adrese „http://bit.ly/Optimalizacia2_cvicenia“.

Záverom tohto úvodu chceme podakovať recenzentom RNDr. Igorovi Fabrimu, Dr. rer. nat., doc. RNDr. Róbertovi Jajcayovi, DrSc. a doc. Mgr. Petrovi Kolmanovi, Ph.D., za cenné pripomienky. Tiež chceme podakovať Mgr. Gabriele Kubičkovej za jazykovú úpravu.

Autori

1 Podmienky komplementarity

V úvodnej kapitole si zopakujeme pojmy a základné tvrdenia z úvodu do matematickej a lineárnej optimalizácie, pozri [10]. Okrem toho zavedieme podmienky komplementarity, ktoré budeme v ďalšom využívať.

Kanonický tvar úlohy lineárneho programovania

DEFINÍCIA. Matematické programovanie je optimalizačná úloha s ohraničeniami v tvare rovníc, respektíve neostrých nerovníc. Ide teda o úlohu

$$\begin{array}{c} \text{opt } z = f(\mathbf{x}) \\ \hline g_1(\mathbf{x}) \square_1 0 \\ \vdots \\ g_m(\mathbf{x}) \square_m 0 \end{array}$$

kde $\mathbf{x} = (x_1, \dots, x_n)^T \in \mathbb{R}^n$, $\text{opt} \in \{\min, \max\}$ a $\square_i \in \{\leq, =, \geq\}$ pre každé i , pre ktoré $1 \leq i \leq m$, pričom f, g_1, \dots, g_m sú funkcie a f nazývame **účelová funkcia**.

V ďalšom n rezervujeme na počet premenných v zadaní úlohy, m bude obyčajne označovať počet ohraničení.

DEFINÍCIA. Bod \mathbf{x} je **prípustné riešenie** úlohy matematického programovania, ak vyhovuje všetkým ohraničeniam (včítane možných ohraničení na nezápornosť premenných, pozri nižšie). Bod \mathbf{x} je **optimálne riešenie** úlohy matematického programovania, ak je riešením tejto úlohy, čiže ak je prípustným riešením s optimálnou hodnotou účelovej funkcie.

Špeciálnym typom matematického programovania je lineárne programovanie.

DEFINÍCIA. Lineárne programovanie je optimalizačná úloha s ohraničeniami v tvare rovníc, respektíve neostrých nerovníc, v ktorej sú všetky funkcie lineárne. Úlohu lineárneho programovania môžeme schematicky zapísť

$$\begin{array}{c} \text{LP: } \text{opt } z = \mathbf{c}^T \cdot \mathbf{x} \\ \hline \mathbb{A} \cdot \mathbf{x} \square \mathbf{b} \\ \hline x_i \geq 0 \text{ pre } i \in I \subseteq \{1, \dots, n\} \end{array}$$

kde $\mathbf{x} = (x_1, \dots, x_n)^T \in \mathbb{R}^n$, $\text{opt} \in \{\min, \max\}$, \mathbb{A} je matica reálnych koeficientov typu $(m \times n)$, $\mathbf{c} = (c_1, \dots, c_n)^T \in \mathbb{R}^n$ a $\mathbf{b} = (b_1, \dots, b_m)^T \in \mathbb{R}^m$. Symbol \square si možno predstaviť ako stĺpec m symbolov z množiny $\{\leq, =, \geq\}$.

V predchádzajúcim zápise sme ohraničenia na nezápornosť premenných, tak ako je to zvykom, uviedli zvlášť. Vždy keď budeme hovoriť o ohraničeniach (bez prílastku), nebudeme mať na mysli ohraničenia na nezápornosť premenných.

Nech sú $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_m$ riadky matice \mathbb{A} úlohy LP. Potom ohraničenia možno zapísť

$$\mathbf{a}_i \cdot \mathbf{x} \square_i b_i \quad 1 \leq i \leq m,$$

kde $\square_i \in \{\leq, =, \geq\}$. Rovnicou $\mathbf{a}_i \cdot \mathbf{x} = b_i$ je určený $(n-1)$ -rozmerný affinný podpriestor n -rozmerného priestoru, čiže nadrovina. Nerovnicami $\mathbf{a}_i \cdot \mathbf{x} \leq b_i$ a $\mathbf{a}_i \cdot \mathbf{x} \geq b_i$ sú teda určené n -rozmerné polpriestory s hraničnou nadrovinou $\mathbf{a}_i \cdot \mathbf{x} = b_i$. Keď uvážime, že pre uhol α , ktorý zvierajú vektory \mathbf{a}_i a \mathbf{x} platí

$$\cos \alpha = \frac{\mathbf{a}_i \cdot \mathbf{x}}{\|\mathbf{a}_i\| \cdot \|\mathbf{x}\|},$$

tak rovniciu

$$\mathbf{a}_i \cdot \mathbf{x} = b_i,$$

čiže

$$\|\mathbf{x}\| \cdot \cos \alpha = \frac{b_i}{\|\mathbf{a}_i\|}$$

splňajú tie vektory \mathbf{x} , ktorých kolmý priemet do \mathbf{a}_i (čiže odvesna pravouhlého trojuholníka s preponou \mathbf{x} a uhlom α) má dĺžku $\frac{b_i}{\|\mathbf{a}_i\|}$. Z toho plynie, že \mathbf{a}_i je vektor kolmý na nadrovinu určenú rovnicou $\mathbf{a}_i \cdot \mathbf{x} = b_i$. Všimnime si tiež, že pre funkciu $g_i(\mathbf{x}) = \mathbf{a}_i \cdot \mathbf{x} - b_i$ je \mathbf{a}_i jej gradient. Teda platí $\nabla g_i(\mathbf{x}) = \mathbf{a}_i$ (presnejšie, $\nabla g_i(\mathbf{x}) = \mathbf{a}_i^T$, pretože \mathbf{a}_i je riadkový vektor a gradient uvažujeme ako stĺpcový vektor).

V úvode do matematickej a lineárnej optimalizácie [10] sme si ukázali, že každá úloha lineárneho programovania sa dá prepísať na ekvivalentnú úlohu v kanonickom tvare.

DEFINÍCIA. **Úloha lineárneho programovania v kanonickom tvaru** je maximalizačná úloha lineárneho programovania, ktorá má iba ohraničenia tvaru rovníc, a v ktorej sú všetky premenné, ako aj pravá strana, nezáporné. Jedná sa teda o úlohu

$$\begin{array}{c} \max z = \mathbf{c}^T \cdot \mathbf{x} \\ \text{KT:} \quad \begin{array}{c} \hline \mathbb{A} \cdot \mathbf{x} = \mathbf{b} \\ \hline \mathbf{x} \geq \mathbf{0} \end{array} \end{array}$$

kde $\mathbf{x}, \mathbf{c}, \mathbf{b}$ a \mathbb{A} sú ako v definícii úlohy lineárneho programovania.

Poznamenajme, že nezápornosť pravej strany tu nebudeme využívať a prenásobením účelovej funkcie hodnotou -1 vieme úlohu previesť na minimalizačnú.

Simplexový algoritmus

Úlohu lineárneho programovania sme riešili **simplexovým algoritmom**, pozri [10]. To je algoritmus, ktorý je definovaný pre úlohu lineárneho programovania v kanonickom tvaru. Simplexový algoritmus si zopakujeme na nasledujúcom príklade.

PRÍKLAD. Vyriešte úlohu lineárneho programovania

$$\begin{array}{r} \max z = 2x_1 + 3x_2 \\ \hline x_1 + 2x_2 \leq 10 \\ x_1 + x_2 \leq 6 \\ x_1 \leq 4 \\ \hline x_1, x_2 \geq 0 \end{array}$$

RIEŠENIE. Najprv prevedieme zadaný problém na úlohu lineárneho programovania v kanonickom tvaru, čiže pridáme nové premenné $v_1, v_2, v_3 \geq 0$.

$$\begin{array}{r} \max z = 2x_1 + 3x_2 \\ \hline x_1 + 2x_2 + v_1 = 10 \\ x_1 + x_2 + v_2 = 6 \\ x_1 + v_3 = 4 \\ \hline x_1, x_2, v_1, v_2, v_3 \geq 0 \end{array}$$

Teraz zapíšeme úvodnú simplexovú tabuľku. V záhlaví tejto tabuľky budú zořadené všetky premenné. Do riadkov v strednej časti tabuľky zapíšeme koeficienty rovníc, a to tak, že každý koeficient bude v stĺpci tej premennej, pri ktorej stojí. Na pravej strane za čiarou budú pravé strany príslušných rovníc. Na ľavú stranu zapíšeme premenné, ktoré sú eliminované, čiže v im zodpovedajúcim stĺpcu je len jeden koeficient nenulový a tento má hodnotu 1 (premennú zapíšeme do toho riadku, v ktorom je hodnota 1). Premenné na ľavej strane nazývame **bázické**, pretože tvoria **bázu** súčasného riešenia. Do spodného riadku pod čiaru zapíšeme koeficienty účelovej funkcie v tvaru

$$z - 2x_1 - 3x_2 = 0,$$

avšak stĺpec premennej z vynecháme.

Túto tabuľku interpretujeme tak, že všetky premenné s výnimkou bázických majú hodnotu nula, pričom hodnota bázických premenných je na pravej strane. Teda v úvodnej tabuľke máme $v_1 = 10$, $v_2 = 6$, $v_3 = 4$, $x_1 = 0$ a $x_2 = 0$. Podobne čítame aj posledný riadok, čiže $z = 0$.

	x_1	x_2	v_1	v_2	v_3	
v_1	1	2	1	0	0	10
v_2	1	1	0	1	0	6
v_3	1	0	0	0	1	4
z	-2	-3	0	0	0	0

Ak sú všetky koeficienty v strednej časti posledného riadku simplexovej tabuľky nezáporné, tak je súčasné riešenie optimálne.

Ak súčasné riešenie nie je optimálne, tak si zvolíme jeden zo stĺpcov, v ktorom sa v poslednom riadku vyskytuje záporné číslo. V našom príklade si vyberieme druhý stĺpec (stĺpec premennej x_2). Teraz nájdeme minimum zo zlomkov $\frac{b_i}{a_{i,2}}$, čiže

$\min\{\frac{10}{2}, \frac{6}{1}\}$ (zlomky pre koeficienty $a_{i,2} \leq 0$ neuvažujeme). Keďže minimom je hodnota $\frac{10}{2} = 5$, koeficient 2 v prvom riadku zakrúžkujeme. Tento koeficient sa nazýva **pivot** a vyššie opísaná voľba zaručuje, že pravá strana zostane nezáporná.

Teraz budeme eliminovať čísla v druhom stĺpci ekvivalentnými riadkovými operáciami (Gaussovou eliminačnou metódou) tak, aby sa pivot zmenil na 1. Dostávame novú simplexovú tabuľku, v ktorej sú hodnoty bázických premenných $x_2 = 5$, $v_2 = 1$ a $v_3 = 4$.

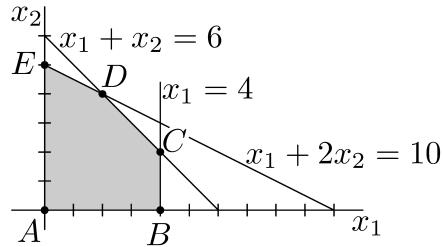
	x_1	x_2	v_1	v_2	v_3	
x_2	$\frac{1}{2}$	1	$\frac{1}{2}$	0	0	5
v_2	$\frac{1}{2}$	0	$-\frac{1}{2}$	1	0	1
v_3	1	0	0	0	1	4
z	$-\frac{1}{2}$	0	$\frac{3}{2}$	0	0	15

V tejto tabuľke je už len jeden stĺpec so zápornou hodnotou v poslednom riadku. Keďže $\min\{\frac{5}{1/2}, \frac{1}{1/2}, \frac{4}{1}\} = \frac{1}{1/2}$, tak pivot leží v druhom riadku. Po eliminovaní prvého stĺpca dostávame

	x_1	x_2	v_1	v_2	v_3	
x_2	0	1	1	-1	0	4
x_1	1	0	-1	2	0	2
v_3	0	0	1	-2	1	2
z	0	0	1	1	0	16

Keďže koeficienty v poslednom riadku sú nezáporné, súčasné riešenie je optimálne. Teda optimálne riešenie našej úlohy je $x_1 = 2$, $x_2 = 4$ ($v_1 = 0$, $v_2 = 0$, $v_3 = 2$) a $z = 16$.

V predchádzajúcim príklade boli iba dve „podstatné“ premenné, x_1 a x_2 , preto možno prípustné riešenia znázorniť v rovine s osami x_1 a x_2 , pozri tieňovanú oblasť na Obrázku 1.



Obrázok 1

Všimnime si, že úvodná simplexová tabuľka zodpovedá bodu A, druhá bodu E a posledná tabuľka zodpovedá bodu D, ktorý je optimálnym riešením úlohy.

Simplexový algoritmus teda pracuje tak, že prechádza od jedného prípustného

riešenia k ďalšiemu, pričom tieto riešenia sú „vrcholmi“ útvaru (viacozmerného mnohostena), ktorý je určený ohraničeniami, a algoritmus sa pohybuje „po hranách“ tohto útvaru.

Ako sme videli, simplexový algoritmus je založený na Gaussovej eliminácii. Úvodná tabuľka má tvar

$$\begin{array}{c} \mathbb{A} \cdot \mathbf{x} = \mathbf{b} \\ \hline z - \mathbf{c}^T \cdot \mathbf{x} = 0 \end{array}$$

pričom do tejto tabuľky zapisujeme iba koeficienty, názvy premenných máme v záhlaví. V každom riadku simplexovej tabuľky (každej, nielen úvodnej) máme jeden koeficient 1 taký, že v stĺpci, v ktorom je tento koeficient, sú všetky ostatné hodnoty 0. Znamená to, že v každej simplexovej tabuľke máme podmaticu rovnajúcu sa identickej matici \mathbb{I} . Stĺpce tejto podmatice môžu byť v tabuľke poprehadzované, avšak spočítanie je komutatívne, preto si ich môžeme uložiť do správneho poradia. Premenné zodpovedajúce 1-kám matice \mathbb{I} sú bázické.

Označme \mathbf{x}_b bázické premenné jednej konkrétnej (hoci finálnej, čiže optimálnej) simplexovej tabuľky. Potom má táto tabuľka tvar (stĺpce sme poprehadzovali do vhodného poradia)

	\mathbf{x}_b^T	\mathbf{x}_n^T	
\mathbf{x}_b	\mathbb{I}	.	.
z	$\mathbf{0}^T$.	.

kde \mathbf{x}_n obsahuje **nebázické** premenné, čiže tie premenné z \mathbf{x} , ktoré nie sú v \mathbf{x}_b , a matice označené \cdot potrebujeme určiť. Rozložme si stĺpce \mathbb{A} na tie, ktoré zodpovedajú bázickým premenným a tie, ktoré zodpovedajú nebázickým premenným. Dostávame $\mathbb{A} = (\mathbb{A}_b | \mathbb{A}_n)$. Podobne rozložme \mathbf{c} na \mathbf{c}_b a \mathbf{c}_n . Potom má úvodná tabuľka tvar

$$\begin{array}{c} \mathbb{A}_b \cdot \mathbf{x}_b + \mathbb{A}_n \cdot \mathbf{x}_n = \mathbf{b} \\ \hline z - \mathbf{c}_b^T \cdot \mathbf{x}_b - \mathbf{c}_n^T \cdot \mathbf{x}_n = 0 \end{array}$$

a nami uvažovaná tabuľka, ktorú sme získali z úvodnej Gaussovou elimináciou, má namiesto \mathbb{A}_b už iba \mathbb{I} . Keďže ekvivalentné riadkové operácie zodpovedajú násobeniu regulárnej maticou zľava, museli sme našu tabuľku dostať z úvodnej prenásobením maticou \mathbb{A}_b^{-1} zľava, pretože $\mathbb{A}_b^{-1} \cdot \mathbb{A}_b = \mathbb{I}$. Keďže

$$\begin{aligned} \mathbb{A}_b \cdot \mathbf{x}_b + \mathbb{A}_n \cdot \mathbf{x}_n &= \mathbf{b} && / \mathbb{A}_b^{-1}. \\ \text{dá} \quad \mathbb{I} \cdot \mathbf{x}_b + \mathbb{A}_b^{-1} \cdot \mathbb{A}_n \cdot \mathbf{x}_n &= \mathbb{A}_b^{-1} \cdot \mathbf{b} \end{aligned}$$

a výraz $\mathbf{0}^T \cdot \mathbf{x}_b$ v poslednom riadku dostaneme po pripočítaní \mathbf{c}_b^T -násobku predchádzajúceho výrazu k

$$z - \mathbf{c}_b^T \cdot \mathbf{x}_b - \mathbf{c}_n^T \cdot \mathbf{x}_n = 0$$

tak naša simplexová tabuľka má tvar

	\mathbf{x}_b^T	\mathbf{x}_n^T	
\mathbf{x}_b	\mathbb{I}	$\mathbb{A}_b^{-1} \cdot \mathbb{A}_n$	$\mathbb{A}_n^{-1} \cdot \mathbf{b}$
z	$\mathbf{0}^T$	$\mathbf{c}_b^T \cdot \mathbb{A}_b^{-1} \cdot \mathbb{A}_n - \mathbf{c}_n^T$	$\mathbf{c}_b^T \cdot \mathbb{A}_b^{-1} \cdot \mathbf{b}$

Simplexová tabuľka je finálna, čiže optimálna, ak má v strednej časti posledného riadku iba nezáporné koeficienty, čiže keď $\mathbf{c}_b^T \cdot \mathbb{A}_b^{-1} \cdot \mathbb{A}_n - \mathbf{c}_n^T \geq \mathbf{0}^T$.

V druhom ročníku sme pracovali s celou simplexovou tabuľkou. Existujú metódy (algoritmy), ktoré si nepamätajú celú simplexovú tabuľku, ale iba \mathbb{A}_b^{-1} a čosi na výber pivota. Týmito metódami sa tu však nebudeme zaoberať, zameriame sa na **metódy vnútorného bodu (central path metódy)**. Na ne budeme potrebovať dualitu.

Dualita

Najprv si na príklade zopakujeme, ako zostrojiť k danej úlohe lineárneho programovania duálnu úlohu.

PRÍKLAD. Uvažujte úlohu lineárneho programovania

$$\begin{array}{l} \min z = 3x_1 + x_2 - x_3 \\ \hline x_1 - 2x_2 + x_3 \geq 2 \\ x_1 + x_2 + 3x_3 = 6 \\ 2x_1 - 5x_2 + x_3 \leq 3 \\ 5x_1 - x_2 - 6x_3 = 1 \\ \hline x_1, x_3 \geq 0, x_2 \in \mathbb{R} \end{array}$$

Zostrojte k nej duálnu úlohu.

RIEŠENIE. Úlohu si najprv prepíšeme tak, aby šli všetky nerovnosti proti smeru optimalizácie. Teda pre minimalizačnú úlohu musia byť všetky nerovnosti typu \geq . To dosiahneme prenásobením opačných nerovníc hodnotou -1 . Potom je daná **primárna úloha** ekvivalentná s úlohou

$$\begin{array}{l} \min z = 3x_1 + x_2 - x_3 \\ \hline x_1 - 2x_2 + x_3 \geq 2 \\ x_1 + x_2 + 3x_3 = 6 \\ -2x_1 + 5x_2 - x_3 \geq -3 \\ 5x_1 - x_2 - 6x_3 = 1 \\ \hline x_1, x_3 \geq 0, x_2 \in \mathbb{R} \end{array}$$

Teraz už môžeme zstrojiť duálnu úlohu. Ak je primárna úloha

$$\begin{array}{c} \min z = \mathbf{c}^T \cdot \mathbf{x} \\ \hline \mathbb{A} \cdot \mathbf{x} \leq \mathbf{b} \\ x_i \geq 0 \text{ pre nejaké } i \end{array} \quad \text{duálna bude} \quad \begin{array}{c} \max w = \mathbf{b}^T \cdot \mathbf{y} \\ \hline \mathbb{A}^T \cdot \mathbf{y} \geq \mathbf{c} \\ y_j \geq 0 \text{ pre nejaké } j \end{array} \quad (1)$$

Znamená to, že $\mathbf{y} = (y_1, \dots, y_m)^T$. Zostáva určiť „nejaké“ i a j a tvar \square a \triangle . Tu to funguje tak, že nezápornej premennej jednej úlohy zodpovedá nerovnica v druhej (idúca proti smeru optimalizácie tejto úlohy) a premenným bez obmedzenia na nezápornosť zodpovedajú rovnice. Takže duálna úloha bude

$$\begin{array}{c} \max w = 2y_1 + 6y_2 - 3y_3 + y_4 \\ \hline y_1 + y_2 - 2y_3 + 5y_4 \leq 3 \\ -2y_1 + y_2 + 5y_3 - y_4 = 1 \\ y_1 + 3y_2 - y_3 - 6y_4 \leq -1 \\ \hline y_1, y_3 \geq 0, y_2, y_4 \in \mathbb{R} \end{array}$$

Všimnime si, že ak sú v primárnej úlohe všetky ohraničenia rovnice a $x_i \geq 0$ pre všetky i , tak prípustné riešenie duálnej úlohy (ktoré je možné chápať ako koeficienty lineárnej kombinácie riadkov matice \mathbb{A}) spĺňa $\mathbb{A}^T \cdot \mathbf{y} \leq \mathbf{c}$, čiže $\mathbf{y}^T \cdot \mathbb{A}^T \leq \mathbf{c}^T$. Kedže \mathbf{x} je prípustné riešenie primárnej úlohy, tak pre každé prípustné riešenie \mathbf{y} duálnej úlohy platí $\mathbf{b}^T \cdot \mathbf{y} = \mathbf{x}^T \cdot \mathbb{A}^T \cdot \mathbf{y} = \mathbf{y}^T \cdot \mathbb{A} \cdot \mathbf{x} \leq \mathbf{c}^T \cdot \mathbf{x}$, čiže hodnota účelovej funkcie duálnej úlohy pre prípustné riešenie \mathbf{y} je dolným ohraničením na hodnotu účelovej funkcie primárnej úlohy pre prípustné riešenie \mathbf{x} . Samozrejme, najlepšie dolné ohraničenie dostaneme, keď bude $\mathbf{b}^T \cdot \mathbf{y}$ čo najväčšie. Podobná závislosť je splnená, aj keď nie sú všetky ohraničenia primárnej úlohy rovnice a $x_i \geq 0$ nie je splnené pre všetky i . Platí nasledujúce tvrdenie

VETA (o dualite). Nech je P minimalizačná primárna úloha lineárneho programovania a D nech je k nej zstrojená duálna úloha, pričom označenie je ako v (1). Potom pre každé prípustné riešenie \mathbf{x} úlohy P a pre každé prípustné riešenie \mathbf{y} úlohy D platí $\mathbf{c}^T \cdot \mathbf{x} \geq \mathbf{b}^T \cdot \mathbf{y}$. Naviac, ak má jedna z úloh P a D optimálne riešenie, tak ho má aj druhá a hodnoty účelových funkcií sa v týchto riešeniach rovnajú.

Rozoberme si, ako vyzerá podľa tejto vety vzťah medzi riešeniami primárnej a duálnej úlohy.

P má	D má
optimálne riešenie	optimálne riešenie
prípustné, ale nie optimálne riešenie	nemá ani len prípustné riešenie
nemá prípustné riešenie	prípustné, ale nie optimálne riešenie, alebo nemá ani len prípustné riešenie

Podmienky komplementarity

V tejto časti zavedieme podmienky komplementarity. Avšak zavedieme ich len pre primárnu úlohu v špeciálnom (v podstate skoro kanonickom) tvaru, pretože len pre takúto úlohu ich budeme využívať. V ďalšom budeme používať tu zavedené označenie. Označme P primárnu úlohu, ktorá je minimalizačná, má nezáporné premenné a všetky ohraničenia sú rovnice. Ďalej označme D' úlohu duálnu k P a označme D úlohu ekvivalentnú s D' , ktorá má iba ohraničenia tvaru rovníc. Potom

$$\begin{array}{lll} P: & \frac{\min \mathbf{c}^T \cdot \mathbf{x}}{\mathbb{A} \cdot \mathbf{x} = \mathbf{b}} & D': \quad \frac{\max \mathbf{b}^T \cdot \mathbf{y}}{\mathbb{A}^T \cdot \mathbf{y} \leq \mathbf{c}} \\ & \mathbf{x} \geq \mathbf{0} & \mathbf{y} \in \mathbb{R}^m \end{array} \quad D: \quad \frac{\max \mathbf{b}^T \cdot \mathbf{y}}{\mathbb{A}^T \cdot \mathbf{y} + \mathbb{I} \cdot \mathbf{s} = \mathbf{c}}$$

$$\mathbf{y} \in \mathbb{R}^m, \mathbf{s} \geq \mathbf{0}$$

kde \mathbb{A} je typu $(m \times n)$ a $\mathbf{s} \in \mathbb{R}^n$. Úlohu D budeme nazývať upravená duálna úloha.

VETA (podmienky komplementarity). Nech je \mathbf{x} prípustné riešenie P a (\mathbf{y}, \mathbf{s}) nech je je prípustné riešenie D . Potom \mathbf{x} je optimálne riešenie P a zároveň (\mathbf{y}, \mathbf{s}) je optimálne riešenie D práve vtedy, keď $\mathbf{x}^T \cdot \mathbf{s} = 0$.

Poznamenajme, že tvrdenie $\mathbf{x}^T \cdot \mathbf{s} = 0$ je ekvivalentné tvrdeniu, že $x_i s_i = 0$ pre každé i , $1 \leq i \leq n$, keďže $\mathbf{x}, \mathbf{s} \geq \mathbf{0}$.

DÔKAZ. Nech sú \mathbf{x} a (\mathbf{y}, \mathbf{s}) ľubovoľné prípustné riešenia P a D . Keďže (\mathbf{y}, \mathbf{s}) je prípustné riešenie D , platí

$$\mathbb{A}^T \cdot \mathbf{y} + \mathbb{I} \cdot \mathbf{s} = \mathbf{c}$$

čiže

$$\mathbf{s} = \mathbf{c} - \mathbb{A}^T \cdot \mathbf{y}.$$

Po vynásobení vektorom \mathbf{x}^T zľava dostaneme

$$\mathbf{x}^T \cdot \mathbf{s} = \mathbf{x}^T \cdot \mathbf{c} - \mathbf{x}^T \cdot \mathbb{A}^T \cdot \mathbf{y} = \mathbf{c}^T \cdot \mathbf{x} - \mathbf{b}^T \cdot \mathbf{y},$$

pretože \mathbf{x} je prípustné riešenie P , čo značí, že $\mathbb{A} \cdot \mathbf{x} = \mathbf{b}$, čiže $\mathbf{x}^T \cdot \mathbb{A}^T = \mathbf{b}^T$.

Znamená to, že $\mathbf{x}^T \cdot \mathbf{s} = 0$ je ekvivalentné tvrdeniu $\mathbf{c}^T \cdot \mathbf{x} = \mathbf{b}^T \cdot \mathbf{y}$. Teda $\mathbf{x}^T \cdot \mathbf{s} = 0$ je ekvivalentné tvrdeniu, že P a D majú v \mathbf{x} a (\mathbf{y}, \mathbf{s}) rovnaké hodnoty účelovej funkcie, čo podľa vety o dualite znamená, že \mathbf{x} a (\mathbf{y}, \mathbf{s}) sú optimálne riešenia P a D . \square

Veta o komplementarite nám ponúka inú možnosť, ako vyriešiť úlohu P . Vôbec nie je potrebné hľadať akési optimum. Stačí nájsť \mathbf{x} , \mathbf{y} a \mathbf{s} také, aby platilo

$$\begin{array}{l} \mathbb{A} \cdot \mathbf{x} = \mathbf{b} \\ \mathbb{A}^T \cdot \mathbf{y} + \mathbb{I} \cdot \mathbf{s} = \mathbf{c} \\ \mathbf{x}^T \cdot \mathbf{s} = 0 \end{array} \tag{2}$$

kde $\mathbf{x}, \mathbf{s} \in \mathbb{R}^n$, $\mathbf{y} \in \mathbb{R}^m$ a $\mathbf{x}, \mathbf{s} \geq \mathbf{0}$. Toto budeme v ďalšom využívať. Sústava (2) je lineárna až na rovnicu $\mathbf{x}^T \cdot \mathbf{s} = 0$. Práve táto rovnica v nej spôsobuje komplikácie, bez nej by bolo hľadanie riešenia sústavy (2) jednoduché.

Cvičenia

CVIČENIE 1.1. Farmár vlastní 120 hektárov pôdy, na ktorej pestuje kukuricu a ovos. Pri kukurici dosahuje zisk 80 €/ha a pri ovse 60 €/ha. Žatva kukurice trvá 2 h/ha a žatva ovsa 2 h/ha. Farmár má na žatvu k dispozícii 320 hodín práce. Na akú rozlohu má zasiať kukuricu a na akú ovos, aby maximalizoval zisk?

Úlohu najprv matematicky sformulujte. Potom ju vyriešte graficky (nakreslite prípustnú množinu podobne ako na Obrázku 1, zobrazte vrstevnice účelovej funkcie a zistite, ktorý bod je optimálny) a vyriešte úlohu aj simplexovým algoritmom.

CVIČENIE 1.2. Nájdite na internete zaujímavý problém lineárneho programovania a vyriešte ho graficky aj simplexovým algoritmom. Riešenie bude obsahovať

1. Slovné znenie.
2. Matematickú formuláciu.

- a) Premenné (mali by byť dve).
- b) Účelovú funkciu.
- c) Aspoň štyri ohraničenia, také, že $(0, 0)$ bude prípustný bod.
- d) Obrázok s prípustnou množinou a vrstevnicami účelovej funkcie.

3. Grafické riešenie.

4. Riešenie simplexovým algoritmom.

Ktorá časť riešenia bude náročnejšia, ak nebude $(0, 0)$ prípustný bod?

CVIČENIE 1.3. Priateľka študenta MPM si praje k narodeninám tyrkysovostrieborný náhrdelník. Náhrdelník má byť aspoň 36 cm dlhý, ale nie dlhší ako 48 cm. Chce, aby v ňom bolo aspoň dvakrát viac strieborných ako tyrkysových korálok. Študent MPM nezaváha a podakuje priateľke za pekne sformulovaný problém lineárneho programovania. V obchode majú korálky s priemerom 0,5 cm v cene: strieborná 2 €/kus, tyrkysová 1€/kus. Študent chce vyrobiť najlacnejší náhrdelník spĺňajúci priateľkine predstavy. Koľko akých korálok má kúpiť, koľko to bude stáť a aký bude náhrdelník dlhý?

Úlohu najprv matematicky sformulujte a potom ju vyriešte graficky a aj (dvojfázovým, pozri [10]) simplexovým algoritmom. Pre uľahčenie riešenia simplexovým algoritmom vytvorte v softvéri Mathematica funkciu `pivot[M_,i_,j_]`, ktorá spraví jeden krok simplexového algoritmu na matici M , pričom pivotom bude prvok $M_{i,j}$. Funkciu `pivot[M_,i_,j_]` použite pre vykonanie krokov simplexového algoritmu.

CVIČENIE 1.4. Odvodte a graficky vyriešte duálnu úlohu k úlohe z Cvičenia 1.1.

CVIČENIE 1.5. Sformulujte a vyriešte duálnu úlohu k úlohe z Cvičenia 1.2.

CVIČENIE 1.6. Nech je U úloha lineárneho programovania v kanonickom tvare. Ukážte, že duálna úloha k duálnej úlohe k U je totžná s U .

2 Metóda vnútorného bodu

Z predchádzajúcej kapitoly vieme (pozri (2)), že namiesto P môžeme riešiť sústavu (systém rovníc)

$$\begin{aligned} \mathbb{A} \cdot \mathbf{x} &= \mathbf{b} \\ Sys^0 : \quad \mathbb{A}^T \cdot \mathbf{y} + \mathbf{s} &= \mathbf{c} \\ \mathbf{x}^T \cdot \mathbf{s} &= 0 \\ \mathbf{x}, \mathbf{s} &\in \mathbb{R}^n, \quad \mathbf{y} \in \mathbb{R}^m \text{ a } \mathbf{x}, \mathbf{s} \geq \mathbf{0} \end{aligned}$$

Podmienka vnútorného bodu

Na riešenie sústavy Sys^0 budeme potrebovať takzvaný **vnútorný bod prípustnej oblasti**. Označme

$$\begin{aligned} \mathcal{F}_P^o &= \{\mathbf{x} \in \mathbb{R}^n; \mathbb{A} \cdot \mathbf{x} = \mathbf{b} \text{ a } \mathbf{x} > \mathbf{0}\} && \text{a} \\ \mathcal{F}_D^o &= \{(\mathbf{y}, \mathbf{s}) \in \mathbb{R}^m \times \mathbb{R}^n; \mathbb{A}^T \cdot \mathbf{y} + \mathbf{s} = \mathbf{c} \text{ a } \mathbf{s} > \mathbf{0}\} && \text{respektíve} \\ \mathcal{F}^o &= \{(\mathbf{x}, \mathbf{y}, \mathbf{s}) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n; \mathbb{A} \cdot \mathbf{x} = \mathbf{b}, \mathbb{A}^T \cdot \mathbf{y} + \mathbf{s} = \mathbf{c} \text{ a } \mathbf{x}, \mathbf{s} > \mathbf{0}\}. \end{aligned}$$

Prvky \mathcal{F}^o sú **vnútorné body** systému Sys^0 .

DEFINÍCIA. Systém Sys^0 spĺňa **IPC** (interior point condition), čiže **podmienku vnútorného bodu**, ak $\mathcal{F}^o \neq \emptyset$, čiže ak $\mathcal{F}_P^o \neq \emptyset$ a $\mathcal{F}_D^o \neq \emptyset$.

Vnútorný bod potrebujeme preto, lebo v ňom spustíme algoritmus. V skutočnosti budú všetky postupne nájdené body algoritmu vnútorné.

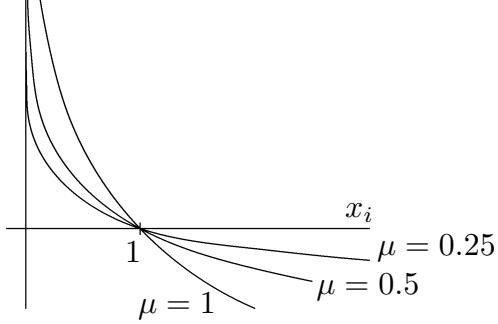
Centrálna cesta

Vnútorný bod $(\mathbf{x}, \mathbf{y}, \mathbf{s}) \in \mathcal{F}^o$, respektíve $\mathbf{x} \in \mathcal{F}_P^o$, nájdeme pomocou **bariérovej funkcie**. Na objasnenie postupu sa vrátim k pôvodnej úlohe P. Problém

$$\begin{array}{c} \min \mathbf{c}^T \cdot \mathbf{x} \\ \hline \mathbb{A} \cdot \mathbf{x} = \mathbf{b} \\ \mathbf{x} \geq \mathbf{0} \end{array} \quad \text{nahradíme} \quad \begin{array}{c} \min \left(\mathbf{c}^T \cdot \mathbf{x} - \mu \sum_{i=1}^n \ln(x_i) \right) \\ \hline \mathbb{A} \cdot \mathbf{x} = \mathbf{b} \\ (\mathbf{x} > \mathbf{0}) \end{array}$$

Funkcia $-\mu \sum_{i=1}^n \ln(x_i)$ sa nazýva **bariéra (bariérová funkcia)**, pričom $\mu > 0$. (Na pohľad sme nespravili nič rozumné, pretože nová sústava je definovaná iba pre

$\mathbf{x} > 0$, no pri iteratívnom algoritme to zmysel mať bude, pozri ďalej.) Na Obrázku 2 máme graf funkcie $-\mu \ln(x_i)$ pre rôzne μ . Z tohto grafu vidno, že pre každé μ bude riešením danej sústavy vnútorný bod $\mathbf{x}^*(\mu)$ patriaci do \mathcal{F}_P^o , a v limite pre $\mu \rightarrow 0$ bude $\mathbf{x}^*(0)$ optimálnym riešením \mathbf{x}^* pôvodnej úlohy P, pozri tiež Obrázok 3.



Obrázok 2

V novej sústave máme iba ohraničenia tvaru rovníc, takže ju môžeme vyriešiť pomocou Lagrangeových multiplikátorov. Lagrangeova funkcia je

$$L(\mathbf{x}, \boldsymbol{\lambda}) = \mathbf{c}^T \cdot \mathbf{x} - \mu \sum_{i=1}^n \ln(x_i) + \boldsymbol{\lambda}^T \cdot (\mathbf{b} - \mathbb{A} \cdot \mathbf{x}).$$

Kedže $\boldsymbol{\lambda}^T \cdot (\mathbf{b} - \mathbb{A} \cdot \mathbf{x})$ je číslo, platí $\boldsymbol{\lambda}^T \cdot (\mathbf{b} - \mathbb{A} \cdot \mathbf{x}) = [\boldsymbol{\lambda}^T \cdot (\mathbf{b} - \mathbb{A} \cdot \mathbf{x})]^T$. Preto

$$L(\mathbf{x}, \boldsymbol{\lambda}) = \mathbf{c}^T \cdot \mathbf{x} - \mu \sum_{i=1}^n \ln(x_i) + (\mathbf{b}^T - \mathbf{x}^T \cdot \mathbb{A}^T) \cdot \boldsymbol{\lambda}.$$

Z toho dostávame gradient (uvážte $\frac{\partial}{\partial x_i} L(\mathbf{x}, \boldsymbol{\lambda})$)

$$\nabla_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\lambda}) = \mathbf{c} - \mu \left(\frac{1}{x_1}, \dots, \frac{1}{x_n} \right)^T - \mathbb{A}^T \cdot \boldsymbol{\lambda}.$$

Položením $\nabla_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\lambda}) = 0$ dostoneme

$$\mathbb{A}^T \cdot \boldsymbol{\lambda} + \mu \left(\frac{1}{x_1}, \dots, \frac{1}{x_n} \right)^T = \mathbf{c}.$$

Porovnajme túto rovnicu s ohraničeniami duálnej úlohy $\mathbb{A}^T \cdot \mathbf{y} + \mathbf{s} = \mathbf{c}$. Vidíme, že $\boldsymbol{\lambda}, \mu \left(\frac{1}{x_1}, \dots, \frac{1}{x_n} \right)^T$ musí byť prípustným bodom upravenej duálnej úlohy D, keďže $\mu \left(\frac{1}{x_1}, \dots, \frac{1}{x_n} \right)^T \geq \mathbf{0}$. Tento bod je dokonca vnútorným, čiže je z \mathcal{F}_D^o .

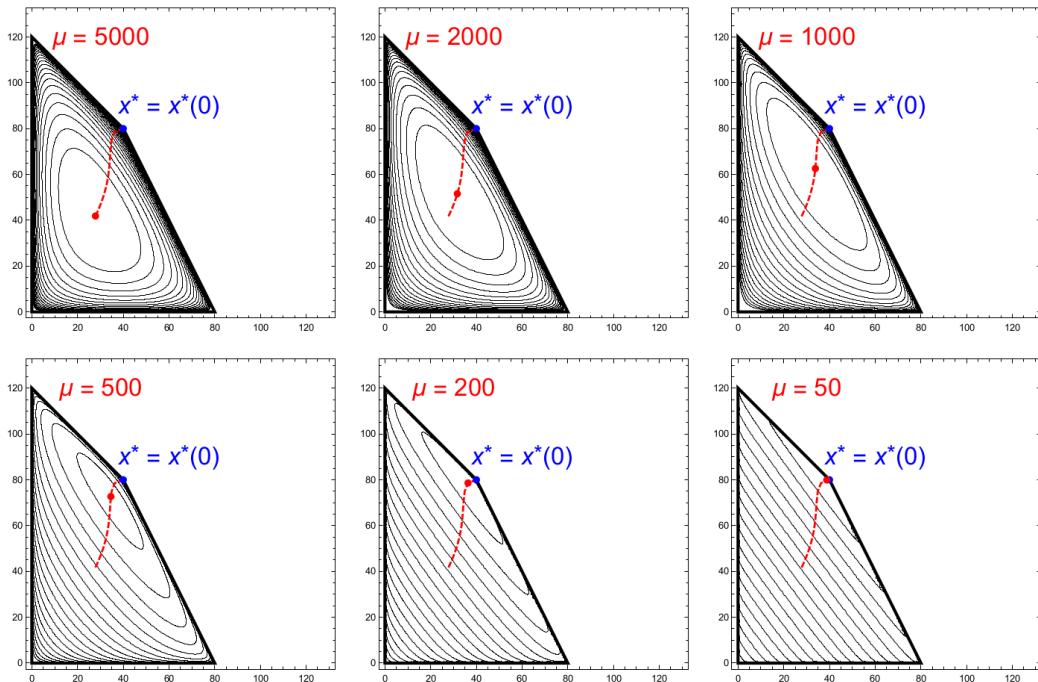
Teda ak chceme nájsť vnútorný bod \mathcal{F}^o , stačí riešiť sústavu

$$\begin{aligned} \mathbb{A} \cdot \mathbf{x} &= \mathbf{b} \\ \mathbb{A}^T \cdot \mathbf{y} + \mathbf{s} &= \mathbf{c} \\ Sys^\mu : \quad x_1 s_1 &= \mu \\ &\vdots \\ &x_n s_n = \mu \\ \mathbf{x}, \mathbf{s} \in \mathbb{R}^n, \quad \mathbf{y} \in \mathbb{R}^m \text{ a } \mathbf{x}, \mathbf{s} > \mathbf{0}. \end{aligned}$$

Všimnime si, že keď budeme mať aparát na riešenie Sys^μ , tak budeme vedieť vyriešiť aj Sys^0 , čiže budeme vedieť nájsť riešenie pôvodnej úlohy P. Stačí postupne zmenšovať μ a riešiť Sys^μ , pričom každá ďalšia iterácia bude bližšie k riešeniu Sys^0 .

DEFINÍCIA. Ak je systém Sys^0 „pekný“, tak pre každé $\mu \geq 0$ má Sys^μ jediné riešenie a body týchto riešení tvoria **centrálnu cestu** (**central path**).

Na Obrázku 3 je situácia znázornená v prípade $n = 2$ pre úlohu z Cvičenia 1.1. Hrubou čiernou čiarou je vykreslená hranica prípustnej množiny. Tenké čierne čiary sú vrstevnice (izočiary) funkcie $\mathbf{c}^T \cdot \mathbf{x} + \mu\phi(\mathbf{x})$, kde sme do bariérovej funkcie zahrnuli aj ohraničenia tvaru nerovníc, pre rôzne hodnoty parametra μ . V obrázkoch je zakreslené aj optimum pôvodnej úlohy $\mathbf{x}^* = \mathbf{x}^*(0) = (40, 80)$ (pozri Cvičenie 2.1). Červená prerušovaná čiara je centrálna cesta. Pre každú hodnotu parametra μ je vykreslené aj optimum $\mathbf{x}^*(\mu)$ modifikovanej úlohy (červený bod), čiže minimum funkcie $\mathbf{c}^T \cdot \mathbf{x} + \mu\phi(\mathbf{x})$. Ako vidno, pre veľké μ sa bod $\mathbf{x}^*(\mu)$ nachádza približne v strede prípustnej množiny riešení a s klesajúcim μ sa približuje (sledujúc centrálnu cestu) k optimu \mathbf{x}^* . Všimnime si tiež, že pre malé μ je funkcia $\mathbf{c}^T \cdot \mathbf{x} + \mu\phi(\mathbf{x})$ vo vnútri prípustnej množiny takmer identická s pôvodnou lineárnom účelovou funkciou $\mathbf{c}^T \cdot \mathbf{x}$.



Obrázok 3

PRÍKLAD 1. Majme úlohu

$$\begin{array}{c} \min -2x_1 + x_2 - 3x_3 \\ \text{P:} \quad \frac{x_1 + x_2 + x_3 = 1}{x_1, x_2, x_3 \geq 0} \end{array}$$

Zostrojte centrálnu cestu.

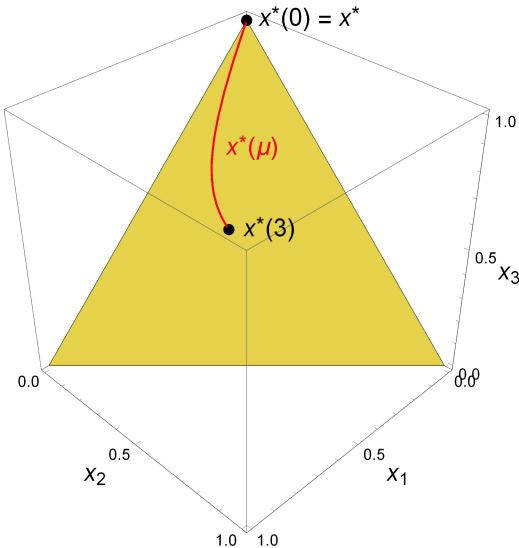
RIEŠENIE. Duálna úloha je

$$\begin{array}{rcl}
 \max y \\
 \hline
 y + s_1 & = -2 \\
 \text{D:} \quad y + s_2 & = 1 \\
 y + s_3 & = -3 \\
 \hline
 s_1, s_2, s_3 \geq 0
 \end{array}$$

Takže body centrálnej cesty sú riešenia sústavy

$$\begin{aligned}
 x_1 + x_2 + x_3 &= 1 \\
 y + s_1 &= -2 \\
 y + s_2 &= 1 \\
 y + s_3 &= -3 \\
 x_1 \cdot s_1 &= \mu \\
 x_2 \cdot s_2 &= \mu \\
 x_3 \cdot s_3 &= \mu
 \end{aligned}$$

Označme riešenie pre dané $\mu > 0$ symbolom $x^*(\mu)$. Potom $x^*(\mu)$ pre $\mu \in \langle 0, 3 \rangle$ získané pomocou softvéru Mathematica sú znázornené červenou krivkou na Obrázku 4, pričom $x^*(3)$ je bod približne v strede žltého trojuholníka a $x^*(0) = x^*$ je optimálne riešenie v bode $[0, 0, 1]$.



Obrázok 4

Newtonova metóda hľadania nulového bodu

Príklad 1 bol jednoduchý, pretože primárna úloha P mala iba jedno ohraničenie. Ako však vyriešiť nelineárnu sústavu Sys^μ keď bude primárna úloha obsahovať veľa

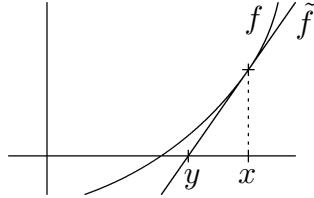
ohraničení a veľa premenných? Keď v sústave Sys^μ dáme konštanty na ľavú stranu, tak našou úlohou je vlastne nájsť nulový bod viachodnotovej funkcie. V tejto krátkej vsunutej podkapitole si ukážeme Newtonovu metódu hľadania nulového bodu viachodnotovej funkcie. Upozorňujeme, že v tejto podkapitole sú \mathbf{x} a \mathbf{y} definované ináč ako vo zvyšku učebnice, budeme totiž požadovať $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$.

Začneme s prípadom $n = 1$. Pripomeňme si, ako pomocou Newtonovej metódy hľadáme nulový bod funkcie, pozri Obrázok 5.

Majme „peknú“ funkciu f a $x \in D_f$, kde D_f je definičný obor f . (Poznamenajme, že „peknou“ je napríklad funkcia so spojitou deriváciou.) Zostrojíme Taylorov rozvoj prvého rádu f v bode x , čiže

$$\tilde{f}(y) = f(x) + f'(x)(y - x).$$

Tu y je premenná a x je konštanta. Grafom $\tilde{f}(y)$ je priamka, ktorá je dotyčnicou ku grafu funkcie f v bode x , a riešením rovnice $\tilde{f}(y) = 0$ je bod, ktorý leží (pre „peknú“ funkciu f a vhodnú polohu x) bližšie k nulovému bodu funkcie f .



Obrázok 5

Podobný postup funguje pre $n \geq 2$. Nech je F n -hodnotová funkcia n premenných, čiže

$$F(\mathbf{x}) = \left(F_1(\mathbf{x}), \dots, F_n(\mathbf{x}) \right)^T.$$

Poznamenajme, že vo všeobecnosti sa n -hodnotová funkcia nazýva vektorové pole v \mathbb{R}^n . Hľadáme nulový bod F , čiže taký, pre ktorý $F(\mathbf{x}) = \mathbf{0}$. Nech je najprv \mathbf{x} bod z definičného oboru F , čiže $\mathbf{x} \in D_F$. Spravme Taylorov rozvoj prvého rádu v \mathbf{x} . Pre zložky F_i funkcie F definujme

$$\tilde{F}_i(\mathbf{y}) = F_i(\mathbf{x}) + \sum_{j=1}^n \frac{\partial}{\partial y_j} F_i(\mathbf{x}) \cdot (y_j - x_j)$$

a položme $\tilde{F}_i(\mathbf{y}) = 0$ pre $1 \leq i \leq n$. Keď tento proces budeme iterovať, tak pre $\mathbf{x} = \mathbf{x}^k$ a $\mathbf{y} = \mathbf{x}^{k+1}$ dostaneme

$$\begin{pmatrix} \tilde{F}_1(\mathbf{x}^{k+1}) \\ \vdots \\ \tilde{F}_n(\mathbf{x}^{k+1}) \end{pmatrix} = \begin{pmatrix} F_1(\mathbf{x}^k) \\ \vdots \\ F_n(\mathbf{x}^k) \end{pmatrix} + \begin{pmatrix} \frac{\partial}{\partial x_1} F_1(\mathbf{x}^k) & \dots & \frac{\partial}{\partial x_n} F_1(\mathbf{x}^k) \\ \vdots & \ddots & \vdots \\ \frac{\partial}{\partial x_1} F_n(\mathbf{x}^k) & \dots & \frac{\partial}{\partial x_n} F_n(\mathbf{x}^k) \end{pmatrix} \cdot \begin{pmatrix} x_1^{k+1} - x_1^k \\ \vdots \\ x_n^{k+1} - x_n^k \end{pmatrix}$$

kde $\tilde{F}(\mathbf{x}^{k+1}) = \mathbf{0}$, pričom maticu typu $(n \times n)$ nazývame **Jacobiho matica** funkcie F v bode \mathbf{x}^k . Označme túto maticu $DF(\mathbf{x}^k)$. Získame skrátený zápis

$$F(\mathbf{x}^k) + DF(\mathbf{x}^k) \cdot (\mathbf{x}^{k+1} - \mathbf{x}^k) = \mathbf{0}.$$

Rozdiel $\mathbf{x}^{k+1} - \mathbf{x}^k$ nazývame **Newtonov krok** a označujeme ho $\Delta\mathbf{x}^k$. Newtonov krok získame vyriešením sústavy lineárnych rovníc

$$DF(\mathbf{x}^k) \cdot \Delta\mathbf{x}^k = -F(\mathbf{x}^k), \quad (3)$$

pričom $\mathbf{x}^{k+1} = \mathbf{x}^k + \Delta\mathbf{x}^k$.

Newtonova metóda pre centrálnu cestu

Teraz túto metódu napasujeme na Sys^μ , avšak poradie prvých dvoch rovníc vymeníme. Teda funkcia F bude mať tvar

$$F^\mu(\mathbf{x}, \mathbf{y}, \mathbf{s}) = \begin{pmatrix} \mathbb{A}^T \cdot \mathbf{y} + \mathbf{s} - \mathbf{c} \\ \mathbb{A} \cdot \mathbf{x} - \mathbf{b} \\ x_1 s_1 - \mu \\ \vdots \\ x_n s_n - \mu \end{pmatrix}$$

pričom potrebujeme riešiť rovnicu $F^\mu(\mathbf{x}, \mathbf{y}, \mathbf{s}) = \mathbf{0}$. Dostávame Jacobiho maticu

$$DF_\mu = \begin{pmatrix} \mathbb{0} & \mathbb{A}^T & \mathbb{I} \\ \mathbb{A} & \mathbb{0} & \mathbb{0} \\ \mathbb{S} & \mathbb{0} & \mathbb{X} \end{pmatrix},$$

kde \mathbb{X} a \mathbb{S} sú diagonálne matice

$$\mathbb{X} = \begin{pmatrix} x_1 & & 0 \\ & \ddots & \\ 0 & & x_n \end{pmatrix} \quad \text{a} \quad \mathbb{S} = \begin{pmatrix} s_1 & & 0 \\ & \ddots & \\ 0 & & s_n \end{pmatrix}.$$

Čiže budeme riešiť systém

$$\begin{pmatrix} \mathbb{0} & \mathbb{A}^T & \mathbb{I} \\ \mathbb{A} & \mathbb{0} & \mathbb{0} \\ \mathbb{S}^k & \mathbb{0} & \mathbb{X}^k \end{pmatrix} \cdot \begin{pmatrix} \Delta\mathbf{x}^k \\ \Delta\mathbf{y}^k \\ \Delta\mathbf{s}^k \end{pmatrix} = - \begin{pmatrix} \mathbb{A}^T \cdot \mathbf{y}^k + \mathbf{s}^k - \mathbf{c} \\ \mathbb{A} \cdot \mathbf{x}^k - \mathbf{b} \\ x_1^k s_1^k - \mu \\ \vdots \\ x_n^k s_n^k - \mu \end{pmatrix}, \quad (4)$$

kde

$$\mathbb{X}^k = \begin{pmatrix} x_1^k & & 0 \\ & \ddots & \\ 0 & & x_n^k \end{pmatrix} \quad \text{a} \quad \mathbb{S}^k = \begin{pmatrix} s_1^k & & 0 \\ & \ddots & \\ 0 & & s_n^k \end{pmatrix}.$$

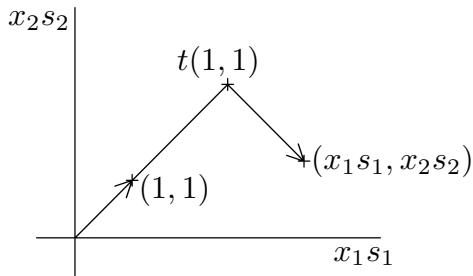
Po vyriešení tohto systému pre nasledujúcu iteráciu kladieme

$$\begin{aligned} \mathbf{x}^{k+1} &= \mathbf{x}^k + \Delta\mathbf{x}^k \\ \mathbf{y}^{k+1} &= \mathbf{y}^k + \Delta\mathbf{y}^k \\ \mathbf{s}^{k+1} &= \mathbf{s}^k + \Delta\mathbf{s}^k \end{aligned}$$

Okolia

Keby sme v algoritnoch išli presne po centrálnej ceste, čiže keby sme našli presné riešenie úlohy Sys^μ , potom by sme μ zmenšili a tak by sme pokračovali, bol by čas výpočtu neúmerne dlhý. Preto sa uspokojíme s približným riešením Sys^μ a čo najskôr μ zmenšíme. Na to však potrebujeme, aby boli približné riešenia v istom okolí centrálnej cesty. Budeme používať dva typy okolí, a to $\mathfrak{N}_2(\beta)$ a $\mathfrak{N}_{-\infty}(\beta)$, kde $\beta \in \langle 0, 1 \rangle$.

Začneme s okolím $\mathfrak{N}_2(\beta)$. Toto okolie je založené na klasickej Euklidovskej metrike a meria vzdialenosť (v skutočnosti pôjde o pomer, pozri definíciu $\mathfrak{N}_2(\beta)$ nižšie) bodu $(x_1 s_1, \dots, x_n s_n)$ od priamky $t(1, \dots, 1)$, $t \in \mathbb{R}$, prechádzajúcej bodom $\mathbf{0}$.



Obrázok 6

Ktorý bod priamky $t(1, \dots, 1)$ je najbližšie k bodu $(x_1 s_1, \dots, x_n s_n)$? Ten, pre ktorý vektorov $(x_1 s_1 - t, \dots, x_n s_n - t)$ a $(1, \dots, 1)$ zvierajú pravý uhol. Pre $n = 2$ je situácia znázornená na Obrázku 6. Kedže skalárny súčin navzájom kolmých vektorov je 0, dostávame

$$(x_1 s_1 - t) + \dots + (x_n s_n - t) = 0,$$

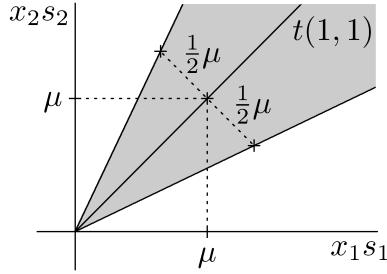
z čoho

$$t = \frac{x_1 s_1 + \dots + x_n s_n}{n} = \frac{\mathbf{x}^T \cdot \mathbf{s}}{n}.$$

Označme takto získané t symbolom μ , čiže $\mu = \frac{\mathbf{x}^T \cdot \mathbf{s}}{n}$. Potom vzdialenosť bodu $(x_1 s_1, \dots, x_n s_n)$ od $\mu(1, \dots, 1)$ je $\|(x_1 s_1 - \mu), \dots, (x_n s_n - \mu)\|$ a okolie $\mathfrak{N}_2(\beta)$ definujeme

$$\mathfrak{N}_2(\beta) = \{(\mathbf{x}, \mathbf{y}, \mathbf{s}) \in \mathcal{F}^o; \|(x_1 s_1 - \mu), \dots, (x_n s_n - \mu)\| \leq \beta \frac{\mathbf{x}^T \cdot \mathbf{s}}{n}\}.$$

Pre $\beta = \frac{1}{2}$ a $n = 2$ je $\mathfrak{N}_2(\frac{1}{2})$ okolie znázornené na Obrázku 7.

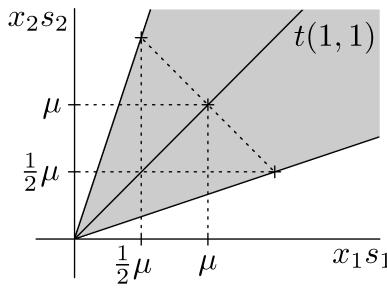


Obrázok 7

Okolie $\mathfrak{N}_{-\infty}(\beta)$ definujeme

$$\mathfrak{N}_{-\infty}(\beta) = \{(\mathbf{x}, \mathbf{y}, \mathbf{s}) \in \mathcal{F}^o; x_i s_i \geq \beta \frac{\mathbf{x}^T \cdot \mathbf{s}}{n} \text{ pre } i = 1, \dots, n\}.$$

Čiže každá zložka $x_i s_i$ musí byť aspoň β -násobok aritmetického priemeru všetkých zložiek vektora $(x_1 s_1, \dots, x_n s_n)$. Pre $\beta = \frac{1}{2}$ a $n = 2$ je $\mathfrak{N}_{-\infty}(\frac{1}{2})$ okolie znázornené na Obrázku 8. (Všimnime si, že aritmetický priemer súradníc rovný μ majú všetky body nadroviny, ktorá prechádza bodom (μ, \dots, μ) a je kolmá na $t(1, \dots, 1)$.)



Obrázok 8

Platí, že $\mathfrak{N}_2(0)$ aj $\mathfrak{N}_{-\infty}(1)$ je práve centrálna cesta. Ak vzrástie β , tak $\mathfrak{N}_2(\beta)$ sa zväčší, naopak $\mathfrak{N}_{-\infty}(\beta)$ sa zmenší. Pre $\beta \in \langle 0, 1 \rangle$ platí, že $\mathfrak{N}_2(\beta) \subseteq \mathfrak{N}_{-\infty}(1 - \beta)$.

Short-step path-following algoritmus

Teraz prejdeme na algoritmy. Začneme so Short-step path-following algoritmom, ktorého význam je skôr teoretický. Tento algoritmus nie je príliš rýchly. Ako napovedá jeho názov, ide po veľmi krátkych krokoch. Je však dobre preskúmaný a je základom pre algoritmy uvádzané v ďalšej kapitole.

Short-step algoritmus, podobne ako ďalšie algoritmy uvádzané v nasledujúcej kapitole, je založený na Newtonovej metóde, čiže budeme vychádzať zo vzťahu (4), avšak už po prvej iterácii Newtonovho algoritmu zmenšíme μ . Keďže algoritmus bude začínať v prípustnom bode $(\mathbf{x}^0, \mathbf{y}^0, \mathbf{s}^0)$, čiže bude platiť $\mathbb{A}^T \cdot \mathbf{y}^0 + \mathbf{s}^0 = \mathbf{c}$ a $\mathbb{A} \cdot \mathbf{x}^0 = \mathbf{b}$, tak aj všetky ďalšie body $(\mathbf{x}^k, \mathbf{y}^k, \mathbf{s}^k)$ budú prípustné. Teda pre tieto body platí $\mathbb{A}^T \cdot \mathbf{y}^k + \mathbf{s}^k - \mathbf{c} = \mathbf{0}$ a $\mathbb{A} \cdot \mathbf{x}^k - \mathbf{b} = \mathbf{0}$.

ALGORITMUS: Short-step path-following algoritmus.

Vstup: $(\mathbf{x}^0, \mathbf{y}^0, \mathbf{s}^0) \in \mathfrak{N}_2(\frac{1}{2})$ a presnosť $\varepsilon > 0$.

Begin

$$(\mathbf{x}, \mathbf{y}, \mathbf{s}) = (\mathbf{x}^0, \mathbf{y}^0, \mathbf{s}^0); \quad \tau = \frac{\mathbf{x}^T \cdot \mathbf{s}}{n}; \quad \sigma = \left(1 - \frac{2}{5\sqrt{n}}\right); \quad \{ \text{ inicializácia } \}$$

While $\tau > \varepsilon$ Do Begin

$$\text{Vyrieš } \begin{pmatrix} \mathbb{0} & \mathbb{A}^T & \mathbb{I} \\ \mathbb{A} & \mathbb{0} & \mathbb{0} \\ \mathbb{S} & \mathbb{0} & \mathbb{X} \end{pmatrix} \cdot \begin{pmatrix} \Delta \mathbf{x} \\ \Delta \mathbf{y} \\ \Delta \mathbf{s} \end{pmatrix} = - \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ x_1 s_1 - \tau \\ \vdots \\ x_n s_n - \tau \end{pmatrix};$$

$$\text{Polož } \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{s} \end{pmatrix} = \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{s} \end{pmatrix} + \begin{pmatrix} \Delta \mathbf{x} \\ \Delta \mathbf{y} \\ \Delta \mathbf{s} \end{pmatrix};$$

$$\tau = \tau \cdot \sigma; \quad (\text{môže byť aj } \tau = \frac{\mathbf{x}^T \cdot \mathbf{s}}{n} \cdot \sigma)$$

End;

End.

Algoritmus skončí, keď $\tau \leq \varepsilon$. Výstupom teda nie je presné, ale iba približné riešenie sústavy Sys^0 . Preto volíme ε malé.

Pri voľbe σ a τ ako v algoritme sú všetky body $(\mathbf{x}, \mathbf{y}, \mathbf{s})$ v $\mathfrak{N}_2(\frac{1}{2})$. Všimnime si koeficient σ vo While cykle. Bez neho by sme riešili sústavu Sys^μ pre hodnotu $\mu = \tau$. My však chceme vyriešiť Sys^0 , preto je tam tento koeficient, ktorý nazývame **centralizačný parameter**. Toto tiež objasňuje definície okolí.

PRÍKLAD. Vyriešte úlohu z Príkladu 1 Short-step path-following algoritmom.

RIEŠENIE. Opíšeme inicializáciu algoritmu, jednotlivé iterácie prenecháme na počítač. Zvoľme si $\mathbf{x}^0 = (0,3; 0,2; 0,5)^T$, $\mathbf{s}^0 = (3, 6, 2)^T$ a $\mathbf{y}^0 = (-5)$. Keďže

$$\begin{aligned} -5 + 3 &+ 2 = 0 \\ -5 &+ 6 - 1 = 0 \\ -5 &+ 2 + 3 = 0 \\ 0,3 + 0,2 + 0,5 - 1 &= 0 \end{aligned}$$

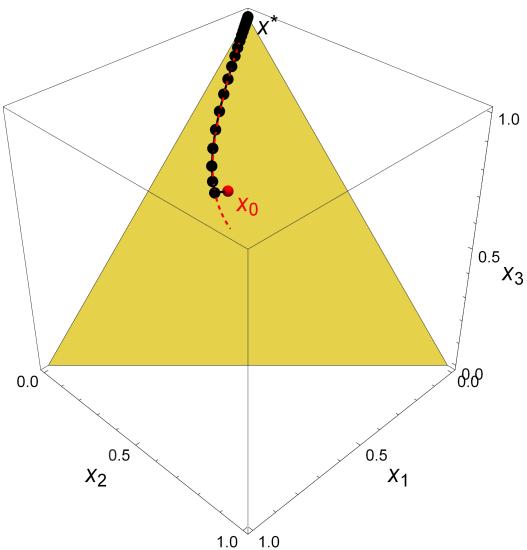
čo znamená, že $\mathbb{A}^T \cdot \mathbf{y} + \mathbf{s} - \mathbf{c} = \mathbf{0}$ a $\mathbb{A} \cdot \mathbf{x} - \mathbf{b} = \mathbf{0}$, tak $(\mathbf{x}^0, \mathbf{y}^0, \mathbf{s}^0) \in \mathcal{F}^o$.

Dalej $x_1^0 s_1^0 = 0,9$, $x_2^0 s_2^0 = 1,2$, $x_3^0 s_3^0 = 1$ a $\frac{\mathbf{x}^T \cdot \mathbf{s}}{3} = \frac{0,9+1,2+1}{3} = 1,0\overline{3}$. Keďže

$$\|(0,9 - 1,0\overline{3}; 1,2 - 1,0\overline{3}; 1 - 1,0\overline{3})\| \ll \frac{1}{2} \cdot 1,0\overline{3},$$

tak $(\mathbf{x}^0, \mathbf{y}^0, \mathbf{s}^0) \in \mathfrak{N}_2(\frac{1}{2})$.

Úlohu sme riešili systémom Mathematica a jednotlivé iterácie sú znázornené na Obrázku 9. Pre $\varepsilon = 10^{-6}$ bolo potrebných 57 iterácií.



Obrázok 9

Cvičenia

CVIČENIE 2.1. Nájdite centrálnu cestu pre úlohu z Cvičenia 1.1. Postupne skonštruujte a vykreslite

1. Bariérovú funkciu, ktorú vykreslíte pomocou Plot3D.
2. Modifikovanú účelovú funkciu, ktorú opäť vykreslíte pomocou Plot3D. Pomocou Manipulate sledujte, ako sa funkcia správa pri zmene μ .
3. Prípustnú množinu, izočiary modifikovanej účelovej funkcie a centrálnu cestu. Izochiary vykreslite pomocou ContourPlot. Pomocou Manipulate sledujte, ako sa izochiary a minimum $\mathbf{x}^*(\mu)$ správajú pri zmene μ , pozri Obrázok 3.

CVIČENIE 2.2. Pomocou Short-step algoritmu vyriešte úlohu z Príkladu 1. Za štartovací bod zvolte $\mathbf{x}^0 = (0,3; 0,2; 0,5)^T$, $\mathbf{s}^0 = (3, 6, 2)^T$, $\mathbf{y}^0 = (-5)$ a presnosť $\varepsilon = 10^{-6}$. Zadanie bude obsahovať tieto časti

1. Fungujúci algoritmus.
2. Vizualizáciu. Zobrazte prípustnú množinu a vykreslite kroky algoritmu, pričom \mathbf{x}^0 bude zobrazený červenou farbou a ďalšie body čiernou farbou.

CVIČENIE 2.3. Upravte zadanie z Cvičenia 2.2.

1. Doplňte zobrazenie centrálnej cesty.
2. Zameňte v algoritme $\mathbf{x} = \mathbf{x} + \Delta\mathbf{x}$ za $\mathbf{x} = \mathbf{x} + \alpha\Delta\mathbf{x}$. Analogickú zmenu spravte pre \mathbf{s} a \mathbf{y} .
3. Meňte parametre τ^0 , σ a α , pričom $\tau^0 \in [0, 3]$, $\sigma \in [0, 1]$ a $\alpha \in [0, 2]$. Vysvetlite zmeny v správaní sa algoritmu. Argumentujte bariérovou funkciou.

3 Algoritmy založené na centrálnej ceste

Metódy vnútorného bodu sa v zásade delia na tri typy algoritmov. Patria sem metódy znižovania potenciálu, metódy affiných transformácií a metódy založené na sledovaní centrálnej cesty. My sa zaobráme iba posledne menovanou skupinou. V tejto kapitole si opíšeme ďalšie tri algoritmy založené na sledovaní centrálnej cesty. Znamená to, že budeme opäť riešiť sústavu (4). Všetky tri algoritmy uvádzané v tejto kapitole sú variáciou Short-step path-following algoritmu, sú však podstatne rýchlejšie.

Mizuno-Todd-Ye predictor-corrector algoritmus

Prvým algoritmom bude algoritmus Mizuno-Todd-Ye. Iteračný krok tohto algoritmu sa skladá z dvoch častí. V prvej časti sa posunie k $\mathbf{0}$ rýchlejšie, aj za cenu toho, že môže „uletiet“ do širšieho okolia \mathfrak{N}_2 . V druhej časti sa vráti bližšie k centrálnej ceste, ale nemusí sa priblížiť k $\mathbf{0}$.

ALGORITMUS: Mizuno-Todd-Ye predictor-corrector algoritmus.

Vstup: $(\mathbf{x}^0, \mathbf{y}^0, \mathbf{s}^0) \in \mathfrak{N}_2(\beta)$, presnosť $\varepsilon > 0$ a $0 < \beta < \bar{\beta} < 1$.

Begin

$$(\mathbf{x}, \mathbf{y}, \mathbf{s}) = (\mathbf{x}^0, \mathbf{y}^0, \mathbf{s}^0); \quad \tau = \frac{\mathbf{x}^T \cdot \mathbf{s}}{n}; \quad \{ \text{ inicializácia } \}$$

While $\tau > \varepsilon$ Do Begin

PREDICTOR STEP ($\sigma = 0$)

$$\text{Vyrieš } \begin{pmatrix} 0 & \mathbf{A}^T & \mathbb{I} \\ \mathbf{A} & 0 & 0 \\ \mathbb{S} & 0 & \mathbb{X} \end{pmatrix} \cdot \begin{pmatrix} \Delta \mathbf{x} \\ \Delta \mathbf{y} \\ \Delta \mathbf{s} \end{pmatrix} = - \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ x_1 s_1 \\ \vdots \\ x_n s_n \end{pmatrix};$$

Zvol' $\alpha \in (0, 1)$ najväčšie možné tak, aby

$$\begin{pmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{s} \end{pmatrix} + \alpha \begin{pmatrix} \Delta \mathbf{x} \\ \Delta \mathbf{y} \\ \Delta \mathbf{s} \end{pmatrix} \in \mathfrak{N}_2(\bar{\beta});$$

$$\text{Polož } \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{s} \end{pmatrix} = \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{s} \end{pmatrix} + \alpha \begin{pmatrix} \Delta \mathbf{x} \\ \Delta \mathbf{y} \\ \Delta \mathbf{s} \end{pmatrix} \quad \text{a} \quad \tau = (1 - \alpha)\tau;$$

CORRECTOR STEP ($\sigma = 1$)

$$\text{Vyrieš } \begin{pmatrix} 0 & A^T & I \\ A & 0 & 0 \\ S & 0 & X \end{pmatrix} \cdot \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta s \end{pmatrix} = - \begin{pmatrix} 0 \\ 0 \\ x_1 s_1 - \tau \\ \vdots \\ x_n s_n - \tau \end{pmatrix};$$

$$\text{Polož } \begin{pmatrix} x \\ y \\ s \end{pmatrix} = \begin{pmatrix} x \\ y \\ s \end{pmatrix} + \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta s \end{pmatrix};$$

End;

End.

Dobré voľby pre β a $\bar{\beta}$ sú $\beta = \frac{1}{4}$ a $\bar{\beta} = \frac{1}{2}$.

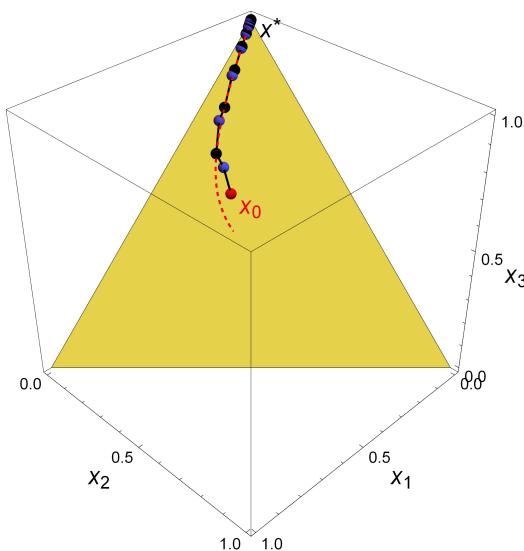
Keď označíme $(x(\alpha), y(\alpha), s(\alpha))^T = (x, y, s)^T + \alpha(\Delta x, \Delta y, \Delta s)^T$, tak v prvom kroku hľadáme maximálne α , pre ktoré

$$\left\| x_1(\alpha)s_1(\alpha) - \frac{x(\alpha)^T \cdot s(\alpha)}{n}, \dots, x_n(\alpha)s_n(\alpha) - \frac{x(\alpha)^T \cdot s(\alpha)}{n} \right\| < \bar{\beta} \cdot \frac{x(\alpha)^T \cdot s(\alpha)}{n} .$$

Toto je sama o sebe optimalizačná úloha. Aby sme ju nemuseli riešiť, uvedieme, že pre $\beta = \frac{1}{4}$ a $\bar{\beta} = \frac{1}{2}$ stačí zvolať

$$\alpha = \min \left\{ \frac{1}{2}, \sqrt{\frac{\tau}{8 \cdot \|(\Delta x_1 \cdot \Delta s_1, \dots, \Delta x_n \cdot \Delta s_n)\|}} \right\} .$$

PRÍKLAD. Vyriešte úlohu z Príkladu 1 Mizuno-Todd-Ye predictor-corrector algoritmom.



Obrázok 10

RIEŠENIE. Opäť si zvolíme $\mathbf{x}^0 = (0,3; 0,2; 0,5)^T$, $\mathbf{s}^0 = (3, 6, 2)^T$ a $\mathbf{y}^0 = (-5)$.

Už vieme, že $(\mathbf{x}^0, \mathbf{y}^0, \mathbf{s}^0) \in \mathcal{F}^o$. Platí tiež $(\mathbf{x}^0, \mathbf{y}^0, \mathbf{s}^0) \in \mathfrak{N}_2(\frac{1}{4})$ (pozri príklad demonštrujúci Short-step path-following algoritmus v predchádzajúcej kapitole).

Úlohu sme riešili systémom Mathematica a jednotlivé iterácie sú znázornené na Obrázku 10. Pre $\varepsilon = 10^{-6}$ bolo potrebných 21 iterácií algoritmu. (Pripomeňme, že Short-step algoritmus potreboval pre presnosť 10^{-6} až 57 iterácií.) Body získané pomocou Predictor step sú zobrazené modrou a body získané pomocou Corrector step sú zobrazené čiernou farbou.

Long-step path-following algoritmus

V tomto algoritme sa jednotlivé iterácie nachádzajú v $\mathfrak{N}_{-\infty}$ okolí, ktoré je zvyčajne väčšie ako \mathfrak{N}_2 okolie. Naviac, na rozdiel od Short-step algoritmu, v ktorom je centralizačný parameter σ blízko 1, tu mu dáme väčšiu voľu. Znamená to, že algoritmus sa nemusí držať tak blízko centrálnej cesty ako Short-step algoritmus. Vďaka tomu je konvergencia Long-step algoritmu vo všeobecnosti rýchlejšia.

ALGORITMUS: Long-step path-following algoritmus.

Vstup: $(\mathbf{x}^0, \mathbf{y}^0, \mathbf{s}^0) \in \mathfrak{N}_{-\infty}(\beta)$, presnosť $\varepsilon > 0$, $0 < \beta < 1$ a $0 < \sigma_{\min} < \sigma_{\max} < 1$.

Begin

$$(\mathbf{x}, \mathbf{y}, \mathbf{s}) = (\mathbf{x}^0, \mathbf{y}^0, \mathbf{s}^0); \quad \tau = \frac{\mathbf{x}^T \cdot \mathbf{s}}{n}; \quad \{ \text{ inicializácia } \}$$

While $\tau > \varepsilon$ Do Begin

Zvoľ $\sigma \in (\sigma_{\min}, \sigma_{\max})$;

$$\text{Vyrieš } \begin{pmatrix} 0 & \mathbb{A}^T & \mathbb{I} \\ \mathbb{A} & 0 & 0 \\ \mathbb{S} & 0 & \mathbb{X} \end{pmatrix} \cdot \begin{pmatrix} \Delta \mathbf{x} \\ \Delta \mathbf{y} \\ \Delta \mathbf{s} \end{pmatrix} = - \begin{pmatrix} 0 \\ 0 \\ x_1 s_1 - \sigma \tau \\ \vdots \\ x_n s_n - \sigma \tau \end{pmatrix};$$

Zvoľ $\alpha \in (0, 1)$ najväčšie možné tak, aby

$$\begin{pmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{s} \end{pmatrix} + \alpha \begin{pmatrix} \Delta \mathbf{x} \\ \Delta \mathbf{y} \\ \Delta \mathbf{s} \end{pmatrix} \in \mathfrak{N}_{-\infty}(\beta);$$

$$\text{Polož } \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{s} \end{pmatrix} = \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{s} \end{pmatrix} + \alpha \begin{pmatrix} \Delta \mathbf{x} \\ \Delta \mathbf{y} \\ \Delta \mathbf{s} \end{pmatrix} \quad \text{a} \quad \tau = (1 - \alpha(1 - \sigma))\tau;$$

End;

End.

Tak ako v Mizuno-Todd-Ye algoritme, aj tu riešime v iteračnom kroku optimizačnú úlohu. Ak sa jej chceme vyhnúť, volíme

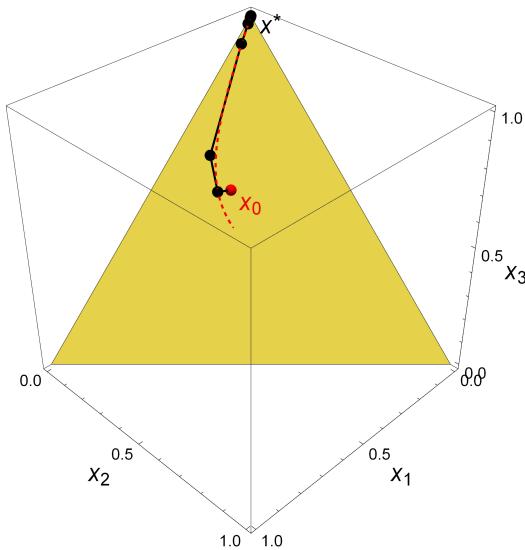
$$\alpha = \sqrt{8} \cdot \beta \cdot \sigma \cdot \frac{1 - \beta}{n(1 + \beta)} .$$

PRÍKLAD. Vyriešte úlohu z Príkladu 1 Long-step path-following algoritmom.

RIEŠENIE. Znova zvolíme $\mathbf{x}^0 = (0,3; 0,2; 0,5)^T$, $\mathbf{s}^0 = (3, 6, 2)^T$ a $\mathbf{y}^0 = (-5)$.

Už vieme, že platí $(\mathbf{x}^0, \mathbf{y}^0, \mathbf{s}^0) \in \mathcal{F}^o$. Kedže $\mathfrak{N}_2(\frac{1}{2}) \subseteq \mathfrak{N}_{-\infty}(\frac{1}{2})$, tak platí aj $(\mathbf{x}^0, \mathbf{y}^0, \mathbf{s}^0) \in \mathfrak{N}_{-\infty}(\frac{1}{2})$.

Úlohu sme riešili systémom Mathematica, pričom sme volili $\beta = 0,1$ a $\sigma = 0,5$. Jednotlivé iterácie sú znázornené na Obrázku 11. Pre $\varepsilon = 10^{-6}$ bolo potrebných 11 iterácií algoritmu.



Obrázok 11

Mehrotra predictor-corrector algoritmus

Posledným algoritmom, ktorý uvádzame, je Mehrotrov algoritmus. Tento algoritmus nepotrebuje začať v prípustnom bode $(\mathbf{x}^0, \mathbf{y}^0, \mathbf{s}^0) \in \mathcal{F}^o$, stačí aby platilo $\mathbf{x}^0, \mathbf{s}^0 > \mathbf{0}$. Problém je v tom, že keď algoritmus začne vo „veľmi zlom“ bode, tak nekonverguje. Naopak, keď začne v „dobrom“ bode, konverguje zvyčajne veľmi rýchlo. Tento algoritmus je oblúbený, pretože nájst bod z \mathcal{F}^o je pomerne náročné. V Mehrotrovom algoritme \mathbf{r}_D , \mathbf{r}_P a \mathbf{r}_C označujú takzvané reziduá duálnej úlohy, primárnej úlohy a podmienok komplementarity. Tieto reziduá definujeme ako $\mathbf{r}_D = \mathbb{A}^T \cdot \mathbf{y} + \mathbf{s} - \mathbf{c}$, $\mathbf{r}_P = \mathbb{A} \cdot \mathbf{x} - \mathbf{b}$ a $\mathbf{r}_C = (x_1 s_1, \dots, x_n s_n)^T$. Pre maticu $\mathbb{Q} = (q_{i,j})$ jej Euklidovskú normu značíme $\|\mathbb{Q}\| = \sqrt{\sum_i \sum_j q_{i,j}^2}$.

ALGORITMUS: Mehrotra predictor-corrector algoritmus.

Vstup: $(\mathbf{x}^0, \mathbf{y}^0, \mathbf{s}^0)$ s $\mathbf{x}^0, \mathbf{s}^0 > \mathbf{0}$ a presnosť $\varepsilon > 0$.

Begin

$$(\mathbf{x}, \mathbf{y}, \mathbf{s}) = (\mathbf{x}^0, \mathbf{y}^0, \mathbf{s}^0); \quad \tau = \frac{\mathbf{x}^T \cdot \mathbf{s}}{n}; \quad \bar{r} = \frac{\|(r_D, r_P, r_C)\|}{1 + \max\{\|\mathbb{A}\|, \|\mathbf{b}\|, \|\mathbf{c}\|\}}; \quad \{ \text{ inicializácia } \}$$

While ($\tau > \varepsilon$ Or $\bar{r} > \varepsilon$) Do Begin

PREDICTOR STEP ($\sigma = 0$)

$$\text{Vyrieš } \begin{pmatrix} \mathbb{0} & \mathbb{A}^T & \mathbb{I} \\ \mathbb{A} & \mathbb{0} & \mathbb{0} \\ \mathbb{S} & \mathbb{0} & \mathbb{X} \end{pmatrix} \cdot \begin{pmatrix} \Delta \mathbf{x}^N \\ \Delta \mathbf{y}^N \\ \Delta \mathbf{s}^N \end{pmatrix} = - \begin{pmatrix} \mathbb{A}^T \cdot \mathbf{y} + \mathbf{s} - \mathbf{c} \\ \mathbb{A} \cdot \mathbf{x} - \mathbf{b} \\ x_1 s_1 \\ \vdots \\ x_n s_n \end{pmatrix} = - \begin{pmatrix} \mathbf{r}_D \\ \mathbf{r}_P \\ \mathbf{r}_C \end{pmatrix};$$

$$\alpha_P^N = \min \left\{ 1, \min_{\Delta x_i^N > 0} \frac{x_i}{\Delta x_i^N} \right\}; \quad \alpha_D^N = \min \left\{ 1, \min_{\Delta s_i^N > 0} \frac{s_i}{\Delta s_i^N} \right\};$$

Polož $\mathbf{x}^N = \mathbf{x} + \alpha_P^N \Delta \mathbf{x}^N$ a $\mathbf{s}^N = \mathbf{s} + \alpha_D^N \Delta \mathbf{s}^N$;

$$\tau^N = \frac{\mathbf{x}^{N T} \cdot \mathbf{s}^N}{n}; \quad \sigma = \left(\frac{\tau^N}{\tau} \right)^3;$$

CORECTOR STEP

$$\text{Vyrieš } \begin{pmatrix} \mathbb{0} & \mathbb{A}^T & \mathbb{I} \\ \mathbb{A} & \mathbb{0} & \mathbb{0} \\ \mathbb{S} & \mathbb{0} & \mathbb{X} \end{pmatrix} \cdot \begin{pmatrix} \Delta \mathbf{x} \\ \Delta \mathbf{y} \\ \Delta \mathbf{s} \end{pmatrix} = - \begin{pmatrix} \mathbb{A}^T \cdot \mathbf{y} + \mathbf{s} - \mathbf{c} \\ \mathbb{A} \cdot \mathbf{x} - \mathbf{b} \\ x_1 s_1 - \sigma \tau \\ \vdots \\ x_n s_n - \sigma \tau \end{pmatrix} - \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \Delta x_1^N \cdot \Delta s_1^N \\ \vdots \\ \Delta x_n^N \cdot \Delta s_n^N \end{pmatrix};$$

$$\alpha_P = \min \left\{ 1, \nu \cdot \min_{\Delta x_i > 0} \frac{x_i}{\Delta x_i} \right\}; \quad \alpha_D = \min \left\{ 1, \nu \cdot \min_{\Delta s_i > 0} \frac{s_i}{\Delta s_i} \right\};$$

Polož $\mathbf{x} = \mathbf{x} + \alpha_P \Delta \mathbf{x}$, $\mathbf{y} = \mathbf{y} + \alpha_D \Delta \mathbf{y}$, a $\mathbf{s} = \mathbf{s} + \alpha_D \Delta \mathbf{s}$;

$$\tau = \frac{\mathbf{x}^T \cdot \mathbf{s}}{n}; \quad \bar{r} = \frac{\|(\mathbf{r}_D, \mathbf{r}_P, \mathbf{r}_C)\|}{1 + \max\{\|\mathbb{A}\|, \|\mathbf{b}\|, \|\mathbf{c}\|\}};$$

End;

End.

Pre ν musí platiť $\nu \in (0, 1)$, pričom ho volíme blízko 1. Povedzme $\nu = 0,99$.

Ako však zvoliť „dobrý“ počiatočný bod $(\mathbf{x}^0, \mathbf{y}^0, \mathbf{s}^0)$? Ukážeme si jednu takúto voľbu. Bod $(\mathbf{x}^0, \mathbf{y}^0, \mathbf{s}^0)$ nájdeme na tri kroky.

Najprv položíme

$$\tilde{\mathbf{x}} = \mathbb{A}^T \cdot (\mathbb{A} \cdot \mathbb{A}^T)^{-1} \cdot \mathbf{b}, \quad \tilde{\mathbf{y}} = (\mathbb{A} \cdot \mathbb{A}^T)^{-1} \cdot \mathbf{b}, \quad \tilde{\mathbf{s}} = \mathbf{c} - \mathbb{A}^T \cdot \tilde{\mathbf{y}}.$$

Teraz sú pre $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}, \tilde{\mathbf{s}})$ splnené ohraničenia $\mathbb{A} \cdot \tilde{\mathbf{x}} = \mathbf{b}$ a $\mathbb{A}^T \cdot \tilde{\mathbf{y}} + \tilde{\mathbf{s}} = \mathbf{c}$, no vektoru $\tilde{\mathbf{x}}$ a $\tilde{\mathbf{s}}$ môžu mať negatívne zložky. Preto označíme

$$\delta_x = \max\left\{-\frac{3}{2} \min_{1 \leq i \leq n} \tilde{x}_i, 0\right\} \quad \text{a} \quad \delta_s = \max\left\{-\frac{3}{2} \min_{1 \leq i \leq n} \tilde{s}_i, 0\right\}$$

a kladieme

$$\bar{\mathbf{x}} = \tilde{\mathbf{x}} + \delta_x (1, \dots, 1)^T \quad \text{a} \quad \bar{\mathbf{s}} = \tilde{\mathbf{s}} + \delta_s (1, \dots, 1)^T.$$

Táto voľba zaručuje $\bar{\mathbf{x}}, \bar{\mathbf{s}} \geq \mathbf{0}$, hoci ohraničenia $\mathbb{A} \cdot \bar{\mathbf{x}} = \mathbf{b}$ a $\mathbb{A}^T \cdot \tilde{\mathbf{y}} + \bar{\mathbf{s}} = \mathbf{c}$ už splnené nie sú. My však potrebujeme vektoru ostro väčšie od $\mathbf{0}$. Preto označíme

$$\bar{\delta}_x = \frac{1}{2} \cdot \frac{1 + \bar{\mathbf{x}}^T \cdot \bar{\mathbf{s}}}{1 + (1, \dots, 1) \cdot \bar{\mathbf{s}}} \quad \text{a} \quad \bar{\delta}_s = \frac{1}{2} \cdot \frac{1 + \bar{\mathbf{x}}^T \cdot \bar{\mathbf{s}}}{1 + (1, \dots, 1) \cdot \bar{\mathbf{x}}}$$

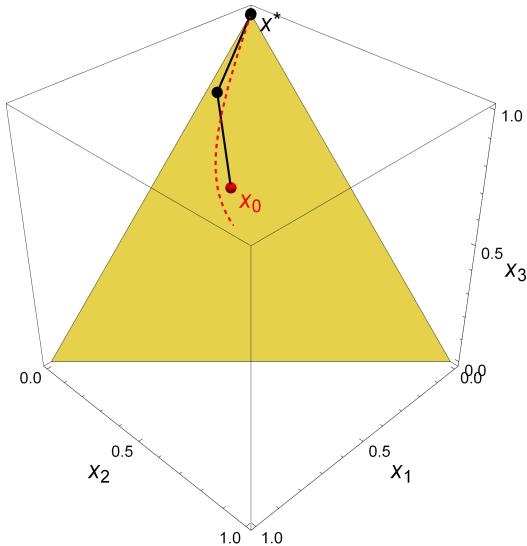
a položíme

$$\mathbf{x}^0 = \bar{\mathbf{x}} + \bar{\delta}_x (1, \dots, 1)^T, \quad \mathbf{s}^0 = \bar{\mathbf{s}} + \bar{\delta}_s (1, \dots, 1)^T, \quad \text{a} \quad \mathbf{y}^0 = \tilde{\mathbf{y}}.$$

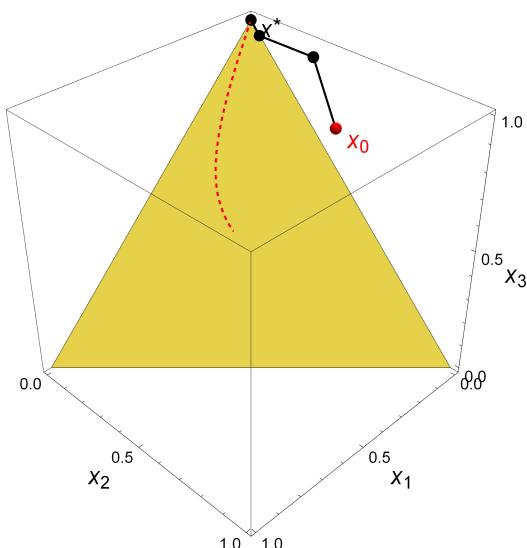
PRÍKLAD. Vyriešte úlohu z Príkladu 1 Mehrotra predictor-corrector algoritmom.

RIEŠENIE. Úlohu sme riešili systémom Mathematica a presnosť sme opäť volili $\varepsilon = 10^{-6}$. Keď sme algoritmus spustili v prípustnom bode $\mathbf{x}^0 = (0,3; 0,2; 0,5)^T$, $\mathbf{s}^0 = (3, 6, 2)^T$ a $\mathbf{y}^0 = (-5)$, tak boli potrebné iba 4 iterácie, pozri Obrázok 12.

Pri štarte v neprípustnom bode $\mathbf{x}^0 = (0,1; 0,5; 0,8)^T$, $\mathbf{s}^0 = (1, 1, 1)^T$ a $\mathbf{y}^0 = (0)$ bolo potrebných 5 iterácií, pozri Obrázok 13.



Obrázok 12



Obrázok 13

Volba bodu z \mathcal{F}^o

Pri Short-step, Mizuno-Todd-Ye a Long-step algoritmoch potrebujeme začať vo vnútornom prípustnom bode $(\mathbf{x}^0, \mathbf{y}^0, \mathbf{s}^0) \in \mathcal{F}^o$. V tejto časti ukážeme, ako to zabezpečiť. (Neukážeme však, ako zabezpečiť aby $(\mathbf{x}^0, \mathbf{y}^0, \mathbf{s}^0) \in \mathfrak{N}_2(\beta)$, respektíve $(\mathbf{x}^0, \mathbf{y}^0, \mathbf{s}^0) \in \mathfrak{N}_{-\infty}(\beta)$.) V skutočnosti nenájdeme vnútorný bod $(\mathbf{x}^0, \mathbf{y}^0, \mathbf{s}^0) \in \mathcal{F}^o$, ale budeme riešiť úplne inú úlohu, ktorej zodpovedajúci vnútorný bod z \mathcal{F}_N^o nájdeme triviálne.

Zvolíme $(\mathbf{x}^0, \mathbf{y}^0, \mathbf{s}^0) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n$ také, že $\mathbf{x}^0, \mathbf{s}^0 > \mathbf{0}$ a zostrojíme úlohy (premenné sú $\mathbf{x}, x_{n+1}, x_{n+2}, \mathbf{y}, y_{m+1}$ a $\mathbf{s}, s_{n+1}, s_{n+2}$)

$$\tilde{P}: \quad \begin{array}{c} \min \mathbf{c}^T \cdot \mathbf{x} + \rho_1 \cdot x_{n+1} \\ \hline \mathbb{A} \cdot \mathbf{x} + (\mathbf{b} - \mathbb{A} \cdot \mathbf{x}^0) \cdot x_{n+1} = \mathbf{b} \\ (\mathbb{A}^T \cdot \mathbf{y}^0 + \mathbf{s}^0 - \mathbf{c}) \cdot \mathbf{x} + x_{n+2} = \rho_2 \\ \hline (\mathbf{x}^T, x_{n+1}, x_{n+2})^T \geq \mathbf{0} \end{array}$$

a

$$\tilde{D}: \quad \begin{array}{c} \max \mathbf{b}^T \cdot \mathbf{y} + \rho_2 \cdot y_{m+1} \\ \hline \mathbb{A}^T \cdot \mathbf{y} + (\mathbb{A}^T \cdot \mathbf{y}^0 + \mathbf{s}^0 - \mathbf{c}) \cdot y_{m+1} + \mathbf{s} = \mathbf{c} \\ (\mathbf{b} - \mathbb{A} \cdot \mathbf{x}^0) \cdot \mathbf{y} + s_{n+1} = \rho_1 \\ y_{m+1} + s_{n+2} = 0 \\ \hline (\mathbf{s}^T, s_{n+1}, s_{n+2})^T \geq \mathbf{0} \end{array}$$

kde $\rho_2 > (\mathbb{A}^T \cdot \mathbf{y}^0 + \mathbf{s}^0 - \mathbf{c}) \cdot \mathbf{x}^0$ a $\rho_1 > (\mathbf{b} - \mathbb{A} \cdot \mathbf{x}^0) \cdot \mathbf{y}^0$. Všimnime si, že pre primárnu úlohu \tilde{P} je \tilde{D} k nej prislúchajúca upravená duálna úloha (tak ako sme to zaviedli v prvej kapitole). Pre voľbu

$$\begin{array}{lll} x_{n+1}^0 = 1 & y_{m+1}^0 = -1 & s_{n+1}^0 = \rho_1 - (\mathbf{b} - \mathbb{A} \cdot \mathbf{x}^0) \cdot \mathbf{y}^0 \\ x_{n+2}^0 = \rho_2 - (\mathbb{A}^T \cdot \mathbf{y}^0 + \mathbf{s}^0 - \mathbf{c}) \cdot \mathbf{x}^0 & & s_{n+2}^0 = 1 \end{array}$$

platí, že $(\mathbf{x}^0, x_{n+1}, x_{n+2}, \mathbf{y}^0, y_{m+1}, \mathbf{s}^0, s_{n+1}, s_{n+2}) \in \mathcal{F}_N^o$, čiže tento vektor (ktorý sme uviedli trochu nepresne, pretože sme kvôli problémom so zdvojeným horným indexom netransponovali \mathbf{x}^0 , \mathbf{y}^0 a \mathbf{s}^0) je prípustným bodom, ktorý splňa $x_i, s_i > 0$ pre $1 \leq i \leq n+2$. Keďže úlohy \tilde{P} a \tilde{D} majú tvar, ktorý požadujeme, môžeme začať algoritmus nie pre P , ale pre \tilde{P} . (Samozrejme, β treba spočítať.) Platí nasledujúce tvrdenie

VETA. Ak pre optimálne riešenie \mathbf{x}^* úlohy P a optimálne riešenie $(\mathbf{y}^*, \mathbf{s}^*)$ úlohy D platí $\rho_2 > (\mathbb{A}^T \cdot \mathbf{y}^0 + \mathbf{s}^0 - \mathbf{c}) \cdot \mathbf{x}^*$ a $\rho_1 > (\mathbf{b} - \mathbb{A} \cdot \mathbf{x}^0) \cdot \mathbf{y}^*$, tak riešenie \tilde{P} je riešením P s $x_{n+1}^* = 0$ a riešenie \tilde{D} je riešením D s $y_{m+1}^* = 0$.

Riešenie systému rovníc pre $(\Delta \mathbf{x}, \Delta \mathbf{y}, \Delta \mathbf{s})^T$

Pri všetkých metódach vnútorného bodu potrebujeme riešiť systém $2n+m$ rovníc o $2n+m$ neznámych. Riešenie tohto systému sa dá urýchliť tak, že v skutočnosti budeme riešiť iba systém m rovníc o m neznámych.

Podľa vzťahu (4) máme riešiť systém

$$\begin{pmatrix} 0 & \mathbb{A}^T & \mathbb{I} \\ \mathbb{A} & 0 & 0 \\ \mathbb{S} & 0 & \mathbb{X} \end{pmatrix} \cdot \begin{pmatrix} \Delta \mathbf{x} \\ \Delta \mathbf{y} \\ \Delta \mathbf{s} \end{pmatrix} = - \begin{pmatrix} \mathbf{r}_D \\ \mathbf{r}_P \\ \mathbf{r}_C \end{pmatrix},$$

ktorý sa vyskytuje v našich algoritmoch. Poznamenajme, že \mathbf{r}_D , \mathbf{r}_P a \mathbf{r}_C nie sú nutne reziduá definované v kroku Predictor step v Mehrotrovom algoritme, sú to jednoducho pravé strany rovníc. Tiež si všimnime, že matica \mathbb{S} je diagonálna, čiže

$$\mathbb{S}^{-1} = \begin{pmatrix} s_1^{-1} & & 0 \\ & \ddots & \\ 0 & & s_n^{-1} \end{pmatrix}.$$

Kedľ sústavu pre $(\Delta \mathbf{x}, \Delta \mathbf{y}, \Delta \mathbf{s})^T$ rozpíšeme, dostávame rovnice

$$\mathbb{A}^T \cdot \Delta \mathbf{y} + \Delta \mathbf{s} = -\mathbf{r}_D \quad (a)$$

$$\mathbb{A} \cdot \Delta \mathbf{x} = -\mathbf{r}_P \quad (b)$$

$$\mathbb{S} \cdot \Delta \mathbf{x} + \mathbb{X} \cdot \Delta \mathbf{s} = -\mathbf{r}_C \quad (c)$$

Zameriame sa najprv na $\Delta \mathbf{y}$. Platí

$$\mathbb{A}^T \cdot \Delta \mathbf{y} + \Delta \mathbf{s} = -\mathbf{r}_D \quad / \mathbb{X} \cdot$$

$$\mathbb{X} \cdot \mathbb{A}^T \cdot \Delta \mathbf{y} + \mathbb{X} \cdot \Delta \mathbf{s} = -\mathbb{X} \cdot \mathbf{r}_D \quad / z \ (c)$$

$$\mathbb{X} \cdot \mathbb{A}^T \cdot \Delta \mathbf{y} - \mathbf{r}_C - \mathbb{S} \cdot \Delta \mathbf{x} = -\mathbb{X} \cdot \mathbf{r}_D \quad / \mathbb{A} \cdot \mathbb{S}^{-1}.$$

$$\mathbb{A} \cdot \mathbb{S}^{-1} \cdot \mathbb{X} \cdot \mathbb{A}^T \cdot \Delta \mathbf{y} - \mathbb{A} \cdot \mathbb{S}^{-1} \cdot \mathbf{r}_C - \mathbb{A} \cdot \mathbb{S}^{-1} \cdot \mathbb{S} \cdot \Delta \mathbf{x} = -\mathbb{A} \cdot \mathbb{S}^{-1} \cdot \mathbb{X} \cdot \mathbf{r}_D \quad / z \ (b)$$

$$\mathbb{A} \cdot \mathbb{S}^{-1} \cdot \mathbb{X} \cdot \mathbb{A}^T \cdot \Delta \mathbf{y} - \mathbb{A} \cdot \mathbb{S}^{-1} \cdot \mathbf{r}_C + \mathbf{r}_P = -\mathbb{A} \cdot \mathbb{S}^{-1} \cdot \mathbb{X} \cdot \mathbf{r}_D$$

Z poslednej rovnice dostávame, že $\Delta \mathbf{y}$ je riešením systému

$$(\mathbb{A} \cdot \mathbb{S}^{-1} \cdot \mathbb{X} \cdot \mathbb{A}^T) \cdot \Delta \mathbf{y} = -\mathbf{r}_P + \mathbb{A} \cdot \mathbb{S}^{-1} \cdot (\mathbf{r}_C - \mathbb{X} \cdot \mathbf{r}_D)$$

a následne vypočítame $\Delta \mathbf{s}$ a $\Delta \mathbf{x}$. Dostávame

$$\Delta \mathbf{s} = -\mathbf{r}_D - \mathbb{A}^T \cdot \Delta \mathbf{y}$$

$$\Delta \mathbf{x} = -\mathbb{S}^{-1} \cdot (\mathbf{r}_C + \mathbb{X} \cdot \Delta \mathbf{s})$$

Záverom poznamenajme, že Newtonova metóda funguje dobre, keď sa druhé derivácie menia pomaly. Preto nemôže byť centralizačný parameter σ príliš malý.

Pre záujemcov o hlbšie štúdium, respektíve pre tých, ktorí sa chcú oboznámiť s ďalšími algoritmami lineárneho programovania, odporúčame špecializované monografie [1,12,14,15,16].

Cvičenia

CVIČENIE 3.1. Upravte svoj algoritmus Short-step z Cvičenia 2.3 tak, že podľa prednášky zrýchlite výpočet $\Delta\mathbf{x}$, $\Delta\mathbf{s}$ a $\Delta\mathbf{y}$.

CVIČENIE 3.2. Vyriešte úlohu z Príkladu 1 Mizuno-Todd-Ye predictor-corrector algoritmom. Štartovací bod a presnosť zvoľte ako v Cvičení 2.2. Algoritmus získate jednoduchou úpravou Short-step algoritmu. Riešenie bude obsahovať

1. Fungujúci algoritmus.
2. Výpočet centrálnej cesty. Vytvorte funkciu $xCP[\tau]$, ktorá bude obsahovať \mathbf{x} -ovú časť riešenia sústavy

$$F^\tau(\mathbf{x}, \mathbf{y}, \mathbf{s}) = \begin{pmatrix} \mathbb{A}^T \cdot \mathbf{y} + \mathbf{s} - \mathbf{c} \\ \mathbb{A} \cdot \mathbf{x} - \mathbf{b} \\ x_1 s_1 - \tau \\ \vdots \\ x_n s_n - \tau \end{pmatrix}$$

3. Vizualizáciu. Zobrazte prípustnú množinu a vykreslite kroky algoritmu, pričom \mathbf{x}^0 bude zobrazený červenou farbou, Predictor step kroky budú zobrazené modrou farbou a Corrector step kroky budú čierne. Zobrazte aj centrálnu cestu.

Zamyslite sa, ako algoritmus funguje, čo robí predictorový a čo correctorový krok.

CVIČENIE 3.3. V Mizuno-Todd-Ye predictor-corrector algoritme z Cvičenia 3.2 spravte nasledujúce zmeny.

1. Vypíšte α v každom kroku.
2. Experimentujte s parametrom α , čiže meňte α a pozorujte zmeny v správaní sa algoritmu.
3. Nastavením $\alpha = 0$ respektíve $\tau = 0$ vypínajte predictor, respektíve corrector, a sledujte zmeny v správaní algoritmu.

CVIČENIE 3.4. Vyriešte úlohu z Príkladu 1 Long-step algoritmom. Štartovací bod a presnosť zvoľte ako v Cvičení 2.2. Riešenie bude obsahovať

1. Fungujúci algoritmus.
2. Výpočet centrálnej cesty.
3. Vizualizáciu. Zobrazte prípustnú množinu, vykreslite kroky algoritmu a centrálnu cestu.

CVIČENIE 3.5. Vyriešte úlohu z Príkladu 1 Mehrotra predictor-corrector algoritmom. Štartovací bod a presnosť zvoľte najprv ako v Cvičení 2.2. Keď vám algoritmus bude fungovať, zmeňte štartovací bod na neprípustný bod $\mathbf{x}^0 = (0,1; 0,5; 0,8)^T$, $\mathbf{s}^0 = (1, 1, 1)^T$, $\mathbf{y}^0 = (0)$. Riešenie bude obsahovať

1. Fungujúci algoritmus.
2. Výpočet centrálnej cesty.
3. Vizualizáciu. Zobrazte prípustnú množinu, vykreslite kroky algoritmu a centrálnu cestu.

CVIČENIE 3.6. V softvéri Matlab analyzujte výkon algoritmov:

- Simplexový algoritmus
- Short-step algoritmus
- Mizuno-Todd-Ye predictor-corrector algoritmus
- Long-step algoritmus
- Mehrotra predictor-corrector algoritmus

v závislosti od počtu premenných n . Teda vzájomne porovnajte

1. Úspešnosť algoritmov, teda či algoritmus našiel správne riešenie.
2. Rýchlosť algoritmov z hľadiska počtu iterácií.
3. Rýchlosť algoritmov z hľadiska času výpočtu.

Do predpripraveného súboru potrebujete doplniť

- a) Vygenerovanie úlohy daného rozmeru $[m, n]$.
- b) Vyriešenie úlohy pomocou algoritmov využívajúcich centrálnu cestu.
- c) Zapisovanie počtu iterácií do poľa `iter_**`, výpočtového času do poľa `time_**` a informáciu o tom, či algoritmus zlyhal do `fail_**` (ak zlyhal, `fail_**(n)=1`).

4 Karushove-Kuhnove-Tuckerove podmienky

V tejto a v ďalších kapitolách budeme riešiť všeobecnejšie úlohy matematického programovania. Budeme uvažovať úlohy v tvare

$$\begin{array}{c} \min f(\mathbf{x}) \\ \hline g_i(\mathbf{x}) \leq 0 & 1 \leq i \leq \ell \\ h_j(\mathbf{x}) = 0 & 1 \leq j \leq m \\ \hline \mathbf{x} \in D \end{array} \quad (5)$$

kde $\mathbf{x} \in \mathbb{R}^n$ a $D = D_f \cap D_{g_1} \cap \dots \cap D_{g_\ell} \cap D_{h_1} \cap \dots \cap D_{h_m}$, čiže D je prienik definičných oborov funkcií f , g_i a h_j . Všimnime si, že ohraničenia na nezápornosť premenných už nebude uvažovať zvlášť, budú zahrnuté medzi funkiami $g_i(\mathbf{x})$.

Množina prípustných riešení je

$$\mathcal{F} = \{\mathbf{x} \in D; g_i(\mathbf{x}) \leq 0, h_j(\mathbf{x}) = 0, 1 \leq i \leq \ell, 1 \leq j \leq m\},$$

čiže našu úlohu môžeme formulovať

$$\begin{array}{c} \min f(\mathbf{x}) \\ \hline \mathbf{x} \in \mathcal{F} \end{array}$$

V tejto kapitole si ukážeme nutné podmienky na to, aby bol vo vnútornom bode D lokálny extrém.

Približný tvar funkcie

Najprv pomocou Taylorovho rozvoja prvého rádu odvodíme približný tvar funkcie viacerých premenných. Nech je $q(\mathbf{x})$ funkcia n premenných a nech $\mathbf{x}^0 \in D_q$. Potom

$$\begin{aligned} q(\mathbf{x}) &\approx q(\mathbf{x}^0) + \frac{\partial}{\partial x_1} q(\mathbf{x}^0) \cdot (x_1 - x_1^0) + \dots + \frac{\partial}{\partial x_n} q(\mathbf{x}^0) \cdot (x_n - x_n^0) \\ &= q(\mathbf{x}^0) + \nabla q(\mathbf{x}^0)^T \cdot (\mathbf{x} - \mathbf{x}^0). \end{aligned}$$

Ak označíme $\mathbf{d} = \mathbf{x} - \mathbf{x}^0$, tak pre malé \mathbf{d} platí

$$q(\mathbf{x}^0 + \mathbf{d}) \approx q(\mathbf{x}^0) + \nabla q(\mathbf{x}^0)^T \cdot \mathbf{d} \quad (6)$$

Jedno ohraničenie v tvare rovnice

Začneme s úlohou, ktorá má iba jedno ohraničenie v tvare rovnice.

$$\begin{array}{c} \min f(\mathbf{x}) \\ \hline h(\mathbf{x}) = 0 \\ \hline \mathbf{x} \in \mathbb{R}^n \end{array}$$

Nech $\mathbf{y} \in \mathcal{F}$. Ak v \mathbf{y} nie je extrém, tak pre nejaký smer \mathbf{d} musí platiť

$$\begin{aligned} f(\mathbf{y} + \mathbf{d}) &< f(\mathbf{y}) & \text{a} \\ h(\mathbf{y} + \mathbf{d}) &= 0. \end{aligned} \tag{7}$$

Čiže $\mathbf{y} + \mathbf{d}$ je prípustné riešenie a zároveň v smere \mathbf{d} hodnota funkcie f klesá. Podľa (6) je systém (7) ekvivalentný s

$$\begin{aligned} \nabla f(\mathbf{y})^T \cdot \mathbf{d} &< 0 \\ \nabla h(\mathbf{y})^T \cdot \mathbf{d} &= 0. \end{aligned}$$

Teraz predpokladajme, že sú v prípustnom bode \mathbf{x} vektory $\nabla f(\mathbf{x})$ a $\nabla h(\mathbf{x})$ nenulové. Potom ak existuje \mathbf{d} také, že $\nabla h(\mathbf{x})^T \cdot \mathbf{d} = 0$, čiže ak existuje \mathbf{d} kolmé na $\nabla h(\mathbf{x})$, a zároveň \mathbf{d} nie je kolmé na $\nabla f(\mathbf{x})$, tak buď $\nabla f(\mathbf{x})^T \cdot \mathbf{d} < 0$ a podľa vyššie uvedeného v \mathbf{x} nie je extrém, alebo $\nabla f(\mathbf{x})^T \cdot \mathbf{d} > 0$ a $\nabla h(\mathbf{x})^T \cdot \mathbf{d} = 0$. Lenže v druhom prípade platí $\nabla h(\mathbf{x})^T \cdot (-\mathbf{d}) = 0$ a $\nabla f(\mathbf{x})^T \cdot (-\mathbf{d}) < 0$, čiže opäť v \mathbf{x} podľa vyššie uvedeného nie je extrém. (Všimnime si, že tu využívame, že \mathbf{x} je vnútorným bodom $D = D_f \cap D_h$.)

Preto ak je v \mathbf{x} extrém, tak pre každé \mathbf{d} , pre ktoré $\nabla h(\mathbf{x})^T \cdot \mathbf{d} = 0$, musí platiť aj $\nabla f(\mathbf{x})^T \cdot \mathbf{d} = 0$. Ináč povedané, ak je v \mathbf{x} extrém, tak každé \mathbf{d} kolmé na $\nabla h(\mathbf{x})$ je kolmé aj na $\nabla f(\mathbf{x})$. Keďže priestor vektorov kolmých na $\nabla h(\mathbf{x})$ je $(n-1)$ -rozmerný rovnako ako priestor vektorov kolmých na $\nabla f(\mathbf{x})$, tak sú tieto priestory pre lokálny extrém \mathbf{x} totožné, pozri Obrázok 14 pre prípad $n = 2$. Ale to znamená, že $\nabla f(\mathbf{x})$ je násobkom $\nabla h(\mathbf{x})$. Čiže existuje $\nu \in \mathbb{R}$, $\nu \neq 0$, také, že

$$\nabla f(\mathbf{x}) = -\nu \cdot \nabla h(\mathbf{x}),$$

čiže

$$\nabla f(\mathbf{x}) + \nu \cdot \nabla h(\mathbf{x}) = \mathbf{0}.$$

Ak zostrojíme funkciu $L(\mathbf{x}, \nu) = f(\mathbf{x}) + \nu \cdot h(\mathbf{x})$, tak naša podmienka tvrdí, že ak je v \mathbf{x} lokálny extrém, tak

$$\nabla_{\mathbf{x}} L(\mathbf{x}, \nu) = \mathbf{0}$$

pre nejaké $\nu \in \mathbb{R}$, $\nu \neq 0$.

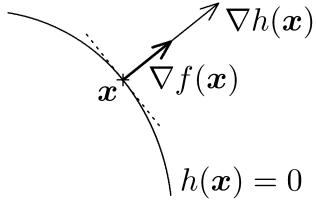
V analýze sme uvažovali iba prípady $\nabla f(\mathbf{x}) \neq \mathbf{0}$ a $\nabla h(\mathbf{x}) \neq \mathbf{0}$. Čo ak $\nabla f(\mathbf{x}) = \mathbf{0}$? Potom, bez ohľadu na to či v \mathbf{x} je alebo nie je extrém, platí

$$\nabla_{\mathbf{x}} L(\mathbf{x}, \nu) = \nabla f(\mathbf{x}) + \nu \cdot \nabla h(\mathbf{x}) = \mathbf{0}$$

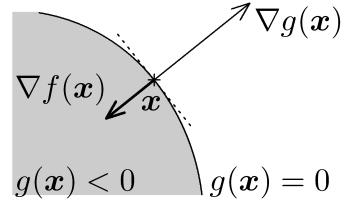
pre $\nu = 0$.

Ked' to zhrnieme (prípad $\nabla h(\mathbf{x}) = \mathbf{0}$ vylučujeme, dôvod vysvetlíme neskôr), tak ak je v \mathbf{x} lokálny extrém a \mathbf{x} je vnútorným bodom D , potom platí

$$\begin{aligned}\nabla_{\mathbf{x}} L(\mathbf{x}, \nu) &= \mathbf{0} \\ h(\mathbf{x}) &= 0.\end{aligned}$$



Obrázok 14



Obrázok 15

Jedno ohraničenie v tvare nerovnice

Teraz budeme uvažovať úlohu, ktorá má iba jedno ohraničenie v tvare nerovnice

$$\begin{array}{c}\min f(\mathbf{x}) \\ \hline g(\mathbf{x}) \leq 0 \\ \hline \mathbf{x} \in \mathbb{R}^n\end{array}$$

Nech $\mathbf{x} \in \mathcal{F}$. Budeme uvažovať dva prípady, $g(\mathbf{x}) = 0$ a $g(\mathbf{x}) < 0$.

1. Prípad $g(\mathbf{x}) = 0$.

Podobne ako v prípade rovnice, ak v \mathbf{y} nie je lokálny extrém, tak pre nejaký smer \mathbf{d} musí platiť

$$\begin{aligned}f(\mathbf{y} + \mathbf{d}) &< f(\mathbf{y}) \quad \text{a} \\ g(\mathbf{y} + \mathbf{d}) &\leq 0,\end{aligned}\tag{8}$$

čiže $\mathbf{y} + \mathbf{d}$ je prípustné riešenie a zároveň v smere \mathbf{d} hodnota funkcie f klesá. Podľa (6) je systém (8) ekvivalentný s

$$\begin{aligned}\nabla f(\mathbf{y})^T \cdot \mathbf{d} &< 0 \\ \nabla g(\mathbf{y})^T \cdot \mathbf{d} &\leq 0.\end{aligned}\tag{9}$$

Teraz predpokladajme, že sú v bode \mathbf{x} vektory $\nabla f(\mathbf{x})$ a $\nabla g(\mathbf{x})$ nenulové. Ako sme ukázali v prípade rovnice, ak $\nabla f(\mathbf{x})$ nie je násobkom $\nabla g(\mathbf{x})$, tak existuje \mathbf{d} také, že $\nabla f(\mathbf{x})^T \cdot \mathbf{d} < 0$ a $\nabla g(\mathbf{x})^T \cdot \mathbf{d} = 0$, čiže vtedy v \mathbf{x} nie je lokálny extrém.

Preto ak v \mathbf{x} lokálny extrém je, musí byť $\nabla f(\mathbf{x})$ násobkom $\nabla g(\mathbf{x})$. A nielen to. Ak sú $\nabla f(\mathbf{x})$ a $\nabla g(\mathbf{x})$ rovnako orientované, tak pre ľubovoľné \mathbf{d} , pre ktoré $\nabla g(\mathbf{x})^T \cdot \mathbf{d} < 0$, platí aj $\nabla f(\mathbf{x})^T \cdot \mathbf{d} < 0$, čo značí, že v \mathbf{x} extrém nie je (pozri vzťahy

(9)). Preto musia byť $\nabla f(\mathbf{x})$ a $\nabla g(\mathbf{x})$ orientované opačne, pozri Obrázok 15 pre prípad $n = 2$. Teda existuje $\lambda > 0$ také, že

$$\nabla f(\mathbf{x}) = -\lambda \cdot \nabla g(\mathbf{x}).$$

Vtedy pre funkciu $L(\mathbf{x}, \lambda) = f(\mathbf{x}) + \lambda \cdot g(\mathbf{x})$ platí

$$\begin{aligned}\nabla_{\mathbf{x}} L(\mathbf{x}, \lambda) &= 0 \\ \lambda > 0, \quad g(\mathbf{x}) &= 0.\end{aligned}$$

Opäť prípad $\nabla g(\mathbf{x}) = \mathbf{0}$ vylučujeme (dôvod vysvetlíme neskôr), avšak čo ak $\nabla f(\mathbf{x}) = \mathbf{0}$? Vtedy platí

$$\begin{aligned}\nabla_{\mathbf{x}} L(\mathbf{x}, \lambda) &= \mathbf{0} \quad \text{pre } \lambda = 0 \\ \text{a} \quad g(\mathbf{x}) &= 0.\end{aligned}$$

2. Prípad $g(\mathbf{x}) < 0$.

V tomto prípade pre každé dostatočne malé \mathbf{d} platí $g(\mathbf{x} + \mathbf{d}) < 0$. Preto ak v \mathbf{x} nie je extrém, potom z (6) plynie, že existuje smer \mathbf{d} , pre ktorý

$$\nabla f(\mathbf{x})^T \cdot \mathbf{d} < 0.$$

Lenže ak existuje \mathbf{d} také, že $\nabla f(\mathbf{x})^T \cdot \mathbf{d} \neq 0$, tak bud' $\nabla f(\mathbf{x})^T \cdot \mathbf{d} < 0$ alebo $\nabla f(\mathbf{x})^T \cdot (-\mathbf{d}) < 0$. (Všimnime si, že tu opäť využívame, že \mathbf{x} je vnútorným bodom D .) Teda ak existuje \mathbf{d} také, že $\nabla f(\mathbf{x})^T \cdot \mathbf{d} \neq 0$, vtedy v \mathbf{x} extrém nie je.

Takže ak v \mathbf{x} lokálny extrém je, musí pre každé \mathbf{d} platiť $\nabla f(\mathbf{x})^T \cdot \mathbf{d} = 0$, čo znamená, že musí platiť $\nabla f(\mathbf{x}) = \mathbf{0}$. Vtedy pre $L(\mathbf{x}, \lambda) = f(\mathbf{x}) + \lambda \cdot g(\mathbf{x})$ platí

$$\begin{aligned}\nabla_{\mathbf{x}} L(\mathbf{x}, \lambda) &= \mathbf{0} \quad \text{pre } \lambda = 0 \\ \text{a} \quad g(\mathbf{x}) &< 0.\end{aligned}$$

Keď obidva prípady, $g(\mathbf{x}) = 0$ aj $g(\mathbf{x}) < 0$ zhrnieme, tak nutnou podmienkou na to, aby bol v \mathbf{x} , ktorý je vnútorným bodom D , lokálny extrém je

$$\begin{aligned}\nabla_{\mathbf{x}} L(\mathbf{x}, \lambda) &= \mathbf{0} \\ \lambda &\geq 0 \\ \lambda \cdot g(\mathbf{x}) &= 0.\end{aligned}$$

Lineárna nezávislosť gradientov ohraničení

V našej analýze sme pripúšťali možnosť $\nabla f(\mathbf{x}) = \mathbf{0}$, no vylučovali sme prípady $\nabla h(\mathbf{x}) = \mathbf{0}$, respektíve $\nabla g(\mathbf{x}) = \mathbf{0}$ pre $g(\mathbf{x}) = 0$. V nasledujúcom príklade ukážeme, prečo sú tieto prípady zlé.

PRÍKLAD. Vyriešte úlohu

$$\begin{array}{c} \min x_1 + x_2 \\ \hline (x_1^2 + x_2^2 - 2)^2 = 0 \\ \hline x_1, x_2 \in \mathbb{R} \end{array}$$

RIEŠENIE. Je zrejmé, že ohraničenie je splnené vtedy, keď $x_1^2 + x_2^2 = 2$, čiže keď sú (x_1, x_2) body kružnice o polomere $\sqrt{2}$ so stredom v $(0, 0)$. Preto je minimum v bode $\mathbf{x}^* = (-1, -1)$, pozri Obrázok 16 (červenými prerusovanými čiarami sú vykreslené vrstevnice úcelovej funkcie).

Úlohu sme vyriešili, avšak ešte určíme $\nabla_{\mathbf{x}} L(\mathbf{x}, \nu)$. Nech

$$L(\mathbf{x}, \nu) = f(\mathbf{x}) + \nu \cdot h(\mathbf{x}) = x_1 + x_2 + \nu \cdot (x_1^2 + x_2^2 - 2)^2.$$

Potom

$$\begin{aligned} \frac{\partial}{\partial x_1} h(\mathbf{x}) &= 2(x_1^2 + x_2^2 - 2) \cdot 2x_1 \\ \frac{\partial}{\partial x_2} h(\mathbf{x}) &= 2(x_1^2 + x_2^2 - 2) \cdot 2x_2 \end{aligned}$$

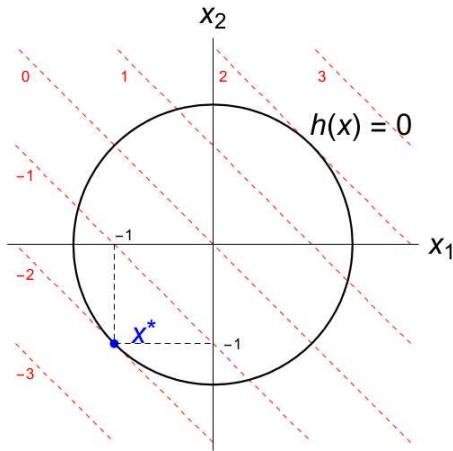
čiže

$$\nabla h(\mathbf{x}) = 4(x_1^2 + x_2^2 - 2) \cdot \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

pre každé \mathbf{x} , pre ktoré $h(\mathbf{x}) = 0$. Znamená to, že pre každé prípustné riešenie $\mathbf{x} \in \mathcal{F}$ máme $\nabla h(\mathbf{x}) = \mathbf{0}$. Preto

$$\nabla_{\mathbf{x}} L(\mathbf{x}, \nu) = \nabla_{\mathbf{x}} (x_1 + x_2 + \nu \cdot (x_1^2 + x_2^2 - 2)^2) = \begin{pmatrix} 1 \\ 1 \end{pmatrix} + \nu \cdot \begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

Teda $\nabla_{\mathbf{x}} L(\mathbf{x}, \nu)$ je nenulový vektor pre každé $\mathbf{x} \in \mathcal{F}$, aj pre optimálne riešenie úlohy.



Obrázok 16

Prípadu, keď $\nabla_{\mathbf{x}} L \neq \mathbf{0}$ pre optimálne riešenie, sa vyhneme pomocou nasledujúcej definície. V tejto definícii je LICQ skratka z linear independence constrained qualification a ide o podmienku na lineárnu nezávislosť gradientov ohraničení.

DEFINÍCIA (LICQ podmienka). Nech je \mathbf{x} prípustné riešenie úlohy matematického programovania (5). Označme tie i , pre ktoré $g_i(\mathbf{x}) = 0$ symbolom $I_{ac}(\mathbf{x})$. Potom v \mathbf{x} je **LICQ podmienka splnená**, ak je množina vektorov

$$\{\nabla g_i(\mathbf{x}); i \in I_{ac}(\mathbf{x})\} \cup \{\nabla h_j(\mathbf{x}); 1 \leq j \leq m\}$$

lineárne nezávislá.

Karushove-Kuhnove-Tuckerove podmienky

Teraz môžeme uviesť nutnú podmienku na to, aby mala úloha (5) vo vnútornom bode D lokálny extrém (pozri tiež odvodenie na strane 51).

VETA (Karushove-Kuhnove-Tuckerove podmienky). Nech je \mathbf{x} lokálne optimum úlohy (5), ktoré leží vnútri D a splňa LICQ podmienku. Potom existujú $\boldsymbol{\lambda} \in \mathbb{R}^\ell$ a $\boldsymbol{\nu} \in \mathbb{R}^m$ také, že pre Lagrangeovu funkciu

$$L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) = f(\mathbf{x}) + \sum_{i=1}^{\ell} \lambda_i \cdot g_i(\mathbf{x}) + \sum_{j=1}^m \nu_j \cdot h_j(\mathbf{x})$$

platí

$$\begin{aligned} \nabla_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) &= \mathbf{0} \\ g_i(\mathbf{x}) &\leq 0 & 1 \leq i \leq \ell \\ \lambda_i &\geq 0 & 1 \leq i \leq \ell \\ \lambda_i \cdot g_i(\mathbf{x}) &= 0 & 1 \leq i \leq \ell \\ h_j(\mathbf{x}) &= 0 & 1 \leq j \leq m. \end{aligned}$$

Karushove-Kuhnove-Tuckerove podmienky budeme v ďalšom označovať skrátene ako KKT podmienky.

PRÍKLAD. Pomocou KKT podmienok vyriešte úlohu

$$\begin{array}{ll} \min & -x_1 + x_2 \\ \text{---} & x_1(x_1^2 - 1) - x_2 \leq 0 \\ \text{---} & x_1, x_2 \in \mathbb{R} \end{array}$$

RIEŠENIE. Najprv zistíme, pre ktoré $\mathbf{x} \in \mathcal{F}$ je splnená LICQ podmienka. Kedže

$$\nabla g(\mathbf{x}) = \nabla(x_1(x_1^2 - 1) - x_2) = \begin{pmatrix} 3x_1^2 - 1 \\ -1 \end{pmatrix},$$

čo je pre každý \mathbf{x} nenulový vektor, tak LICQ podmienka je splnená vždy. Zjavne $D_f = \mathbb{R}^2$. Zostrojme Lagrangeovu funkciu

$$L(x_1, x_2, \lambda) = -x_1 + x_2 + \lambda \cdot (x_1(x_1^2 - 1) - x_2)$$

a zostavme KKT podmienky, čiže $\nabla_{\mathbf{x}} L(\mathbf{x}, \lambda) = \mathbf{0}$, $g(\mathbf{x}) \leq 0$, $\lambda \geq 0$ a $\lambda \cdot g(\mathbf{x}) = 0$. Dostávame

$$L'_{x_1}(x_1, x_2, \lambda) = -1 + \lambda \cdot (3x_1^2 - 1) = 0 \quad (\text{a})$$

$$L'_{x_2}(x_1, x_2, \lambda) = 1 + \lambda \cdot (-1) = 0 \quad (\text{b})$$

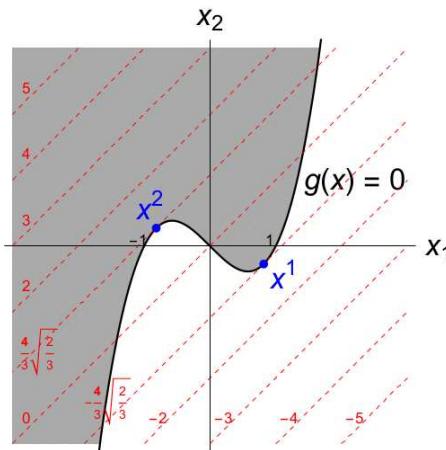
$$x_1(x_1^2 - 1) - x_2 \leq 0 \quad (\text{c})$$

$$\lambda \geq 0 \quad (\text{d})$$

$$\lambda \cdot (x_1(x_1^2 - 1) - x_2) = 0 \quad (\text{e})$$

pričom prvé dve rovnice tvrdia, že parciálne derivácie Lagrangeovej funkcie sa rovnajú 0.

Z (b) dostávame $\lambda = 1$. Dosadením $\lambda = 1$ do (a) získame $3x_1^2 = 2$, čiže $x_1 = \sqrt{\frac{2}{3}}$, respektíve $x_1 = -\sqrt{\frac{2}{3}}$. Teraz z (e) máme $x_2 = x_1(\frac{2}{3} - 1)$, čiže $x_2 = -\frac{1}{3}x_1$ a (c) a (d) sú splnené. Získali sme dva body, v ktorých môžu byť lokálne extrémy. A súce $(x_1, x_2, \lambda) = \left(\sqrt{\frac{2}{3}}, -\frac{1}{3}\sqrt{\frac{2}{3}}, 1\right)$ a $(x_1, x_2, \lambda) = \left(-\sqrt{\frac{2}{3}}, \frac{1}{3}\sqrt{\frac{2}{3}}, 1\right)$. V bode $\mathbf{x}^1 = \left(\sqrt{\frac{2}{3}}, -\frac{1}{3}\sqrt{\frac{2}{3}}\right)$ je lokálne minimum, zatiaľ čo v bode $\mathbf{x}^2 = \left(-\sqrt{\frac{2}{3}}, \frac{1}{3}\sqrt{\frac{2}{3}}\right)$ lokálne minimum nie je. (V \mathbf{x}^2 je lokálne maximum pri ohraničení $g(\mathbf{x}) = 0$, ale nie pri $g(\mathbf{x}) \leq 0$.) Globálne minimum úloha nemá, pozri Obrázok 17.



Obrázok 17

Platí nasledujúce tvrdenie.

VETA. Ak sú funkcie f, g_1, \dots, g_ℓ konvexné a h_1, \dots, h_m sú lineárne, tak pre vnútorné body D splňajúce LICQ podmienku sú KKT podmienky nielen nutné, ale aj postačujúce na existenciu globálneho extrému.

Poznamenanajme, že úloha spomenutá v predchádzajúcej vete sa nazýva úloha konvexného programovania, pozri nasledujúcu kapitolu. Všimnime si, že v predchádzajúcim príklade KKT podmienky neboli postačujúce na existenciu globálneho extrému. Prečo?

Porovnanie s prv uvádzanými vetami

V tejto časti porovnáme KKT podmienky s tvrdeniami, ktoré sme uviedli v druhom ročníku, pozri [10]. Vtedy sme LICQ podmienku ani podmienku požadujúcu, aby bol \mathbf{x} vnútorným bodom D_f , kvôli jednoduchosti neuvádzali. Osobitne sme uvažovali prípad, keď boli všetky ohraničenia rovnice a osobitne prípad, keď boli všetky ohraničenia nerovnice.

Teda najprv budeme uvažovať úlohu

$$\begin{array}{ll} \text{opt } f(\mathbf{x}) \\ \hline h_j(\mathbf{x}) = 0 & 1 \leq j \leq m \\ \mathbf{x} \in \mathbb{R}^n \end{array} \quad (10)$$

kde $\text{opt} \in \{\min, \max\}$. Mali sme tvrdenie

VETA. Ak v kritickom bode funkcie $L(\mathbf{x}, \boldsymbol{\nu}) = f(\mathbf{x}) + \sum_{j=1}^m \nu_j \cdot h_j(\mathbf{x})$ existujú všetky druhé parciálne derivácie podľa x -ov a pre hlavné minory Hessovej matice L platí

- (i) $D_1 > 0, D_2 > 0, \dots, D_i > 0, \dots, D_n > 0$, tak je tento kritický bod lokálnym minimom (10);
- (ii) $D_1 < 0, D_2 > 0, \dots, (-1)^i D_i > 0, \dots, (-1)^n D_n > 0$, tak je tento kritický bod lokálnym maximom (10).

Táto veta tvrdí viac ako KKT podmienky, s ktorými je v súlade. Totiž bod $(\mathbf{x}, \boldsymbol{\nu})$ je kritický, ak sa v ňom všetky parciálne derivácie rovnajú 0, čiže ak

$$\begin{aligned} \nabla_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\nu}) &= 0 \\ \frac{\partial}{\partial \nu_j} L(\mathbf{x}, \boldsymbol{\nu}) &= 0 \quad \text{pre } 1 \leq j \leq m. \end{aligned}$$

Kedže $\frac{\partial}{\partial \nu_j} L(\mathbf{x}, \boldsymbol{\nu}) = h_j(\mathbf{x})$, druhá sada rovníc tvrdí $h_j(\mathbf{x}) = 0$ pre $1 \leq j \leq m$.

Teraz budeme uvažovať úlohu

$$\begin{array}{ll} \text{opt } f(\mathbf{x}) \\ \hline g_i(\mathbf{x}) \geq 0 & 1 \leq i \leq \ell \\ \mathbf{x} \in \mathbb{R}^n \end{array} \quad (11)$$

kde $\text{opt} \in \{\min, \max\}$. Všimnime si, že nerovnosti sú v tejto úlohe opačne v porovnaní s (5). Mali sme tvrdenie

VETA. Ak je v \mathbf{x} extrém (čiže riešenie) úlohy (11), tak tento bod vyhovuje ohraničeniam a existujú nezáporné $\lambda_1, \dots, \lambda_\ell$ také, že

$$\begin{aligned} \frac{\partial}{\partial x_k} f(\mathbf{x}) + \sum_{i=1}^{\ell} \lambda_i \cdot \frac{\partial}{\partial x_k} g_i(\mathbf{x}) &= 0 \quad \text{pre } 1 \leq k \leq n \\ \lambda_i \cdot g_i(\mathbf{x}) &= 0 \quad \text{pre } 1 \leq i \leq \ell \end{aligned}$$

pričom ak opt je min, tak pred sumou budeme uvažovať $-$, zatiaľ čo ak opt je max, budeme pred sumou uvažovať $+$.

Aj tátó veta je v súlade s KKT podmienkami. Totiž ak opt je min, tak má platiť

$$\frac{\partial}{\partial x_k} f(\mathbf{x}) - \sum_{i=1}^{\ell} \lambda_i \cdot \frac{\partial}{\partial x_k} g_i(\mathbf{x}) = \frac{\partial}{\partial x_k} f(\mathbf{x}) + \sum_{i=1}^{\ell} \lambda_i \cdot \frac{\partial}{\partial x_k} (-g_i(\mathbf{x})) = 0$$

pre $1 \leq k \leq n$, čo je presne podmienka $\nabla_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\lambda}) = \mathbf{0}$, keďže žiadame, aby platilo $-g_i(\mathbf{x}) \leq 0$. Ostatné podmienky, čiže $g_i(\mathbf{x}) \leq 0$, $\lambda_i \geq 0$ a $\lambda_i \cdot g_i(\mathbf{x}) = 0$, sú priamo v znení vety.

Ak opt je max, tak má platiť

$$\frac{\partial}{\partial x_k} f(\mathbf{x}) + \sum_{i=1}^{\ell} \lambda_i \cdot \frac{\partial}{\partial x_k} g_i(\mathbf{x}) = 0$$

pre $1 \leq k \leq n$. Lenže $\max f(\mathbf{x}) = -\min(-f(\mathbf{x}))$ a vtedy nám KKT podmienky (pre $-g_i(\mathbf{x}) \leq 0$) dajú $\nabla_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\lambda}) = \mathbf{0}$. Keď tento výraz rozpišeme, dostaneme

$$\frac{\partial}{\partial x_k} (-f(\mathbf{x})) + \sum_{i=1}^{\ell} \lambda_i \cdot \frac{\partial}{\partial x_k} (-g_i(\mathbf{x})) = 0$$

pre $1 \leq k \leq n$, čo je po prenásobení -1 rovnica uvedená vyššie. Ostatné podmienky sú opäť priamo v znení vety.

Cvičenia

CVIČENIE 4.1 (REVENUE MANAGEMENT [1]). Spoločnosť Leisure Air má dve lietadlá s kapacitou 132 sedadiel, pričom jedno má bázu v Pittsburghu a druhé v Newarku. Každé ráno letí lietadlo z Pittsburghu do Orlanda s medzipristátím v Charlotte a druhé lietadlo letí z Newarku do Myrtle Beach, tiež s medzipristátím v Charlotte. Podvečer sa obidve lietadlá vracajú, nás však budú zaujímať iba ranné lety. Vďaka nim spoločnosť predáva lety z Pittsburghu a Newarku do Charlotte, Orlanda a Myrtle Beach a z Charlotte do Orlanda a Myrtle Beach.

Spoločnosť predáva letenky dvoch kategórií. Discount-fare Q class a full-fare Y class. Letenky kategórie Q musia byť rezervované v predstihu, minimálne 14 dní pred odletom, zatiaľ čo letenky kategórie Y môžu byť rezervované kedykoľvek, bez postihu za zmenu termínu letu. Znamená to, že spoločnosť predáva na ranné lety 16 typov leteniek. V nasledujúcej tabuľke máme predpokladaný dopyt a cenu leteniek na konkrétny deň. Spoločnosť sa podľa tohto dopytu musí rozhodnúť, koľko leteniek ktorej kategórie bude na jednotlivé lety predávať, pričom jej cieľom je maximalizácia výnosu.

Let	Odlet	Prílet	Kategória	Cena	Dopyt
1	Pittsburg	Charlotte	Q	178	33
2	Pittsburg	Myrtle Beach	Q	268	44
3	Pittsburg	Orlando	Q	228	45
4	Pittsburg	Charlotte	Y	380	16
5	Pittsburg	Myrtle Beach	Y	456	6
6	Pittsburg	Orlando	Y	560	11
7	Newark	Charlotte	Q	199	26
8	Newark	Myrtle Beach	Q	249	56
9	Newark	Orlando	Q	349	39
10	Newark	Charlotte	Y	385	15
11	Newark	Myrtle Beach	Y	444	7
12	Newark	Orlando	Y	580	9
13	Charlotte	Myrtle Beach	Q	179	64
14	Charlotte	Myrtle Beach	Y	380	8
15	Charlotte	Orlando	Q	224	46
16	Charlotte	Orlando	Y	582	10

1. Sformulujte problém ako úlohu lineárneho programovania.
2. Nájdite kanonický tvar úlohy.
3. Vyriešte úlohu v softvéri Mathematica (použite Vás program na Mehrotrov algoritmus). Tu treba dosadiť príslušné hodnoty do \mathbb{A} , \mathbf{c} a \mathbf{b} , zadať štartovací bod $(\mathbf{x}^0, \mathbf{y}^0, \mathbf{s}^0)$ (rozmyslite si, ktoré KKT podmienky môžu byť narušené), spustiť algoritmus a vizualizovať kroky \mathbf{x}^k (napríklad pomocou `ListPlot`).

CVIČENIE 4.2 (ANALÝZA SENZITIVITY). Skúšajte meniť jednotlivé zložky vektoru pravej strany \mathbf{b} z Cvičenia 4.1. Zistite, zväčšenie ktorej z nich prinesie najväčšie zvýšenie výnosov.

V Cvičeniach 4.3 až 4.7 zistite:

1. Je f konvexná funkcia?
2. Sú ohraničenia dané konvexnými funkciami? (Niektoré možno treba najprv upraviť do požadovaného tvaru.)
3. Nájdite všetky body $(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu})$ splňajúce KKT podmienky. Môžete si pomôcť softvérom Mathematica.
4. Ktorý bod je optimálne riešenie úlohy?

CVIČENIE 4.3. Vyriešte úlohu

$$\begin{array}{ll} \min f(\mathbf{x}) = 2x_1 + x_2 \\ \hline x_1^2 + x_2^2 \leq 4 \end{array}$$

CVIČENIE 4.4. Vyriešte úlohu

$$\begin{array}{ll} \min f(\mathbf{x}) = \frac{50}{x_1} + \frac{20}{x_2} + x_1 x_2 \\ \hline x_1 \geq 1 \quad x_2 \geq 1 \end{array}$$

CVIČENIE 4.5. Vyriešte úlohu

$$\begin{array}{l} \min f(\mathbf{x}) = -3x_1^2 + x_2^2 + 2x_3^2 + 2(x_1 + x_2 + x_3) \\ \hline x_1^2 + x_2^2 + x_3^2 = 1 \end{array}$$

CVIČENIE 4.6. Vyriešte úlohu

$$\begin{array}{l} \min f(\mathbf{x}) = \sqrt{x_1} + \sqrt{x_2} \\ \hline x_1^2 - x_2 \geq 1 \end{array}$$

CVIČENIE 4.7. Zmeňte účelovú funkciu v Cvičení 4.6 na $f(\mathbf{x}) = \sqrt{x_1^2} + \sqrt{x_2^2}$ a vysvetlite, prečo KKT podmienky v obidvoch týchto cvičeniach zlyhali, hoci riešenie je triviálne.

5 Dualita v matematickom programovaní

Budeme uvažovať úlohu (5), ktorú sme zaviedli v predchádzajúcej kapitole

$$\text{P: } \begin{array}{c} \min f(\mathbf{x}) \\ \hline g_i(\mathbf{x}) \leq 0 & 1 \leq i \leq \ell \\ h_j(\mathbf{x}) = 0 & 1 \leq j \leq m \\ \hline \mathbf{x} \in D \end{array}$$

kde $\mathbf{x} \in \mathbb{R}^n$ a $D = D_f \cap D_{g_1} \cap \dots \cap D_{g_\ell} \cap D_{h_1} \cap \dots \cap D_{h_m}$. Úlohu (5) nazveme primárna úloha a označíme ju P.

Zostrojme Lagrangeovu funkciu

$$L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) = f(\mathbf{x}) + \sum_{i=1}^{\ell} \lambda_i \cdot g_i(\mathbf{x}) + \sum_{j=1}^m \nu_j \cdot h_j(\mathbf{x}),$$

kde $\lambda_i \geq 0$ pre $1 \leq i \leq \ell$. Definujme

$$\psi(\boldsymbol{\lambda}, \boldsymbol{\nu}) = \inf_{\mathbf{x} \in D} \{L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu})\} = \inf_{\mathbf{x} \in D} \left\{ f(\mathbf{x}) + \sum_{i=1}^{\ell} \lambda_i \cdot g_i(\mathbf{x}) + \sum_{j=1}^m \nu_j \cdot h_j(\mathbf{x}) \right\}. \quad (12)$$

Funkciu $\psi(\boldsymbol{\lambda}, \boldsymbol{\nu})$ nazývame duálna účelová funkcia.

Konkávnosť $\psi(\boldsymbol{\lambda}, \boldsymbol{\nu})$

Ukážeme, že bez ohľadu na funkcie f , g_i a h_j je ψ konkávna funkcia.

Pripomeňme, že všeobecná funkcia $q(\mathbf{y})$ je **konkávna**, ak je jej definičný obor konvexná množina, čiže ak pre každé $\mathbf{a}, \mathbf{b} \in D_q$ a $\theta \in \langle 0, 1 \rangle$ platí $\theta \cdot \mathbf{a} + (1-\theta) \cdot \mathbf{b} \in D_q$ a naviac

$$q(\theta \cdot \mathbf{a} + (1-\theta) \cdot \mathbf{b}) \geq \theta \cdot q(\mathbf{a}) + (1-\theta) \cdot q(\mathbf{b}).$$

Naopak, $q(\mathbf{y})$ je **konvexná** ak je D_q konvexná množina a pre každé $\mathbf{a}, \mathbf{b} \in D_q$ a $\theta \in \langle 0, 1 \rangle$ platí

$$q(\theta \cdot \mathbf{a} + (1-\theta) \cdot \mathbf{b}) \leq \theta \cdot q(\mathbf{a}) + (1-\theta) \cdot q(\mathbf{b}).$$

Lineárna funkcia je konvexná aj konkávna zároveň.

Pre zjednodušenie zápisu označme $\mathbb{R}_+ = \langle 0, \infty \rangle$.

VETA. Funkcia $\psi(\boldsymbol{\lambda}, \boldsymbol{\nu})$ je konkávna.

DÔKAZ. V skutočnosti ukážeme, že je $-\psi$ konvexná.

Najprv si všimnime, že ak sú $q_1(\mathbf{y})$ a $q_2(\mathbf{y})$ konvexné funkcie definované na konvexných množinách D_{q_1} a D_{q_2} , tak je konvexná aj $q(\mathbf{y}) = \max\{q_1(\mathbf{y}), q_2(\mathbf{y})\}$. Totiž prienik konvexných množín je konvexná množina, čiže $D_q = D_{q_1} \cap D_{q_2}$ je konvexná, a ďalej pre každé $\mathbf{a}, \mathbf{b} \in D_q$ a $\theta \in \langle 0, 1 \rangle$ platí

$$\begin{aligned} q(\theta \cdot \mathbf{a} + (1-\theta) \cdot \mathbf{b}) &= \max \{q_1(\theta \cdot \mathbf{a} + (1-\theta) \cdot \mathbf{b}), q_2(\theta \cdot \mathbf{a} + (1-\theta) \cdot \mathbf{b})\} \\ &\leq \max \{\theta \cdot q_1(\mathbf{a}) + (1-\theta) \cdot q_1(\mathbf{b}), \theta \cdot q_2(\mathbf{a}) + (1-\theta) \cdot q_2(\mathbf{b})\} \\ &\leq \max \{\theta \cdot q_1(\mathbf{a}), \theta \cdot q_2(\mathbf{a})\} + \max \{(1-\theta) \cdot q_1(\mathbf{b}), (1-\theta) \cdot q_2(\mathbf{b})\} \\ &= \theta \cdot q(\mathbf{a}) + (1-\theta) \cdot q(\mathbf{b}) \end{aligned}$$

Dané tvrdenie možno rozšíriť na viac funkcií q_k , dokonca aj na nekonečnú množinu týchto funkcií. Teda ak sú q_k pre každé k z indexovej množiny I konvexné funkcie, tak je konvexná aj

$$q(\mathbf{y}) = \sup \{q_k(\mathbf{y}); k \in I\}.$$

Teraz uvažujme

$$q_{\mathbf{x}}(\boldsymbol{\lambda}, \boldsymbol{\nu}) = -f(\mathbf{x}) - \sum_{i=1}^{\ell} \lambda_i \cdot g_i(\mathbf{x}) - \sum_{j=1}^m \nu_j \cdot h_j(\mathbf{x}),$$

kde $\mathbf{x} \in D$. Pre každé \mathbf{x} z indexovej množiny D je funkcia $q_{\mathbf{x}}(\boldsymbol{\lambda}, \boldsymbol{\nu})$ lineárna, čiže konvexná (aj konkávna), pričom konvexná je aj každá množina $D_{q_{\mathbf{x}}} = \mathbb{R}_+^\ell \times \mathbb{R}^m$. Podľa predchádzajúceho je

$$q(\boldsymbol{\lambda}, \boldsymbol{\nu}) = \sup_{\mathbf{x} \in D} \left\{ -f(\mathbf{x}) - \sum_{i=1}^{\ell} \lambda_i \cdot g_i(\mathbf{x}) - \sum_{j=1}^m \nu_j \cdot h_j(\mathbf{x}) \right\}$$

konvexná funkcia. To však znamená, že $\psi(\boldsymbol{\lambda}, \boldsymbol{\nu}) = -q(\boldsymbol{\lambda}, \boldsymbol{\nu})$ je konkávna. \square

Duálna úloha

Pre funkciu ψ definovanú vzťahom (12) platí nasledujúce tvrdenie.

VETA. Nech je \mathbf{x} prípustné riešenie primárnej úlohy P a nech $\boldsymbol{\lambda} \geq \mathbf{0}$. Potom pre každé $\boldsymbol{\nu}$ platí

$$f(\mathbf{x}) \geq \psi(\boldsymbol{\lambda}, \boldsymbol{\nu}).$$

DÔKAZ. Keďže platí $g_i(\mathbf{x}) \leq 0$ a zároveň $\lambda_i \geq 0$ pre $1 \leq i \leq \ell$, tak platí aj $\sum_{i=1}^{\ell} \lambda_i \cdot g_i(\mathbf{x}) \leq 0$. Naviac, $h_j(\mathbf{x}) = 0$ pre $1 \leq j \leq m$, takže $\sum_{j=1}^m \nu_j \cdot h_j(\mathbf{x}) = 0$. Preto

$$\sum_{i=1}^{\ell} \lambda_i \cdot g_i(\mathbf{x}) + \sum_{j=1}^m \nu_j \cdot h_j(\mathbf{x}) \leq 0,$$

čiže

$$\begin{aligned}
f(\mathbf{x}) &\geq f(\mathbf{x}) + \sum_{i=1}^{\ell} \lambda_i \cdot g_i(\mathbf{x}) + \sum_{j=1}^m \nu_j \cdot h_j(\mathbf{x}) \\
&\geq \inf_{\tilde{\mathbf{x}} \in D} \left\{ f(\tilde{\mathbf{x}}) + \sum_{i=1}^{\ell} \lambda_i \cdot g_i(\tilde{\mathbf{x}}) + \sum_{j=1}^m \nu_j \cdot h_j(\tilde{\mathbf{x}}) \right\} \\
&= \psi(\boldsymbol{\lambda}, \boldsymbol{\nu}). \quad \square
\end{aligned}$$

Podľa predchádzajúcej vety pre každé prípustné riešenie $\mathbf{x} \in \mathcal{F}$ a pre každé $\boldsymbol{\lambda}$ a $\boldsymbol{\nu}$, kde $\boldsymbol{\lambda} \geq \mathbf{0}$, platí $f(\mathbf{x}) \geq \psi(\boldsymbol{\lambda}, \boldsymbol{\nu})$. Tento vzťah platí aj pre optimálne riešenie \mathbf{x}^* primárnej úlohy P. Znamená to, že pre každé $\boldsymbol{\lambda} \geq \mathbf{0}$ a $\boldsymbol{\nu}$ je $\psi(\boldsymbol{\lambda}, \boldsymbol{\nu})$ dolné ohraničenie pre $f(\mathbf{x}^*)$. Samozrejme, najlepšie dolné ohraničenie získame, keď bude $\psi(\boldsymbol{\lambda}, \boldsymbol{\nu})$ maximálne možné. Úlohu

$$\begin{array}{c}
\max \psi(\boldsymbol{\lambda}, \boldsymbol{\nu}) \\
\text{D: } \frac{\boldsymbol{\lambda} \geq \mathbf{0}}{\boldsymbol{\lambda} \in \mathbb{R}^{\ell}, \boldsymbol{\nu} \in \mathbb{R}^m}
\end{array}$$

kde $\psi(\boldsymbol{\lambda}, \boldsymbol{\nu})$ je definovaná vzťahom (12), nazývame **duálna úloha** k primárnej úlohe P.

DEFINÍCIA. Nech je \mathbf{x}^* optimálne riešenie P a $(\boldsymbol{\lambda}^*, \boldsymbol{\nu}^*)$ nech je optimálne riešenie D. Vzťah $f(\mathbf{x}^*) \geq \psi(\boldsymbol{\lambda}^*, \boldsymbol{\nu}^*)$ nazývame **slabá dualita**. Ak platí $f(\mathbf{x}^*) = \psi(\boldsymbol{\lambda}^*, \boldsymbol{\nu}^*)$, čiže ak majú účelové funkcie P a D rovnaké optimálne hodnoty, vtedy hovoríme, že platí **silná dualita**.

Ako sme ukázali, slabá dualita platí vždy. Silná dualita vždy neplatí, niekedy však splnená je.

Úlohu

$$\begin{array}{c}
\min f(\mathbf{x}) \\
\frac{g_i(\mathbf{x}) \leq 0}{\mathbf{A} \cdot \mathbf{x} = \mathbf{b}} \quad 1 \leq i \leq \ell \\
\mathbf{x} \in D
\end{array}$$

kde f, g_1, \dots, g_ℓ sú konvexné funkcie a \mathbf{A} je matica typu $(m \times n)$, nazývame **úloha konvexného programovania**. Pripomeňme, že $-\psi(\boldsymbol{\lambda}, \boldsymbol{\nu})$ je konvexná vždy. Ak v úlohe konvexného programovania existuje prípustné riešenie $\mathbf{x} \in \mathcal{F}$ také, že $g_i(\mathbf{x}) < 0$ pre všetky ohraničenia tvaru nerovnice, vtedy hovoríme, že úloha splňa **Slaterovu podmienku**. Ak úloha konvexného programovania splňa Slaterovu podmienku, tak silná dualita platí. Dokonca stačí, aby boli nerovnice $g_i(\mathbf{x}) < 0$ splnené pre nelineárne funkcie g_i (keďže niektoré z ohraničení môžu byť lineárne), a už vtedy silná dualita platí.

V Kapitolách 7 a 8 budeme riešiť úlohy konvexného programovania, ktoré splňajú Slaterovu podmienku. Znamená to, že budeme riešiť úlohy, pre ktoré silná dualita platí.

Sedlový bod

Silná dualita platí aj vtedy, keď má Lagrangeova funkcia $L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu})$ sedlový bod.

DEFINÍCIA. Majme funkciu $F(\mathbf{x}, \mathbf{y}) : \mathbb{R}^n \times \mathbb{R}^p \rightarrow \mathbb{R}$, kde $\mathbf{x} \in \mathbb{R}^n$ a $\mathbf{y} \in \mathbb{R}^p$. Ak platí

$$\min_{\mathbf{x} \in \mathbb{R}^n} \{F(\mathbf{x}, \mathbf{b})\} = F(\mathbf{a}, \mathbf{b}) = \max_{\mathbf{y} \in \mathbb{R}^p} \{F(\mathbf{a}, \mathbf{y})\},$$

tak (\mathbf{a}, \mathbf{b}) je **sedlový bod** funkcie F .

VETA. Ak je (\mathbf{a}, \mathbf{b}) je **sedlový bod** funkcie F , tak

$$\max_{\mathbf{y} \in \mathbb{R}^p} \left\{ \min_{\mathbf{x} \in \mathbb{R}^n} \{F(\mathbf{x}, \mathbf{y})\} \right\} = F(\mathbf{a}, \mathbf{b}) = \min_{\mathbf{x} \in \mathbb{R}^n} \left\{ \max_{\mathbf{y} \in \mathbb{R}^p} \{F(\mathbf{x}, \mathbf{y})\} \right\}.$$

DÔKAZ. Zjavne pre každé $(\mathbf{a}', \mathbf{b}')$ platí

$$\min_{\mathbf{x} \in \mathbb{R}^n} \{F(\mathbf{x}, \mathbf{b}')\} \leq F(\mathbf{a}', \mathbf{b}') \leq \max_{\mathbf{y} \in \mathbb{R}^p} \{F(\mathbf{a}', \mathbf{y})\}.$$

Teraz si zvolíme za \mathbf{b}' takú hodnotu \mathbf{y} , pre ktorú je $\min_{\mathbf{x}} F(\mathbf{x}, \mathbf{y})$ maximálne a za \mathbf{a}' si zvolíme takú hodnotu \mathbf{x} , pre ktorú je $\max_{\mathbf{y}} F(\mathbf{x}, \mathbf{y})$ minimálne. (Predpokladáme, že také hodnoty \mathbf{a}' a \mathbf{b}' existujú.) Potom

$$\begin{aligned} \max_{\mathbf{y} \in \mathbb{R}^p} \left\{ \min_{\mathbf{x} \in \mathbb{R}^n} \{F(\mathbf{x}, \mathbf{y})\} \right\} &= \min_{\mathbf{x} \in \mathbb{R}^n} \{F(\mathbf{x}, \mathbf{b}')\} \leq F(\mathbf{a}', \mathbf{b}') \leq \\ &\leq \max_{\mathbf{y} \in \mathbb{R}^p} \{F(\mathbf{a}', \mathbf{y})\} = \min_{\mathbf{x} \in \mathbb{R}^n} \left\{ \max_{\mathbf{y} \in \mathbb{R}^p} \{F(\mathbf{x}, \mathbf{y})\} \right\}. \end{aligned}$$

Naopak, keďže (\mathbf{a}, \mathbf{b}) je sedlový bod, tak platí

$$\begin{aligned} \max_{\mathbf{y} \in \mathbb{R}^p} \left\{ \min_{\mathbf{x} \in \mathbb{R}^n} \{F(\mathbf{x}, \mathbf{y})\} \right\} &\geq \min_{\mathbf{x} \in \mathbb{R}^n} \{F(\mathbf{x}, \mathbf{b})\} = F(\mathbf{a}, \mathbf{b}) = \\ &= \max_{\mathbf{y} \in \mathbb{R}^p} \{F(\mathbf{a}, \mathbf{y})\} \geq \min_{\mathbf{x} \in \mathbb{R}^n} \left\{ \max_{\mathbf{y} \in \mathbb{R}^p} \{F(\mathbf{x}, \mathbf{y})\} \right\}, \end{aligned}$$

čo spolu s predchádzajúcou nerovnosťou dáva tvrdenie vety. \square

Táto veta platí (s istou dávkou opatrnosti) aj keď max nahradíme sup a keď min nahradíme inf. Predpokladajme, že má Lagrangeova funkcia sedlový bod $(\mathbf{u}, (\mathbf{v}, \mathbf{w}))$, čiže

$$\inf_{\mathbf{x} \in D} \{L(\mathbf{x}, \mathbf{v}, \mathbf{w})\} = L(\mathbf{u}, \mathbf{v}, \mathbf{w}) = \sup_{\boldsymbol{\lambda} \in \mathbb{R}_+^\ell, \boldsymbol{\nu} \in \mathbb{R}^m} \{L(\mathbf{u}, \boldsymbol{\lambda}, \boldsymbol{\nu})\}.$$

Potom podľa vyššie uvedenej vety platí

$$\sup_{\boldsymbol{\lambda}, \boldsymbol{\nu}} \left\{ \inf_{\mathbf{x}} \{L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu})\} \right\} = L(\mathbf{u}, \mathbf{v}, \mathbf{w}) = \inf_{\mathbf{x}} \left\{ \sup_{\boldsymbol{\lambda}, \boldsymbol{\nu}} \{L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu})\} \right\},$$

kde $\mathbf{x} \in D$, $\boldsymbol{\lambda} \in \mathbb{R}_+^\ell$ a $\boldsymbol{\nu} \in \mathbb{R}^m$. Čo tvrdia jednotlivé strany tejto rovnice?

$$\inf_{\mathbf{x} \in D} \left\{ f(\mathbf{x}) + \sum_{i=1}^{\ell} \lambda_i \cdot g_i(\mathbf{x}) + \sum_{j=1}^m \nu_j \cdot h_j(\mathbf{x}) \right\} = \psi(\boldsymbol{\lambda}, \boldsymbol{\nu}).$$

Preto

$$\sup_{\boldsymbol{\lambda} \in \mathbb{R}_+^\ell, \boldsymbol{\nu} \in \mathbb{R}^m} \left\{ \inf_{\mathbf{x} \in D} \{L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu})\} \right\} = \sup_{\boldsymbol{\lambda} \in \mathbb{R}_+^\ell, \boldsymbol{\nu} \in \mathbb{R}^m} \{\psi(\boldsymbol{\lambda}, \boldsymbol{\nu})\} = \psi(\boldsymbol{\lambda}^*, \boldsymbol{\nu}^*)$$

ak má D optimálne riešenie $(\boldsymbol{\lambda}^*, \boldsymbol{\nu}^*)$. Naopak

$$\sup_{\boldsymbol{\lambda} \in \mathbb{R}_+^\ell, \boldsymbol{\nu} \in \mathbb{R}^m} \left\{ f(\mathbf{x}) + \sum_{i=1}^{\ell} \lambda_i \cdot g_i(\mathbf{x}) + \sum_{j=1}^m \nu_j \cdot h_j(\mathbf{x}) \right\} = \begin{cases} f(\mathbf{x}) & \text{ak } \mathbf{x} \in \mathcal{F}, \\ \infty & \text{ak } \mathbf{x} \notin \mathcal{F}. \end{cases}$$

Totiž ak $g_i(\mathbf{x}) > 0$ pre nejaké i , $1 \leq i \leq \ell$, tak zvolíme $\lambda_i \rightarrow \infty$ a dostaneme $\sup = \infty$. Naopak, ak $h_j(\mathbf{x}) \neq 0$ pre nejaké j , $1 \leq j \leq m$, tak pre $h_j(\mathbf{x}) > 0$ zvolíme $\nu_j \rightarrow \infty$ a pre $h_j(\mathbf{x}) < 0$ zvolíme $\nu_j \rightarrow -\infty$. V obidvoch prípadoch dostaneme opäť $\sup = \infty$. Preto

$$\inf_{\mathbf{x} \in D} \left\{ \sup_{\boldsymbol{\lambda} \in \mathbb{R}_+^\ell, \boldsymbol{\nu} \in \mathbb{R}^m} \{L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu})\} \right\} = \inf_{\mathbf{x} \in \mathcal{F}} \{f(\mathbf{x})\} = f(\mathbf{x}^*)$$

ak má P optimálne riešenie \mathbf{x}^* . Odvodili sme tvrdenie

VETA. Ak má Lagrangeova funkcia $L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu})$ sedlový bod $(\mathbf{u}, (\mathbf{v}, \mathbf{w}))$, P má optimálne riešenie \mathbf{x}^* a D má optimálne riešenie $(\boldsymbol{\lambda}^*, \boldsymbol{\nu}^*)$, tak potom $\mathbf{x}^* = \mathbf{u}$ a $(\boldsymbol{\lambda}^*, \boldsymbol{\nu}^*) = (\mathbf{v}, \mathbf{w})$. Naviac, platí silná dualita, čiže $f(\mathbf{x}^*) = \psi(\boldsymbol{\lambda}^*, \boldsymbol{\nu}^*)$.

PRÍKLAD. K úlohe

$$\text{P:} \quad \frac{\min x^2}{\begin{array}{c} 3-x \leq 0 \\ x \in \mathbb{R} \end{array}}$$

zostrojte duálnu úlohu D, nájdite optimum P aj D a zostrojte diagram Lagrangeovej funkcie.

RIEŠENIE. Duálna úloha je

$$\frac{\max \psi(\lambda)}{\begin{array}{c} \lambda \geq 0 \\ \lambda \in \mathbb{R} \end{array}}$$

avšak $\psi(\lambda)$ ešte musíme určiť. Zostrojíme Lagrangeovu funkciu

$$L(x, \lambda) = x^2 + \lambda(3 - x).$$

Teraz $\psi(\lambda) = \inf_{x \in \mathbb{R}} \{x^2 + \lambda(3 - x)\}$, pričom infimum určíme pomocou derivácie. Kedže grafom $x^2 + \lambda(3 - x)$ je parabola „chvostami nahor“, kritický bod bude minimum. Rovnica

$$(x^2 + \lambda(3 - x))' = 2x - \lambda = 0$$

má riešenie $x = \frac{\lambda}{2}$. Preto

$$\psi(\lambda) = L\left(\frac{\lambda}{2}, \lambda\right) = \left(\frac{\lambda}{2}\right)^2 + \lambda\left(3 - \frac{\lambda}{2}\right) = -\frac{\lambda^2}{4} + 3\lambda.$$

Dostávame

$$\begin{array}{c} \max -\frac{\lambda^2}{4} + 3\lambda \\ \hline \lambda \geq 0 \\ \hline \lambda \in \mathbb{R} \end{array}$$

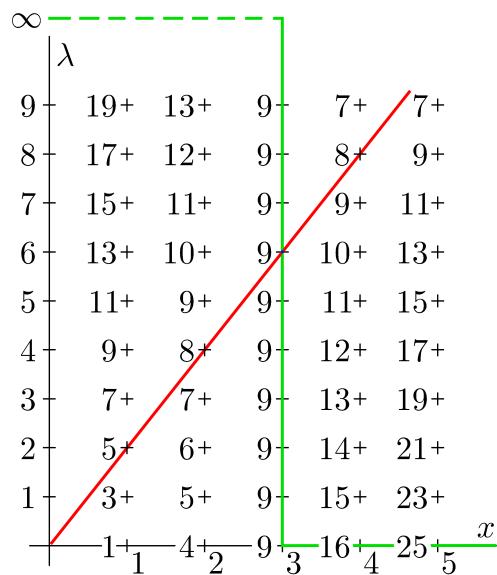
Zjavne riešením P je $x^* = 3$ s hodnotou účelovej funkcie $3^2 = 9$. Naopak, riešením D je $\lambda^* = 6$, kedy $(-\frac{\lambda^2}{4} + 3\lambda)' = 0$ a zároveň $\lambda \geq 0$, s hodnotou účelovej funkcie $-\frac{6^2}{4} + 3 \cdot 6 = 9$. Čiže platí silná dualita.

Teraz už poznáme funkciu $\psi(\lambda) = \inf_x \{L(x, \lambda)\}$ a vieme, že vtedy $x = \frac{\lambda}{2}$. Ako však vyzerá $\sup_{\lambda \geq 0} \{L(x, \lambda)\}$? Platí

$$\sup_{\lambda \geq 0} \{L(x, \lambda)\} = \sup_{\lambda \geq 0} \{x^2 + \lambda(3-x)\} = \begin{cases} \infty & \text{ak } 3-x > 0 \\ 9 & \text{ak } 3-x = 0 \\ x^2 & \text{ak } 3-x < 0 \end{cases}$$

totiž v prvom prípade $\lambda \rightarrow \infty$, v druhom môže mať λ ľubovoľnú nezápornú hodnotu, a v treťom $\lambda = 0$.

Naše zistenia sme zakreslili do Obrázku 18, ktorý má iné škálovanie na osi x než na osi λ . Kótami sme označili hodnotu $L(x, \lambda)$ pre mrežové body $(x, \lambda) \in \mathbb{Z}^2$, $1 \leq x \leq 5$ a $0 \leq \lambda \leq 9$. Grafom $L(x, \lambda)$ je hyperbolický paraboloid, čo je priamková plocha. Jedna siet priamok je nad pôdorysom $x = c$ a druhá nad $x - \lambda = c$. Červenou čiarou sme vyznačili $\psi(\lambda)$ a zelenou $\sup_{\lambda \geq 0} \{L(x, \lambda)\}$ ako časť $L(x, \lambda)$. (Pripomeňme, že pre $x < 3$ je $\sup_{\lambda} \{L(x, \lambda)\} = \infty$ pre $\lambda \rightarrow \infty$.)



Obrázok 18

Všimnime si, že hoci má Lagrangeova funkcia z predchádzajúceho príkladu v $(3, 6)$ sedlový bod, tak $\psi(\lambda)$ a najmä $\sup_{\lambda \geq 0} \{L(x, \lambda)\}$ majú komplikovanejší priebeh. Podobne komplikované to bude vždy, lebo $L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu})$ je v $(\boldsymbol{\lambda}, \boldsymbol{\nu})$ lineárna funkcia.

Využitie duality

Dualita dáva dobrú podmienku na zastavenie iteračného algoritmu. Totiž ak je \mathbf{x} prípustné riešenie primárnej úlohy P a $(\boldsymbol{\lambda}, \boldsymbol{\nu})$ je prípustné riešenie duálnej úlohy D, pričom $f(\mathbf{x}) - \psi(\boldsymbol{\lambda}, \boldsymbol{\nu}) < \varepsilon$, tak algoritmus môžeme ukončiť.

Duálna úloha D môže byť jednoduchšia ako primárna úloha P. To preto, že $\psi(\boldsymbol{\lambda}, \boldsymbol{\nu})$ je za každých okolností konkávna, čiže

$$\begin{array}{c} \min -\psi(\boldsymbol{\lambda}, \boldsymbol{\nu}) \\ \hline -\boldsymbol{\lambda} \leq \mathbf{0} \\ \hline \boldsymbol{\lambda} \in \mathbb{R}^\ell, \boldsymbol{\nu} \in \mathbb{R}^m \end{array}$$

je úloha konvexného programovania. Problém je, že $\psi(\boldsymbol{\lambda}, \boldsymbol{\nu})$ vo všeobecnosti (ak nie sú funkcie f , g_i a h_j „pekné“) nedokážeme analyticky vyjadriť. Vieme však, že ak sú v P iba ohraničenia tvaru nerovníc a $\tilde{\mathbf{x}}$ je riešenie úlohy $\inf_{\mathbf{x} \in D} \{L(\mathbf{x}, \boldsymbol{\lambda})\}$, teda keď $\psi(\boldsymbol{\lambda}) = f(\tilde{\mathbf{x}}) + \sum_{i=1}^{\ell} \lambda_i \cdot g_i(\tilde{\mathbf{x}})$, tak

$$\nabla \psi(\boldsymbol{\lambda}) = (g_1(\tilde{\mathbf{x}}), \dots, g_\ell(\tilde{\mathbf{x}}))^T.$$

Na tomto pozorovaní je založený primárno-duálny algoritmus Uzawova metóda, pozri Kapitolu 8.

Karushove-Kuhnove-Tuckerove podmienky

Ukážeme si, že ak pre úlohy P a D platí silná dualita, tak sú KKT podmienky vo vnútornom bode definičného oboru D splnené.

Nech je \mathbf{x}^* optimálne riešenie primárnej úlohy P, pričom \mathbf{x}^* je vnútorným bodom D, a nech je $(\boldsymbol{\lambda}^*, \boldsymbol{\nu}^*)$ optimálne riešenie duálnej úlohy D. Potom platí

$$\begin{aligned} f(\mathbf{x}^*) &= \psi(\boldsymbol{\lambda}^*, \boldsymbol{\nu}^*) = \inf_{\mathbf{x} \in \mathbb{R}^n} \left\{ f(\mathbf{x}) + \sum_{i=1}^{\ell} \lambda_i^* \cdot g_i(\mathbf{x}) + \sum_{j=1}^m \nu_j^* \cdot h_j(\mathbf{x}) \right\} \leq \\ &\leq f(\mathbf{x}^*) + \sum_{i=1}^{\ell} \lambda_i^* \cdot g_i(\mathbf{x}^*) + \sum_{j=1}^m \nu_j^* \cdot h_j(\mathbf{x}^*) \leq f(\mathbf{x}^*). \end{aligned}$$

Tu prvá nerovnica je splnená triviálne a druhá platí preto, že $\lambda_i^* \cdot g_i(\mathbf{x}^*) \leq 0$ a $\nu_j^* \cdot h_j(\mathbf{x}^*) = 0$. Dostali sme $f(\mathbf{x}^*) \leq f(\mathbf{x}^*)$, čo znamená, že všetky nerovnice musia byť splnené ako rovnice. Teda $\lambda_i^* \cdot g_i(\mathbf{x}^*) = 0$ pre $1 \leq i \leq \ell$. Naviac, $\inf_{\mathbf{x} \in \mathbb{R}^n} \{L(\mathbf{x}, \boldsymbol{\lambda}^*, \boldsymbol{\nu}^*)\}$ sa dosahuje pre konkrétny bod $\mathbf{x} = \mathbf{x}^*$, čo spolu s tým, že \mathbf{x}^* je vnútorným bodom D znamená, že v \mathbf{x}^* sú všetky parciálne derivácie $L(\mathbf{x}, \boldsymbol{\lambda}^*, \boldsymbol{\nu}^*)$ podľa x -ov nulové. Čiže $\nabla_{\mathbf{x}} L(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\nu}^*) = \mathbf{0}$. Získavame nutné podmienky na to, aby bolo prípustné riešenie \mathbf{x}^* , ktoré leží vo vnútri definičného oboru

D , optimálnym riešením P. Musia existovať $\boldsymbol{\lambda}^* \in \mathbb{R}_+^\ell$ a $\boldsymbol{\nu}^* \in \mathbb{R}^m$ také, že

$$\begin{aligned}\nabla_{\mathbf{x}} L(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\nu}^*) &= \mathbf{0} \\ g_i(\mathbf{x}^*) &\leq 0 & 1 \leq i \leq \ell \\ \lambda_i^* &\geq 0 & 1 \leq i \leq \ell \\ \lambda_i^* \cdot g_i(\mathbf{x}^*) &= 0 & 1 \leq i \leq \ell \\ h_j(\mathbf{x}^*) &= 0 & 1 \leq j \leq m\end{aligned}\tag{13}$$

čo sú práve KKT podmienky. Treba však zdôrazniť, že tu sme ich odvodili za predpokladu, že platí silná dualita.

Cvičenia

CVIČENIE 5.1. Ukážte, že keď zostavíme duálnu úlohu tak, ako sme ju definovali v tejto kapitole pre úlohu lineárneho programovania, tak dostaneme presne tú úlohu, ktorú sme označili za duálnu v prvej kapitole.

CVIČENIE 5.2. Majme úlohu

$$\begin{array}{c}\min x \\ \hline x^2 \leq 0 \\ \hline x \in \mathbb{R}\end{array}$$

Všimnite si, že sa jedná o úlohu konvexného programovania, ktorá nemá vnútorný prípustný bod. Zostrojte $\psi(\lambda) = \inf_x L(x, \lambda)$ a nájdite optimálne riešenie primárnej aj duálnej úlohy. Určte $\sup_\lambda L(x, \lambda)$ a zostrojte diagram analogický Obrázku 18. Má Lagrangeova funkcia $L(x, \lambda)$ sedlový bod? Sú pre optimálne riešenie x^* splnené KKT podmienky?

CVIČENIE 5.3. Platí v Cvičeniach 4.3, 4.4, 4.5 a 4.6 silná dualita?

CVIČENIE 5.4. Majme úlohu konvexného programovania

$$\begin{array}{c}\min x^2 + 1 \\ \hline (x - 2)(x - 5) \leq 0\end{array}$$

1. Pre primárnu úlohu

- a) Určte účelovú funkciu f a funkcie ohraničení g a h .
- b) Nájdite množinu prípustných riešení \mathcal{F} , optimálne riešenie x^* a hodnotu účelovej funkcie v x^* .
- c) Vykreslite v jednom obrázku $f(x)$, \mathcal{F} , x^* a $p^* = f(x^*)$.

2. Pre Lagrangeovu funkciu

- a) Určte Lagrangeovu funkciu $L(x, \lambda)$ a vykreslite ju.

- b) Vykreslite grafy funkcií $f(x)$ a $\psi(\lambda)$ ako krivky $(x, 0, f(x))$ a $(x(\lambda), \lambda, \psi(\lambda))$, pozri duálnu úlohu nižšie.
- c) Graficky overte, že platí $p^* \geq \psi(\lambda)$ pre všetky $\lambda \geq 0$.
- d) Vyreslite $L(x, \lambda)$ aj v obrázku pre primárnu úlohu pomocou Manipulate cez λ .

3. Pre duálnu úlohu

- a) Sformulujte duálnu úlohu s účelovou funkciou ψ .
- b) Vykreslite funkciu $\psi(\lambda)$.
- c) Vypočítajte a vykreslite duálne optimálne riešenie λ^* a hodnotu d^* duálnej účelovej funkcie v λ^* .
- d) Overte, či platí silná dualita $p^* = d^*$.

4. Nájdite primárne a duálne optimá aj pomocou KKT podmienok.

CVIČENIE 5.5. Majme úlohu konvexného programovania

$$\begin{array}{c} \min x_1^2 + x_2^2 \\ \hline (x_1 - 1)^2 + (x_2 - 1)^2 \leq r^2 \\ (x_1 - 1)^2 + (x_2 + 1)^2 \leq r^2 \end{array}$$

V závislosti od voľby parametra r sú tri možnosti:

- Optimálne riešenie neexistuje (množina prípustných riešení je prázdna).
- Optimálne riešenie leží na hranici množiny prípustných riešení.
- Optimálne riešenie leží vo vnútri množiny prípustných riešení.

Zvoľte parameter r tak, aby optimálne riešenie ležalo na hranici množiny prípustných riešení, ale aby sa nedosahovalo v bode $(0, 0)$, čiže aby sa nedosahovalo v globálnom minime účelovej funkcie. Ďalej pokračujte ako v Cvičení 5.4 a vykreslite všetko, čo dokážete. Lagrangeovu funkciu $L(x_1, x_2, \lambda_1, \lambda_2)$ vykreslite pomocou Manipulate cez λ_1, λ_2 spolu s účelovou funkciou $f(x_1, x_2)$ a množinou prípustných riešení \mathcal{F} .

6 Úlohy s ohraničeniami tvaru rovníc

V tejto kapitole budeme riešiť úlohy konvexného programovania s lineárnymi ohraničeniami tvaru rovníc. Najprv však budeme riešiť úlohu bez ohraničení

$$\begin{array}{c} \min f(\mathbf{x}) \\ \mathbf{x} \in D_f \end{array}$$

kde f je spojité, konvexná, dvakrát spojito diferencovateľná funkcia a D_f je konvexná množina.

Newtonova metóda

Podobne ako v úlohách lineárneho programovania, naše algoritmy budú iteračné a budú založené na Newtonovej metóde. Predpokladajme, že sme v bode \mathbf{x} , ktorý sme dostali v poslednej iterácii a chceme z neho nájsť nový bod s nižšou hodnotou funkcie f . Keďže f je dvakrát spojito diferencovateľná funkcia, môžeme ju approximovať pomocou Taylorovho rozvoja druhého rádu

$$\hat{f}(\mathbf{x} + \mathbf{d}) = f(\mathbf{x}) + \nabla f(\mathbf{x})^T \cdot \mathbf{d} + \frac{1}{2} \mathbf{d}^T \cdot \nabla^2 f(\mathbf{x}) \cdot \mathbf{d}. \quad (14)$$

V ďalšom nebudeme úlohu riešiť pre funkciu f , ale pre kvadratickú funkciu \hat{f} s n -rozmernou premennou \mathbf{d} , pretože táto funkcia je jednoduchšia. Ak chceme nájsť smer \mathbf{d} minimalizujúci $\hat{f}(\mathbf{x} + \mathbf{d})$, musia byť všetky parciálne derivácie rovné 0. Zo vzťahu (14) dostávame

$$\nabla_{\mathbf{d}} \hat{f}(\mathbf{x} + \mathbf{d}) = \nabla f(\mathbf{x}) + \nabla^2 f(\mathbf{x}) \cdot \mathbf{d} = \mathbf{0}$$

z čoho získame

$$\mathbf{d} = -\nabla^2 f(\mathbf{x})^{-1} \cdot \nabla f(\mathbf{x}).$$

Túto hodnotu nazývame **Newtonov krok** (**Newton step**) a značíme ju $\Delta \mathbf{x}_{nt}$. Teda

$$\Delta \mathbf{x}_{nt} = -\nabla^2 f(\mathbf{x})^{-1} \cdot \nabla f(\mathbf{x}).$$

Znamená to, že $\Delta \mathbf{x}_{nt}$ získame ako riešenie rovnice

$$\nabla^2 f(\mathbf{x}) \cdot \Delta \mathbf{x}_{nt} = -\nabla f(\mathbf{x}). \quad (15)$$

Teraz určíme, aké vylepšenie nám ponúka naša approximácia, keď sa posunieme z bodu \mathbf{x} do $\mathbf{x} + \Delta \mathbf{x}_{nt}$, pozri Obrázok 19. Keďže f je dvakrát spojito diferencovateľná, Hessova matica f je symetrická, čiže $\nabla^2 f(\mathbf{x})^T = \nabla^2 f(\mathbf{x})$. Lenže potom je

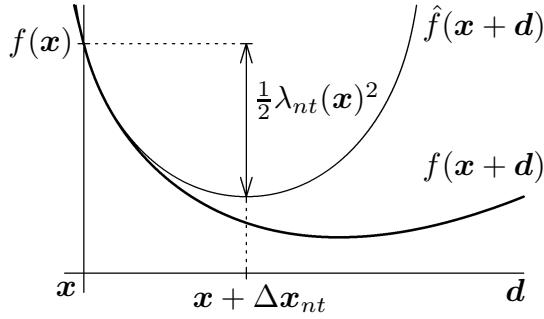
symetrická aj $\nabla^2 f(\mathbf{x})^{-1}$, čiže $(\nabla^2 f(\mathbf{x})^{-1})^T = \nabla^2 f(\mathbf{x})^{-1}$. Máme

$$\begin{aligned} f(\mathbf{x}) - \hat{f}(\mathbf{x} + \Delta \mathbf{x}_{nt}) &= f(\mathbf{x}) - \left[f(\mathbf{x}) + \nabla f(\mathbf{x})^T \cdot (-\nabla^2 f(\mathbf{x})^{-1} \cdot \nabla f(\mathbf{x})) + \right. \\ &\quad \left. + \frac{1}{2} (-\nabla^2 f(\mathbf{x})^{-1} \cdot \nabla f(\mathbf{x}))^T \cdot \nabla^2 f(\mathbf{x}) \cdot (-\nabla^2 f(\mathbf{x})^{-1} \cdot \nabla f(\mathbf{x})) \right] \\ &= \nabla f(\mathbf{x})^T \cdot \nabla^2 f(\mathbf{x})^{-1} \cdot \nabla f(\mathbf{x}) - \frac{1}{2} \nabla f(\mathbf{x})^T \cdot \nabla^2 f(\mathbf{x})^{-1} \cdot \nabla f(\mathbf{x}) \\ &= \frac{1}{2} \nabla f(\mathbf{x})^T \cdot \nabla^2 f(\mathbf{x})^{-1} \cdot \nabla f(\mathbf{x}). \end{aligned}$$

Hodnotu

$$\lambda_{nt}(\mathbf{x}) = \sqrt{\nabla f(\mathbf{x})^T \cdot \nabla^2 f(\mathbf{x})^{-1} \cdot \nabla f(\mathbf{x})}$$

nazývame **Newtonov decrement**. Platí teda, že $f(\mathbf{x}) - \hat{f}(\mathbf{x} + \Delta \mathbf{x}_{nt}) = \frac{1}{2} \lambda_{nt}(\mathbf{x})^2$. Situácia je ozrejmená na Obrázku 19, kde \mathbf{d} predstavuje ľubovoľný násobok Newtonovho kroku, čiže $\mathbf{d} = t \cdot \Delta \mathbf{x}_{nt}$.



Obrázok 19

Newtonov decrement možno vyjadriť aj pomocou Newtonovho kroku. Z výrazu (15) získame

$$\nabla f(\mathbf{x}) = -\nabla^2 f(\mathbf{x}) \cdot \Delta \mathbf{x}_{nt}$$

a dosadením do $\lambda_{nt}(\mathbf{x})^2$ dostávame

$$\begin{aligned} \lambda_{nt}(\mathbf{x})^2 &= (\nabla^2 f(\mathbf{x}) \cdot \Delta \mathbf{x}_{nt})^T \cdot \nabla^2 f(\mathbf{x})^{-1} \cdot (\nabla^2 f(\mathbf{x}) \cdot \Delta \mathbf{x}_{nt}) \\ &= \Delta \mathbf{x}_{nt}^T \cdot \nabla^2 f(\mathbf{x})^T \cdot \Delta \mathbf{x}_{nt} \\ &= \Delta \mathbf{x}_{nt}^T \cdot \nabla^2 f(\mathbf{x}) \cdot \Delta \mathbf{x}_{nt}. \end{aligned}$$

Teda

$$\lambda_{nt}(\mathbf{x}) = \sqrt{\Delta \mathbf{x}_{nt}^T \cdot \nabla^2 f(\mathbf{x}) \cdot \Delta \mathbf{x}_{nt}}.$$

Ked' teraz prenásobíme (15) zľava $\Delta \mathbf{x}_{nt}^T$, dostávame

$$\lambda_{nt}(\mathbf{x})^2 = \Delta \mathbf{x}_{nt}^T \cdot \nabla^2 f(\mathbf{x}) \cdot \Delta \mathbf{x}_{nt} = -\Delta \mathbf{x}_{nt}^T \cdot \nabla f(\mathbf{x}).$$

Postup, ktorý sme opísali, dáva nasledujúci algoritmus.

ALGORITMUS: Newtonov algoritmus.

Vstup: $\mathbf{x} \in D_f$, presnosť $\varepsilon > 0$.

Begin

Repeat Begin

Vyrieš $\nabla^2 f(\mathbf{x}) \cdot \Delta \mathbf{x}_{nt} = -\nabla f(\mathbf{x})$;

$\lambda_{nt}(\mathbf{x})^2 = -\nabla f(\mathbf{x})^T \cdot \Delta \mathbf{x}_{nt}$;

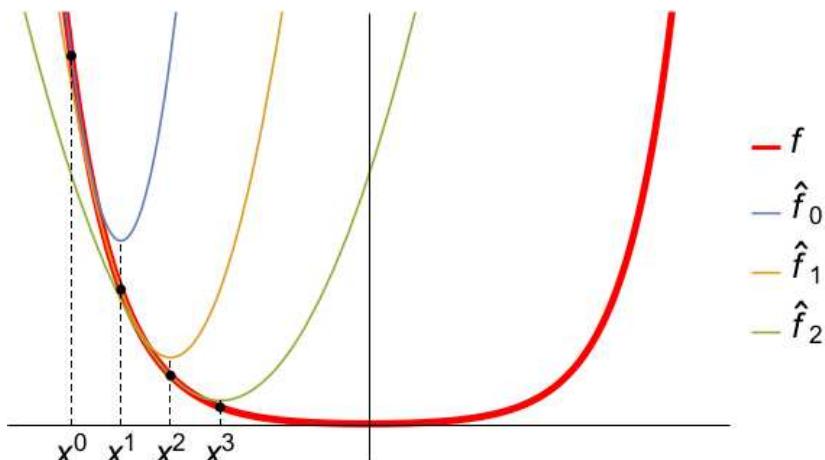
Quit If $\frac{\lambda_{nt}(\mathbf{x})^2}{2} < \varepsilon$;

$\mathbf{x} = \mathbf{x} + \Delta \mathbf{x}_{nt}$;

End;

End.

Jednotlivé iterácie Newtonovho algoritmu sú znázornené na Obrázku 20 pre funkciu jednej premennej $f(x) = e^{-x} + e^x$, pričom sme za počiatočný bod zvolili $x^0 = -6$. Ďalšími iteráciami, ktoré sme získali kvadratickými aproximáciami, sú $x^1 = -5,00001$, $x^2 = -4,0001$ a $x^3 = -3,00077$.



Obrázok 20

Poznamenajme, že klasický Newtonov algoritmus hľadá nulový bod funkcie pomocou derivácie. Zdalo by sa, že tu používame iný algoritmus, pretože nehľadáme nulový bod, ale extrém, naviac používame nielen prvé, ale aj druhé parciálne derivácie. Avšak my v skutočnosti hľadáme kritický bod funkcie $f(\mathbf{x})$, čiže riešime rovnicu $\nabla f(\mathbf{x}) = \mathbf{0}$, kde $\nabla f(\mathbf{x})$ je viachodnotová funkcia. Keď teraz vo vzťahu (3) (pozri kapitolu 2) zvolíme $F(\mathbf{x}) = \nabla f(\mathbf{x})$, tak $DF(\mathbf{x})$ bude Hessova matica $\nabla^2 f(\mathbf{x})$, čiže dostaneme vzťah (15).

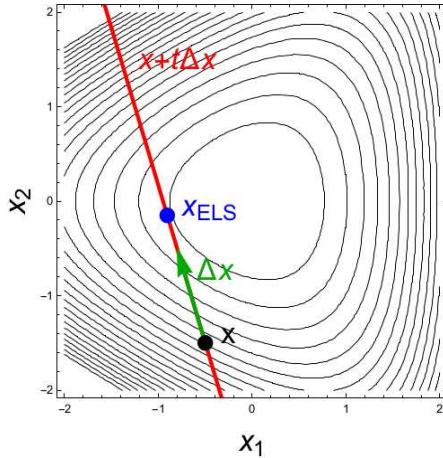
Backtracking line search

Priama Newtonova metóda, tak ako sme ju opísali v predchádzajúcej časti, sa nepoužíva. Funkcia \hat{f} je iba aproximáciou f , preto nemusí byť najvhodnejšie posunúť za z \mathbf{x} do bodu $\mathbf{x} + \Delta \mathbf{x}_{nt}$. Ak chceme získať bod s najlepšou hodnotou účelovej

funkcie, mali by sme vyriešiť úlohu

$$\frac{\min f(\mathbf{x} + t \cdot \Delta \mathbf{x}_{nt})}{t \geq 0}$$

čo je optimalizačná úloha funkcie jednej premennej t , a posunúť sa z \mathbf{x} do bodu $\mathbf{x} + t^* \cdot \Delta \mathbf{x}_{nt}$. Takýto postup sa nazýva **exact line search** a využíva sa napríklad pri gradientnej metóde s optimálnym parametrom. Exact line search je znázornený na Obrázku 21 pre funkciu $f(x_1, x_2) = e^{0,5(x_1 - \sqrt{3}x_2)} + e^{0,5(\sqrt{3}x_1 - x_2)} + e^{x_1}$, pričom $\mathbf{x} = (-0,5; -1,5)$ a $\Delta \mathbf{x} = (-0,3; 1)$. Bod \mathbf{x}_{ELS} je hľadané optimum.



Obrázok 21

My budeme používať iný postup, ktorý nevyžaduje riešenie optimalizačnej úlohy. Budeme používať Backtracking line search.

ALGORITMUS: Backtracking line search.

Vstup: $\mathbf{x} \in D_f$, smer $\Delta \mathbf{x}$, $\alpha \in (0, \frac{1}{2})$ a $\beta \in (0, 1)$.

Begin

$t = 1;$

 While $f(\mathbf{x} + t \cdot \Delta \mathbf{x}) > f(\mathbf{x}) + \alpha \cdot t \cdot \nabla f(\mathbf{x})^T \cdot \Delta \mathbf{x}$ Do Begin

$t = \beta \cdot t;$

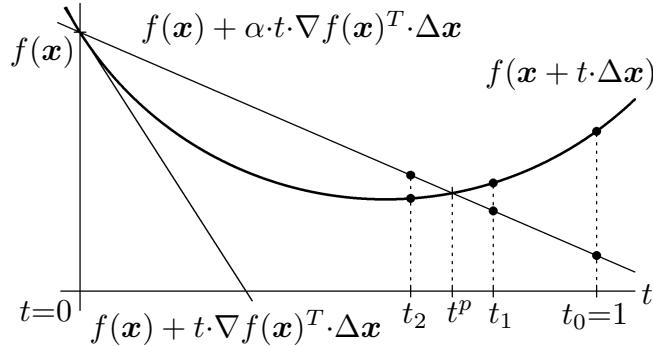
 End;

End.

Princíp práce algoritmu Backtracking line search je znázornený na Obrázku 22. Priamka $y = f(\mathbf{x}) + \nabla f(\mathbf{x})^T \cdot (t \cdot \Delta \mathbf{x})$, čiže $y = f(\mathbf{x}) + t \cdot \nabla f(\mathbf{x})^T \cdot \Delta \mathbf{x}$ v premennej t je dotyčnica ku grafu funkcie f v bode \mathbf{x} . Teraz ak zmiernime sklon tejto priamky (všimnime si, že $\alpha < 1$, dokonca platí $0 < \alpha < \frac{1}{2}$), tak keďže f je konvexná, priamka $y = f(\mathbf{x}) + \alpha \cdot t \cdot \nabla f(\mathbf{x})^T \cdot \Delta \mathbf{x}$ môže preťať graf funkcie f . Algoritmus potom nájde takú hodnotu parametra t , pre ktorú $f(\mathbf{x}) + \alpha \cdot t \cdot \nabla f(\mathbf{x})^T \cdot \Delta \mathbf{x}$ nie je pod grafom funkcie f .

Algoritmus skončí buď hneď ako začne pre $t = 1$ (to sa stane v prípade, keď je $\Delta \mathbf{x}$ malé), alebo skončí pre $t \in (\beta \cdot t^p, t^p)$, kde t^p je hodnota, v ktorej sa grafy funkcie $f(\mathbf{x} + t \cdot \Delta \mathbf{x})$ a priamky $f(\mathbf{x}) + \alpha \cdot t \cdot \nabla f(\mathbf{x})^T \cdot \Delta \mathbf{x}$ pretnú. Iterácie algoritmu

Backtracking line search sú znázornené na Obrázku 22, pričom $t_1 = \beta \cdot t_0$, $t_2 = \beta \cdot t_1$ a porovnávame hodnoty v čiernych bodoch. V tomto prípade algoritmus skončí pre $t = t_2$.



Obrázok 22

Všimnime si, že keď $\Delta\mathbf{x}$ bude Newtonov krok, tak v Backtracking line search testujeme podmienku

$$f(\mathbf{x} + t \cdot \Delta\mathbf{x}_{nt}) > f(\mathbf{x}) + \alpha \cdot t \cdot \nabla f(\mathbf{x})^T \cdot \Delta\mathbf{x}_{nt},$$

pričom $\nabla f(\mathbf{x})^T \cdot \Delta\mathbf{x}_{nt} = -\lambda_{nt}(\mathbf{x})^2$, kde $\lambda_{nt}(\mathbf{x})$ je Newtonov decrement.

Tlmený Newtonov algoritmus

Teraz už môžeme uviesť iteratívny algoritmus, ktorý rieši minimalizačnú úlohu bez ohraničení. Tento algoritmus je založený na Newtonovej metóde a využíva Backtracking line search.

ALGORITMUS: Tlmený (damped) Newtonov algoritmus.

Vstup: $\mathbf{x} \in D_f$, presnosť $\varepsilon > 0$.

Begin

Repeat Begin

Vyrieš $\nabla^2 f(\mathbf{x}) \cdot \Delta\mathbf{x}_{nt} = -\nabla f(\mathbf{x})$;

$\lambda_{nt}(\mathbf{x})^2 = -\nabla f(\mathbf{x})^T \cdot \Delta\mathbf{x}_{nt}$;

Quit If $\frac{\lambda_{nt}(\mathbf{x})^2}{2} < \varepsilon$;

Urči t pomocou Backtracking line search;

$\mathbf{x} = \mathbf{x} + t \cdot \Delta\mathbf{x}_{nt}$;

End;

End.

V ďalšom budeme upravovať tlmený Newtonov algoritmus, čiže budeme aj ďalej využívať Backtracking line search.

Úlohy s lineárnymi ohraničeniami tvaru rovníc

Teraz prejdeme na úlohy s lineárnymi ohraničeniami tvaru rovníc

$$\begin{array}{c} \min f(\mathbf{x}) \\ \hline \mathbb{A} \cdot \mathbf{x} = \mathbf{b} \\ \hline \mathbf{x} \in D_f \end{array}$$

Tu \mathbb{A} je matica typu $(m \times n)$, pričom $m \leq n$, f je dvakrát spojito diferencovateľná konvexná funkcia a D_f je konvexná množina. Lagrangeova funkcia je

$$L(\mathbf{x}, \boldsymbol{\nu}) = f(\mathbf{x}) + \boldsymbol{\nu}^T \cdot (\mathbb{A} \cdot \mathbf{x} - \mathbf{b}).$$

Kedže naša úloha konvexného programovania nemá ohraničenia tvaru nerovníc, tak splňa Slaterovu podmienku (pozri predchádzajúcu kapitolu). Teda pre ňu platí silná dualita, čo znamená, že optimálne riešenia primárnej úlohy \mathbf{x}^* a duálnej úlohy $\boldsymbol{\nu}^*$ spĺňajú KKT podmienky (13) ak je \mathbf{x}^* vnútorný bod D_f . Potom pre $(\mathbf{x}^*, \boldsymbol{\nu}^*)$ platí

$$\begin{aligned} \nabla_{\mathbf{x}} L(\mathbf{x}^*, \boldsymbol{\nu}^*) &= \nabla_{\mathbf{x}} f(\mathbf{x}^*) + \mathbb{A}^T \cdot \boldsymbol{\nu}^* = \mathbf{0} \\ \mathbb{A} \cdot \mathbf{x}^* &= \mathbf{b} \end{aligned} \tag{16}$$

Všimnime si, že gradienty ohraničení sú riadky matice \mathbb{A} . Teda naša úloha splňa LICQ podmienku práve vtedy, keď sú riadky \mathbb{A} lineárne nezávislé. Nuž a podľa vety na strane 40 keď úloha konvexného programovanie splňa LICQ podmienku a \mathbf{x} je vnútorný bod D_f , tak sú KKT podmienky nielen nutné, ale aj postačujúce na existenciu globálneho extrému.

Prípad konvexnej kvadratickej funkcie

Najprv budeme úlohu riešiť v prípade, že je f konvexná kvadratická funkcia. Vtedy

$$f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \cdot \mathbb{P} \cdot \mathbf{x} + \mathbf{q}^T \cdot \mathbf{x} + r,$$

kde \mathbb{P} je symetrická kladne definitná matica typu $(n \times n)$, čo znamená, že pre každé $z \in \mathbb{R}^n \setminus \{\mathbf{0}\}$ platí $z^T \cdot \mathbb{P} \cdot z > 0$, ďalej \mathbf{q} je vektor z \mathbb{R}^n a $r \in \mathbb{R}$. Všimnime si, že $D_f = \mathbb{R}^n$.

Z KKT podmienok (16) dostávame, že pre optimálne riešenie $(\mathbf{x}^*, \boldsymbol{\nu}^*)$ má platiť

$$\begin{aligned} \nabla_{\mathbf{x}} L(\mathbf{x}^*, \boldsymbol{\nu}^*) &= \nabla_{\mathbf{x}} f(\mathbf{x}^*) + \mathbb{A}^T \cdot \boldsymbol{\nu}^* = \mathbb{P} \cdot \mathbf{x}^* + \mathbf{q} + \mathbb{A}^T \cdot \boldsymbol{\nu}^* = \mathbf{0} \\ \mathbb{A} \cdot \mathbf{x}^* &= \mathbf{b} \end{aligned}$$

čiže

$$\begin{pmatrix} \mathbb{P} & \mathbb{A}^T \\ \mathbb{A} & \mathbb{0} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{x}^* \\ \boldsymbol{\nu}^* \end{pmatrix} = \begin{pmatrix} -\mathbf{q} \\ \mathbf{b} \end{pmatrix},$$

čo je systém $n + m$ lineárnych rovníc o $n + m$ neznámych. Ak je matica systému regulárna, tak úloha splňa LICQ podmienku a jediným riešením je

$$\begin{pmatrix} \mathbf{x}^* \\ \boldsymbol{\nu}^* \end{pmatrix} = \begin{pmatrix} \mathbb{P} & \mathbb{A}^T \\ \mathbb{A} & \mathbb{0} \end{pmatrix}^{-1} \cdot \begin{pmatrix} -\mathbf{q} \\ \mathbf{b} \end{pmatrix}.$$

Ak však matica systému nie je regulárna, tak úloha LICQ podmienku nespĺňa. V takom prípade ak systém má riešenie, tak všetky riešenia sú optimálne, a ak systém riešenie nemá, tak úloha nemá žiadne prípustné riešenie.

Zdôrazníme skutočnosť, že ak je matica vyššie uvedeného systému regulárna, tak optimálne riešenie úlohy získame ako riešenie systému lineárnych rovníc, čiže veľmi jednoducho.

Newtonova metóda pre lineárne ohraničenia tvaru rovníc

Opäť by sme mohli postupovať tak, ako pri odvodení algoritmu pre úlohu bez ohraničení. Čiže mohli by sme spraviť approximáciu \hat{f} účelovej funkcie f pomocou Taylorovho polynómu druhého rádu a touto approximáciou by sme nahradili f v Lagrangeovej funkcií. Teraz však použijeme priamo Newtonovu metódu v tvare, v akom sme ju uviedli v Kapitole 2.

Majme viachodnotovú funkciu $F(\mathbf{y})$, pričom hľadáme jej nulový bod. Teda riešime rovnicu $F(\mathbf{y}) = \mathbf{0}$. Newtonov algoritmus navrhuje posunúť sa z bodu \mathbf{y} v smere Newtonovho kroku $\Delta\mathbf{y}_{nt}$, pričom $\Delta\mathbf{y}_{nt}$ je riešením sústavy lineárnych rovníc (3)

$$DF(\mathbf{y}) \cdot \Delta\mathbf{y}_{nt} = -F(\mathbf{y}),$$

kde $DF(\mathbf{y})$ je Jacobiho matica funkcie F v bode \mathbf{y} .

Teraz aplikujeme tento postup na našu úlohu. Podľa (16) hľadáme nulový bod funkcie $F(\mathbf{x}, \boldsymbol{\nu})$, kde

$$F(\mathbf{x}, \boldsymbol{\nu}) = \begin{pmatrix} \nabla f(\mathbf{x}) + \mathbb{A}^T \cdot \boldsymbol{\nu} \\ \mathbb{A} \cdot \mathbf{x} - \mathbf{b} \end{pmatrix}.$$

Jacobiho matica funkcie F je

$$DF(\mathbf{x}, \boldsymbol{\nu}) = \begin{pmatrix} \nabla^2 f(\mathbf{x}) & \mathbb{A}^T \\ \mathbb{A} & 0 \end{pmatrix}.$$

Znamená to, že ak je \mathbf{x} prípustné riešenie, teda ak \mathbf{x} splňa rovnicu $\mathbb{A} \cdot \mathbf{x} = \mathbf{b}$, tak Newtonov krok $(\Delta\mathbf{x}_{nt}, \Delta\boldsymbol{\nu}_{nt})$ získame vyriešením sústavy lineárnych rovníc

$$\begin{pmatrix} \nabla^2 f(\mathbf{x}) & \mathbb{A}^T \\ \mathbb{A} & 0 \end{pmatrix} \cdot \begin{pmatrix} \Delta\mathbf{x}_{nt} \\ \Delta\boldsymbol{\nu}_{nt} \end{pmatrix} = -\begin{pmatrix} \nabla f(\mathbf{x}) + \mathbb{A}^T \boldsymbol{\nu} \\ \mathbf{0} \end{pmatrix}.$$

Všimnime si, že ak $\mathbb{A} \cdot \mathbf{x} = \mathbf{b}$, tak potom pre ľubovoľné t platí aj $\mathbb{A} \cdot (\mathbf{x} + t \cdot \Delta\mathbf{x}_{nt}) = \mathbf{b}$. Čiže aj ďalší bod bude prípustným riešením.

Newtonov decrement bude rovnaký ako v prípade bez ohraničení.

$$\lambda_{nt}(\mathbf{x}) = \sqrt{-\nabla f(\mathbf{x})^T \cdot \Delta\mathbf{x}_{nt}}. \quad (17)$$

Opäť platí, že $f(\mathbf{x}) - \inf_{\mathbf{d}} \{\hat{f}(\mathbf{x} + \mathbf{d}); \mathbb{A} \cdot (\mathbf{x} + \mathbf{d}) = \mathbf{b}\} = \lambda_{nt}(\mathbf{x})^2/2$. Dostávame nasledujúci algoritmus.

ALGORITMUS: Newtonova metóda pre úlohy s lineárnymi rovnicami.

Vstup: $\mathbf{x} \in D_f$, pre ktoré $\mathbb{A} \cdot \mathbf{x} = \mathbf{b}$, ďalej $\boldsymbol{\nu}$ (hoci $\boldsymbol{\nu} = \mathbf{0}$) a presnosť $\varepsilon > 0$.

Begin

Repeat Begin

 Urči Newtonov krok $(\Delta\mathbf{x}_{nt}, \Delta\boldsymbol{\nu}_{nt})$ a Newtonov decrement $\lambda_{nt}(\mathbf{x})$;

 Quit If $\frac{\lambda_{nt}(\mathbf{x})^2}{2} < \varepsilon$;

 Urči t pomocou Backtracking line search;

$\mathbf{x} = \mathbf{x} + t \cdot \Delta\mathbf{x}_{nt}$;

End;

End.

Newtonova metóda začínajúca v neprípustnom bode

Teraz náš algoritmus pozmeníme tak, aby sme ho mohli spustiť z bodu $\mathbf{x} \in D_f$, pričom \mathbf{x} nebude nutne prípustný bod. Podľa predchádzajúcej časti máme nájsť nulový bod funkcie

$$F(\mathbf{x}, \boldsymbol{\nu}) = \begin{pmatrix} \nabla f(\mathbf{x}) + \mathbb{A}^T \cdot \boldsymbol{\nu}^* \\ \mathbb{A} \cdot \mathbf{x} - \mathbf{b} \end{pmatrix},$$

pričom Jacobiho matica bude opäť

$$DF(\mathbf{x}, \boldsymbol{\nu}) = \begin{pmatrix} \nabla^2 f(\mathbf{x}) & \mathbb{A}^T \\ \mathbb{A} & 0 \end{pmatrix}.$$

Znamená to, že Newtonov krok $(\Delta\mathbf{x}_{nt}, \Delta\boldsymbol{\nu}_{nt})$ získame vyriešením sústavy lineárnych rovnic

$$\begin{pmatrix} \nabla^2 f(\mathbf{x}) & \mathbb{A}^T \\ \mathbb{A} & 0 \end{pmatrix} \cdot \begin{pmatrix} \Delta\mathbf{x}_{nt} \\ \Delta\boldsymbol{\nu}_{nt} \end{pmatrix} = - \begin{pmatrix} \nabla f(\mathbf{x}) + \mathbb{A}^T \boldsymbol{\nu} \\ \mathbb{A} \cdot \mathbf{x} - \mathbf{b} \end{pmatrix}.$$

V algoritme už nemôžeme používať Newtonov decrement $\lambda_{nt}(\mathbf{x})$, pretože hodnota účelovej funkcie nemusí klesať. (Môže sa dokonca stať, že sa hodnota účelovej funkcie sústavne zvyšuje, avšak postupne prichádzame do množiny prípustných riešení.) Znamená to, že sa Newtonov decrement nemusí zmenšovať. V nejakom kroku, keď sme ešte ďaleko od prípustného riešenia, môže byť Newtonov decrement dokonca nulový (to sa podľa (17) stane vtedy, keď $\Delta\mathbf{x}_{nt} \perp \nabla f(\mathbf{x})$), čo by spôsobilo predčasné ukončenie algoritmu. Budeme preto používať reziduum

$$\mathbf{r}(\mathbf{x}, \boldsymbol{\nu}) = (\mathbf{r}_D(\mathbf{x}, \boldsymbol{\nu}), \mathbf{r}_P(\mathbf{x}, \boldsymbol{\nu}))^\top,$$

kde

$$\mathbf{r}_D(\mathbf{x}, \boldsymbol{\nu}) = \nabla f(\mathbf{x}) + \mathbb{A}^T \cdot \boldsymbol{\nu} \quad \text{a} \quad \mathbf{r}_P(\mathbf{x}, \boldsymbol{\nu}) = \mathbb{A} \cdot \mathbf{x} - \mathbf{b}.$$

Čiže \mathbf{r}_D a \mathbf{r}_P sú reziduá získané z pravej strany sústavy. Tieto reziduá musia klesať, pretože $(\mathbf{x}, \boldsymbol{\nu})$ sa postupne blíži k nulovému bodu funkcie F .

Tiež upravíme Backtracking line search tak, že bude postupovať podľa normy \mathbf{r} .

ALGORITMUS: Newtonova metóda pre rovnice z neprípustného bodu.

Vstup: $\mathbf{x} \in D_f$ a $\boldsymbol{\nu}$ (hoci $\boldsymbol{\nu} = \mathbf{0}$), presnosť $\varepsilon > 0$, $\alpha \in (0, \frac{1}{2})$ a $\beta \in (0, 1)$.

Begin

Repeat

Urči Newtonov krok $(\Delta\mathbf{x}_{nt}, \Delta\boldsymbol{\nu}_{nt})$ vyriešením sústavy lineárnych rovníc

$$\begin{pmatrix} \nabla^2 f(\mathbf{x}) & \mathbb{A}^T \\ \mathbb{A} & \mathbb{0} \end{pmatrix} \cdot \begin{pmatrix} \Delta\mathbf{x}_{nt} \\ \Delta\boldsymbol{\nu}_{nt} \end{pmatrix} = - \begin{pmatrix} \nabla f(\mathbf{x}) + \mathbb{A}^T \boldsymbol{\nu} \\ \mathbb{A} \cdot \mathbf{x} - \mathbf{b} \end{pmatrix}$$

$t = 1;$ { backtracking podľa normy r }

While $\|\mathbf{r}(\mathbf{x} + t \cdot \Delta\mathbf{x}_{nt}, \boldsymbol{\nu} + t \cdot \Delta\boldsymbol{\nu}_{nt})\| > (1 - \alpha t) \cdot \|\mathbf{r}(\mathbf{x}, \boldsymbol{\nu})\|$ Do Begin

$t = \beta \cdot t;$

End;

$\mathbf{x} = \mathbf{x} + t \cdot \Delta\mathbf{x}_{nt}; \quad \boldsymbol{\nu} = \boldsymbol{\nu} + t \cdot \Delta\boldsymbol{\nu}_{nt};$

Until $(\mathbb{A} \cdot \mathbf{x} = \mathbf{b} \text{ And } \|\mathbf{r}(\mathbf{x}, \boldsymbol{\nu})\| < \varepsilon)$;

End.

Ak bude v algoritme po prechode While cyklom čo len raz $t = 1$, (alebo ak tento cyklus raz preskočíme), čiže ak bude $\mathbf{x} = \mathbf{x} + \Delta\mathbf{x}_{nt}$ a $\boldsymbol{\nu} = \boldsymbol{\nu} + \Delta\boldsymbol{\nu}_{nt}$, tak od toho okamihu budú všetky ďalšie iterácie prípustné riešenia. Tiež si všimnime podmienku ukončenia algoritmu. Z hľadiska implementácie na počítači $\mathbb{A} \cdot \mathbf{x} = \mathbf{b}$ znamená, že $\|\mathbb{A} \cdot \mathbf{x} - \mathbf{b}\| \ll \varepsilon$.

Eliminácia rovníc

Úlohu

$$\begin{array}{c} \min f(\mathbf{x}) \\ \hline \mathbb{A} \cdot \mathbf{x} = \mathbf{b} \\ \hline \mathbf{x} \in D_f \end{array}$$

môžeme riešiť aj ináč. Môžeme využiť skutočnosť, že lineárne ohraničenia tvaru rovníc v podstate iba znižujú rozmer (dimenziu) úlohy.

Pripomeňme, že \mathbb{A} je matica typu $(m \times n)$, kde $m \leq n$. Upravme \mathbb{A} pomocou Gaussovej eliminácie na tvar $(\mathbb{I} \ \overline{\mathbb{A}})$. Tu je \mathbb{I} matica typu $(m \times m)$ a $\overline{\mathbb{A}}$ je typu $(m \times (n-m))$. (V skutočnosti môžu byť stĺpce výslednej matice $(\mathbb{I} \ \overline{\mathbb{A}})$ trochu poprehadzované.) Kedže ekvivalentné riadkové operácie na \mathbb{A} zodpovedajú násobeniu \mathbb{A} regulárhou maticou zľava, tak platí $\mathbb{A} = \mathbb{P} \cdot (\mathbb{I} \ \overline{\mathbb{A}})$, kde \mathbb{P} je nejaká regulárna matica. Zvolme

$$\mathbb{F} = \begin{pmatrix} -\overline{\mathbb{A}} \\ \mathbb{I} \end{pmatrix}.$$

Kedže platí

$$\mathbb{A} \cdot \mathbb{F} = \mathbb{P} \cdot (\mathbb{I} \ \overline{\mathbb{A}}) \cdot \begin{pmatrix} -\overline{\mathbb{A}} \\ \mathbb{I} \end{pmatrix} = \mathbb{P} \cdot (-\overline{\mathbb{A}} + \overline{\mathbb{A}}) = \mathbb{P} \cdot \mathbb{0} = \mathbb{0},$$

tak sú riadky \mathbb{F}^T kolmé na riadky \mathbb{A} , pričom \mathbb{F}^T je typu $((n-m) \times n)$. Znamená to, že \mathbb{F}^T je matica, ktorej obraz je jadro \mathbb{A} . Teda $\mathbb{A} \cdot \mathbf{y} = \mathbf{0}$ platí práve vtedy, keď je \mathbf{y}^T lineárna kombinácia riadkov \mathbb{F}^T . Preto je naša úloha s ohraničeniami tvaru rovníc ekvivalentná s úlohou

$$\frac{\min f(\mathbb{F} \cdot \mathbf{z} + \mathbf{x}^0)}{\mathbf{z} \in \mathbb{R}^{n-m} \cap D}$$

ktorá už ohraničenia tvaru rovníc nemá. Tu \mathbf{x}^0 je jedno riešenie sústavy $\mathbb{A} \cdot \mathbf{x} = \mathbf{b}$ a $\mathbb{F} \cdot \mathbf{z}$ je lineárna kombinácia stĺpcov \mathbb{F} (čiže riadkov \mathbb{F}^T). Keď označíme funkciu $\bar{f}(\mathbf{z}) = f(\mathbb{F} \cdot \mathbf{z} + \mathbf{x}^0)$, dostávame, že nás problém je ekvivalentný s úlohou

$$\frac{\min \bar{f}(\mathbf{z})}{\mathbf{z} \in \mathbb{R}^{n-m} \cap D_{\bar{f}}}$$

pričom $D_{\bar{f}}$ ešte potrebujeme určiť. Týmto sa tu zaoberať nebudeme. Ukážeme si, ako možno z riešenia posledne uvádzanej úlohy získať riešenie pôvodnej úlohy. Najprv však odvodíme jeden vzťah.

Nech je $q = q(\mathbf{x})$ funkcia n premenných a \mathbb{E} nech je matica typu $(n \times (n-m))$. Uvažujme $\bar{q}(\mathbf{z}) = q(\mathbb{E} \cdot \mathbf{z} + \mathbf{b})$ a zostrojme gradienty $\nabla_{\mathbf{z}} \bar{q}(\mathbf{z}^\circ)$ a $\nabla_{\mathbf{x}} q(\mathbf{x}^\circ)$ pre \mathbf{x}° a \mathbf{z}° vyhovujúce vzťahu $\mathbf{x}^\circ = \mathbb{E} \cdot \mathbf{z}^\circ$. Platí

$$\begin{aligned} \frac{\partial}{\partial z_i} \bar{q}(\mathbf{z}^\circ) &= \frac{\partial}{\partial z_i} q(e_{1,i} \cdot z_i^\circ + c_1^i, \dots, e_{n,i} \cdot z_i^\circ + c_n^i) = \\ &= \frac{\partial}{\partial x_1} f(\mathbf{x}^\circ) \cdot e_{1,i} + \dots + \frac{\partial}{\partial x_n} f(\mathbf{x}^\circ) \cdot e_{n,i} = (e_{1,i}, \dots, e_{n,i}) \cdot \nabla_{\mathbf{x}} q(\mathbf{x}^\circ) \end{aligned}$$

kde c_j^i sú konštanty, $c_j^i = x_j^\circ - e_{j,i} \cdot z_i^\circ$. (Vo výpočte sme využili známy vzťah $f'_t(g(t), h(t)) = f'_{x_1}(g(t), h(t)) \cdot g'(t) + f'_{x_2}(g(t), h(t)) \cdot h'(t)$.) Kedže tento výpočet je správny pre každé i , $1 \leq i \leq n-m$, dostávame $\nabla_{\mathbf{z}} \bar{q}(\mathbf{z}^\circ) = \mathbb{E}^T \cdot \nabla_{\mathbf{x}} q(\mathbf{x}^\circ)$. Pre našu úlohu z toho plynie, že $\nabla_{\mathbf{z}} \bar{f}(\mathbf{z}^\circ) = \mathbb{F}^T \cdot \nabla_{\mathbf{x}} f(\mathbf{x}^\circ)$, ak $\mathbf{x}^\circ = \mathbb{F} \cdot \mathbf{z}^\circ + \mathbf{x}^0$.

Teraz nech je \mathbf{z}^* optimálne riešenie redukovanej úlohy bez ohraničení. Potom pre optimálne riešenie $(\mathbf{x}^*, \boldsymbol{\nu}^*)$ pôvodnej úlohy a jej duálu platí

$$\mathbf{x}^* = \mathbb{F} \cdot \mathbf{z}^* + \mathbf{x}^0 \quad \text{a} \quad \boldsymbol{\nu}^* = -(\mathbb{A} \cdot \mathbb{A}^T)^{-1} \cdot \mathbb{A} \cdot \nabla f(\mathbf{x}^*).$$

Je zrejmé, že optimálnym riešením pôvodnej úlohy je \mathbf{x}^* , avšak optimalita $\boldsymbol{\nu}^*$ na prvý pohľad zrejmá nie je. Na to potrebujeme podľa KKT podmienok ukázať

$$\nabla_{\mathbf{x}} L(\mathbf{x}^*, \boldsymbol{\nu}^*) = \mathbf{0},$$

čiže

$$\nabla f(\mathbf{x}^*) + \mathbb{A}^T \cdot (-(\mathbb{A} \cdot \mathbb{A}^T)^{-1} \cdot \mathbb{A} \cdot \nabla f(\mathbf{x}^*)) = \mathbf{0}.$$

To ukážeme tak, že vynásobíme vektor na ľavej strane rovnice regulárnu maticou. Ak bude súčin nulový vektor, musí byť vektor na ľavej strane nulový (kedže stĺpce regulárnej matice sú lineárne nezávislé). Kedže už vieme, že

$$\begin{pmatrix} \mathbb{F}^T \\ \mathbb{A} \end{pmatrix}$$

je regulárna matica, vynásobíme ľavú stranu rovnice práve touto maticou. Súčin rozdelíme na dve časti. Najprv vynásobíme ľavú stranu maticou \mathbb{F}^T a potom maticou \mathbb{A} . Dostávame

$$\begin{aligned}\mathbb{F}^T & \left[\nabla f(\mathbf{x}^*) + \mathbb{A}^T \cdot \left(-(\mathbb{A} \cdot \mathbb{A}^T)^{-1} \cdot \mathbb{A} \cdot \nabla f(\mathbf{x}^*) \right) \right] = \\ & = \mathbb{F}^T \cdot \nabla_{\mathbf{x}} f(\mathbf{x}^*) + (\mathbb{F}^T \cdot \mathbb{A}^T) \cdot \left(-(\mathbb{A} \cdot \mathbb{A}^T)^{-1} \cdot \mathbb{A} \cdot \nabla f(\mathbf{x}^*) \right) = \\ & = \nabla_{\mathbf{z}} \bar{f}(\mathbf{z}^*) + \mathbb{0} \cdot \left(-(\mathbb{A} \cdot \mathbb{A}^T)^{-1} \cdot \mathbb{A} \cdot \nabla f(\mathbf{x}^*) \right) = \mathbb{0}\end{aligned}$$

pretože $\nabla_{\mathbf{z}} \bar{f}(\mathbf{z}^*) = \mathbf{0}$. Okrem toho

$$\begin{aligned}\mathbb{A} & \left[\nabla f(\mathbf{x}^*) + \mathbb{A}^T \cdot \left(-(\mathbb{A} \cdot \mathbb{A}^T)^{-1} \cdot \mathbb{A} \cdot \nabla f(\mathbf{x}^*) \right) \right] = \\ & = \mathbb{A} \cdot \nabla f(\mathbf{x}^*) - (\mathbb{A} \cdot \mathbb{A}^T) \cdot (\mathbb{A} \cdot \mathbb{A}^T)^{-1} \cdot \mathbb{A} \cdot \nabla f(\mathbf{x}^*) = \\ & = \mathbb{A} \cdot (\nabla f(\mathbf{x}^*) - \nabla f(\mathbf{x}^*)) = \mathbb{0}.\end{aligned}$$

Znamená to, že ν^* je naozaj optimálne riešenie duálnej úlohy.

Cvičenia

CVIČENIE 6.1. Keď sme uvažovali konvexnú kvadratickú funkciu, tak sme odvodili, že optimum pri ohraničeniach tvaru rovníc získame, keď vyriešime sústavu lineárnych rovníc. Prečo sa takto jednoducho nedali riešiť úlohy lineárneho programovania? Veď tam bola účelová funkcia dokonca lineárna a nie kvadratická.

CVIČENIE 6.2. Ukážte, že keď v algoritme „Newtonova metóda pre rovnice z neprípustného bodu“ zvolíme vnejakej iterácii $t = 1$, tak všetky ďalšie body \mathbf{x} už budú prípustné riešenia.

CVIČENIE 6.3. Pomocou Newtonovej metódy vyriešte úlohu bez ohraničení

$$\min f(\mathbf{x}) = f(x_1, x_2) = e^{x_1+3x_2} + e^{x_1-3x_2} + e^{-x_1}.$$

1. Vykreslite účelovú funkciu pomocou Plot3D aj pomocou ContourPlot pre hodnoty $(x_1, x_2) \in \langle -2, 1 \rangle \times \langle -1, 1 \rangle$. Pre jednoduchšie naprogramovanie algoritmu je vhodné všetky funkcie definovať ako funkcie vektorových argumentov, teda napríklad účelovú funkciu $f(\mathbf{x}) = f(x_1, x_2)$ definovať ako $f[\{x_1, x_2\}]$. Potom je možné f používať ako $f[x]$, kde x je dvojzložkový vektor.

2. Definujte

- a) Gradient $\nabla f(\mathbf{x})$. Vykreslite $-\nabla f(\mathbf{x})$ pomocou VectorPlot.
- b) Hessovu maticu $\mathbb{H}_f(\mathbf{x}) = \nabla^2 f(\mathbf{x})$ a graficky overte, že je pozitívne semidefinitná (hlavné minory sú nezáporné), čiže $f(\mathbf{x})$ je konvexná funkcia.
- c) Newtonov krok $\Delta \mathbf{x}_{nt} = -\mathbb{H}_f^{-1}(\mathbf{x}) \cdot \nabla f(\mathbf{x})$, nakreslite ho pomocou funkcie VectorPlot a porovnajte s grafom v časti a).
- d) Newtonov decrement $\lambda^2(\mathbf{x}) = \nabla f(\mathbf{x})^T \cdot \mathbb{H}_f^{-1}(\mathbf{x}) \cdot \nabla f(\mathbf{x})$.

Kvôli zrýchleniu algoritmu a vykreslovania je vhodné tieto veličiny definovať tak, aby sa vypočítali symbolicky iba raz na začiatku (treba použiť „=“ a nie „:=“), aby sa pri vyčíslovaní už len dosadzovalo.

3. Naprogramujte priamu Newtonovu metódu ($t = 1$), pričom za štartovací bod zvoľte $\mathbf{x}^0 = (-0,9; 0,7)$.
4. Vizualizujte kroky algoritmu ako lomenú čiaru na grafe ContouPlot $f(\mathbf{x})$ spolu s VectorPlot $\Delta\mathbf{x}_{nt}$.
5. Naprogramujte pomocné algoritmy.
 - a) Exact line search. Vytvorte funkciu $Exact[x_-, \Delta x_-]$, ktorá vráti t minimalizujúce $f(\mathbf{x} + t \cdot \Delta\mathbf{x})$.
 - b) Backtracking line search. Vytvorte funkciu $Backtracking[x_-, \Delta x_-]$, ktorá vráti t podľa algoritmu Backtracking line search.
6. Naprogramujte tlmený Newtonov algoritmus ($t < 1$), čiže na výpočet použite algoritmy z časti 5.
7. Porovnajte fungovanie algoritmov (kroky algoritmu, počet iterácií, výpočtový čas):
 - a) Priamej Newtonovej metódy.
 - b) Tlmenej Newtonovej metódy s Exact line search.
 - c) Tlmenej Newtonovej metódy s Backtracking line search.

Vypisujte t v každej iterácii.

8. Pre gradientnú metódu (krok definovaný ako $\Delta\mathbf{x} = -\nabla f(\mathbf{x})$) urobte porovnanie ako v časti 7.
9. Porovnajte Newtonovu a gradientnu metódu.

CVIČENIE 6.4. Pomocou Newtonovej metódy pre úlohu s ohraničeniami tvaru rovníc vyriešte

$$\begin{array}{c} \min f(\mathbf{x}) = (x_1 + 1)^4 + 3(x_2 - 1)^2 + (x_3 - 1)^2 \\ \hline x_1 - x_2 + x_3 = 0,3 \end{array}$$

Otvorte si predpripravený súbor a doplňte v ňom chýbajúce časti. Pre jednoduchšie naprogramovanie algoritmu je vhodné definovať všetky funkcie ako funkcie vektorových argumentov. Teda napríklad účelovú funkciu $f(\mathbf{x}) = f(x_1, x_2, x_3)$ ako $f[\{x_{1-}, x_{2-}, x_{3-}\}]$, respektívne duálne reziduum $\mathbf{r}_d(\mathbf{y})$ ako $rd[\{x_{1-}, x_{2-}, x_{3-}, \nu_{-}\}]$.

1. Definície.

- a) Účelová funkcia $f(\mathbf{x})$.
- b) Gradient $\nabla f(\mathbf{x})$ a Hessova matica $\mathbb{H}_f(\mathbf{x}) = \nabla^2 f(\mathbf{x})$.
- c) Reziduá. Primárne $\mathbf{r}_p(\mathbf{x}, \nu) = \mathbb{A} \cdot \mathbf{x} - \mathbf{b}$, duálne $\mathbf{r}_d(\mathbf{x}, \nu) = \nabla f(\mathbf{x}) + \mathbb{A}^T \cdot \nu$ a celkové $\mathbf{r}(\mathbf{x}, \nu) = (\mathbf{r}_d, (\mathbf{x}, \nu), \mathbf{r}_p, (\mathbf{x}, \nu))$.
- d) Jacobiho matica

$$Dr(\mathbf{x}, \nu) = \begin{pmatrix} \nabla^2 f(\mathbf{x}) & \mathbb{A}^T \\ \mathbb{A} & 0 \end{pmatrix}.$$

Kvôli zrýchleniu algoritmu a vykresľovania je vhodné tieto veličiny definovať tak, aby sa vypočítali symbolicky iba raz na začiatku (treba použiť „=“ a nie „:=“), aby sa pri vyčíslovaní už len dosadzovalo.

2. Naprogramujte priamu Newtonovu metódu ($t = 1$), pričom za štartovací bod zvoľte $\mathbf{x}^0 = (1,5; -1 : 0)$ a $\nu^0 = 0$.

3. Naprogramujte pomocné algoritmy.
 - a) Exact line search. Vytvorte funkciu $Exact[x_\cdot, \Delta x_\cdot]$, ktorá vráti t minimalizujúce $\|\mathbf{r}(\mathbf{y} + t \cdot \Delta \mathbf{y})\|$.
 - b) Backtracking line search. Vytvorte funkciu $Backtracking[y_\cdot, \Delta y_\cdot]$, ktorá vráti t podľa algoritmu Backtracking line search.
4. Naprogramujte tlmený Newtonov algoritmus ($t < 1$), čiže na výpočet použite algoritmy z časti 3.
5. Porovnajte fungovanie algoritmov (kroky algoritmu vo vizualizácii, počet iterácií, výpočtový čas):
 - a) Priamej Newtonovej metódy.
 - b) Tlmenej Newtonovej metódy s Exact line search.
 - c) Tlmenej Newtonovej metódy s Backtracking line search.

Vypisujte t v každej iterácii.

7 Úlohy konvexného programovania

Budeme riešiť úlohu konvexného programovania

$$\begin{array}{c} \min f(\mathbf{x}) \\ \hline g_i(\mathbf{x}) \leq 0 & 1 \leq i \leq \ell \\ \mathbb{A} \cdot \mathbf{x} = \mathbf{b} \\ \hline \mathbf{x} \in D \end{array} \quad (18)$$

kde $D = D_f \cap D_{g_1} \cap \dots \cap D_{g_\ell}$, funkcie f, g_1, \dots, g_ℓ sú konvexné, dvakrát spojito diferencovateľné, \mathbb{A} je typu $(m \times n)$, pričom $m \leq n$, a existuje $\mathbf{x}^0 \in \mathcal{F}^o$, kde

$$\mathcal{F}^o = \{\mathbf{x} \in D; g_i(\mathbf{x}) < 0 \text{ pre } 1 \leq i \leq \ell \text{ a } \mathbb{A} \cdot \mathbf{x} = \mathbf{b}\}.$$

Znamená to, že úloha (18) splňa Slaterovu podmineku, takže pre ňu platí silná duálita a optimálne riešenie (v prípade, že nie je na hranici D) splňa KKT podmienky.

Nech je \mathbf{x}^* optimálne riešenie úlohy (18) a $(\boldsymbol{\lambda}^*, \boldsymbol{\nu}^*)$ nech je optimálne riešenie úlohy duálnej k (18). Potom podľa KKT podmienok (pozri (13) v závere Kapitoly 5) platí

$$\begin{aligned} \nabla f(\mathbf{x}^*) + \sum_{i=1}^{\ell} \lambda_i^* \cdot \nabla g_i(\mathbf{x}^*) + \mathbb{A}^T \cdot \boldsymbol{\nu}^* &= \mathbf{0} \\ g_i(\mathbf{x}^*) &\leq 0 \quad 1 \leq i \leq \ell \\ \boldsymbol{\lambda}^* &\geq \mathbf{0} \\ \lambda_i^* \cdot g_i(\mathbf{x}^*) &= 0 \quad 1 \leq i \leq \ell \\ \mathbb{A} \cdot \mathbf{x}^* &= \mathbf{b} \end{aligned}$$

Úlohu budeme riešiť pomocou bariérovej funkcie metódou vnútorného bodu.

Bariérová metóda

Nech je $\underline{J}(u)$ „funkcia“ definovaná

$$\underline{J}(u) = \begin{cases} 0 & \text{ak } u \leq 0, \\ \infty & \text{ak } u > 0. \end{cases}$$

Potom je úloha (18) ekvivalentná s úlohou

$$\begin{array}{c} \min f(\mathbf{x}) + \sum_{i=1}^{\ell} \underline{\mathbb{J}}(g_i(\mathbf{x})) \\ \hline \mathbb{A} \cdot \mathbf{x} = \mathbf{b} \\ \hline \mathbf{x} \in D \end{array}$$

Problém je v tom, že $\underline{\mathbb{J}}(u)$ nie je diferencovateľná (a vlastne to ani nie je funkcia). Preto ju aproximujeme. Dobrou aproximáciou je (porovnaj s bariérou v lineárnom programovaní, ktorú sme zaviedli v Kapitole 2, pozri Obrázok 2)

$$\tilde{\underline{\mathbb{J}}}(u) = -\left(\frac{1}{t}\right) \ln(-u),$$

pričom pre veľké t sa funkcia $\tilde{\underline{\mathbb{J}}}(u)$ približuje k $\underline{\mathbb{J}}(u)$. V ďalšom teda stačí, keď budeme riešiť úlohu

$$\begin{array}{c} \min f(\mathbf{x}) + \sum_{i=1}^{\ell} -\left(\frac{1}{t}\right) \ln(-g_i(\mathbf{x})) \\ \hline \mathbb{A} \cdot \mathbf{x} = \mathbf{b} \\ \hline \mathbf{x} \in D \end{array}$$

pretože pre $t \rightarrow \infty$ sa bude riešenie tejto úlohy blížiť k riešeniu úlohy (18), pozri Obrázok 3 v Kapitole 2.

Táto úloha už nemá ohraničenia tvaru nerovníc. Znamená to, že ak je účelová funkcia tejto úlohy konvexná, môžeme použiť metódy uvedené v predchádzajúcej kapitole.

VETA. Ak sú f, g_1, \dots, g_ℓ konvexné funkcie, tak je konvexná aj funkcia

$$f(\mathbf{x}) + \sum_{i=1}^{\ell} -\left(\frac{1}{t}\right) \ln(-g_i(\mathbf{x})).$$

DÔKAZ. Nech $1 \leq i \leq \ell$, $\mathbf{a}, \mathbf{b} \in D_{g_i}$ a $\theta \in \langle 0, 1 \rangle$. Keďže g_i je konvexná funkcia, tak platí

$$g_i(\theta \cdot \mathbf{a} + (1-\theta) \cdot \mathbf{b}) \leq \theta \cdot g_i(\mathbf{a}) + (1-\theta) \cdot g_i(\mathbf{b}). \quad (19)$$

Teraz označme $q(u) = -\left(\frac{1}{t}\right) \ln(-u)$. Keďže $\ln(u)$ je konkávna rastúca funkcia, $\ln(-u)$ je konkávna klesajúca a $-\ln(-u)$ je konvexná rastúca. Teda $q(u)$ je konvexná rastúca, pretože $t > 0$. Znamená to, že

$$q(g_i(\theta \cdot \mathbf{a} + (1-\theta) \cdot \mathbf{b})) \leq q(\theta \cdot g_i(\mathbf{a}) + (1-\theta) \cdot g_i(\mathbf{b})) \leq \theta \cdot q(g_i(\mathbf{a})) + (1-\theta) \cdot q(g_i(\mathbf{b})),$$

pričom v prvej nerovnosti sme využili vzťah (19) a rastúkosť q , v druhej sme využili konvexnosť q .

Keďže súčet konvexných funkcií je konvexná funkcia, tak je konvexná aj funkcia $f(\mathbf{x}) + \sum_{i=1}^{\ell} -\left(\frac{1}{t}\right) \ln(-g_i(\mathbf{x}))$. \square

Výraz

$$\Phi(\mathbf{x}) = - \sum_{i=1}^{\ell} \ln(-g_i(\mathbf{x}))$$

nazývame **logaritmická bariéra**. Neskôr budeme v Newtonovej metóde potrebovať $\nabla\Phi(\mathbf{x})$ a $\nabla^2\Phi(\mathbf{x})$. Kedže $\frac{\partial}{\partial x_j}\Phi(\mathbf{x}) = \sum_{i=1}^{\ell} \frac{1}{-g_i(\mathbf{x})} \cdot \frac{\partial}{\partial x_j} g_i(\mathbf{x})$, tak

$$\begin{aligned} \nabla\Phi(\mathbf{x}) &= \sum_{i=1}^{\ell} \frac{1}{-g_i(\mathbf{x})} \cdot \nabla g_i(\mathbf{x}) && \text{a} \\ \nabla^2\Phi(\mathbf{x}) &= \sum_{i=1}^{\ell} \frac{1}{g_i(\mathbf{x})^2} \cdot \nabla g_i(\mathbf{x}) \cdot \nabla g_i(\mathbf{x})^T + \sum_{i=1}^{\ell} \frac{1}{-g_i(\mathbf{x})} \cdot \nabla^2 g_i(\mathbf{x}) \end{aligned}$$

Centrálna cesta

Pre zjednodušenie výpočtu vynásobíme účelovú funkciu t . Potom stačí, keď budeme riešiť úlohu

$$\begin{array}{c} \min t \cdot f(\mathbf{x}) + \sum_{i=1}^{\ell} -\ln(-g_i(\mathbf{x})) \\ \hline \mathbb{A} \cdot \mathbf{x} = \mathbf{b} \\ \mathbf{x} \in D \end{array} \quad \text{čiže} \quad \begin{array}{c} \min t \cdot f(\mathbf{x}) + \Phi(\mathbf{x}) \\ \hline \mathbb{A} \cdot \mathbf{x} = \mathbf{b} \\ \mathbf{x} \in D \end{array} \quad (20)$$

pre $t \rightarrow \infty$.

DEFINÍCIA. Ak má úloha (20) pre každé t jediné optimálne riešenie $\mathbf{x}^*(t)$, tak tie-to riešenia tvoria krivku parametrizovanú parametrom $t \in (0, \infty)$, ktorá sa nazýva **centrálna cesta (central path)**.

Všimnime si, že keby bola úloha (18) úlohou lineárneho programovania, pričom jedinými ohraničeniami typu nerovníc by boli ohraničenia na nezápornosť premenných, tak by sme dostali centrálnu cestu definovanú v úvode Kapitoly 2.

Kedže (20) má iba ohraničenia tvaru rovníc, Lagrangeova funkcia je

$$L(\mathbf{x}, \boldsymbol{\nu}) = t \cdot f(\mathbf{x}) + \Phi(\mathbf{x}) + \boldsymbol{\nu}^T \cdot (\mathbb{A} \cdot \mathbf{x} - \mathbf{b})$$

a z KKT podmienok dostávame, že okrem $\mathbb{A} \cdot \mathbf{x}^*(t) = \mathbf{b}$ a $g_i(\mathbf{x}^*(t)) < 0$, kde $1 \leq i \leq \ell$, musí platiť $\nabla L(\mathbf{x}^*(t), \tilde{\boldsymbol{\nu}}) = \mathbf{0}$, čiže

$$t \cdot \nabla f(\mathbf{x}^*(t)) + \nabla \Phi(\mathbf{x}^*(t)) + \mathbb{A}^T \cdot \tilde{\boldsymbol{\nu}} = \mathbf{0},$$

čo dá po rozpisani

$$t \cdot \nabla f(\mathbf{x}^*(t)) + \sum_{i=1}^{\ell} \frac{1}{-g_i(\mathbf{x}^*(t))} \cdot \nabla g_i(\mathbf{x}^*(t)) + \mathbb{A}^T \cdot \tilde{\boldsymbol{\nu}} = \mathbf{0} \quad (21)$$

pre nejaké $\tilde{\boldsymbol{\nu}}$.

LEMA. Každý bod centrálnej cesty $\mathbf{x}^*(t)$ určuje, spolu so zodpovedajúcim vektorom $\tilde{\boldsymbol{\nu}}$, duálne prípustný bod $(\boldsymbol{\lambda}^*(t), \boldsymbol{\nu}^*(t))$ pôvodnej úlohy (18). Pre tento duálne prípustný bod platí $\lambda_i^*(t) = -\frac{1}{t \cdot g_i(\mathbf{x}^*(t))}$, kde $1 \leq i \leq \ell$, a $\boldsymbol{\nu}^*(t) = \tilde{\boldsymbol{\nu}}/t$.

DÔKAZ. Isto $\boldsymbol{\lambda}^*(t) > \mathbf{0}$. Lagrangeova funkcia pre pôvodnú úlohu (18) je

$$L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) = f(\mathbf{x}) + \sum_{i=1}^{\ell} \lambda_i \cdot g_i(\mathbf{x}) + \boldsymbol{\nu}^T \cdot (\mathbb{A} \cdot \mathbf{x} - \mathbf{b})$$

a duálna účelová funkcia je

$$\psi(\boldsymbol{\lambda}, \boldsymbol{\nu}) = \min_{\mathbf{x}} \left\{ f(\mathbf{x}) + \sum_{i=1}^{\ell} \lambda_i \cdot g_i(\mathbf{x}) + \boldsymbol{\nu}^T \cdot (\mathbb{A} \cdot \mathbf{x} - \mathbf{b}) \right\}.$$

Pre $\lambda_i^*(t) = -\frac{1}{t \cdot g_i(\mathbf{x}^*(t))}$, $1 \leq i \leq \ell$, a $\boldsymbol{\nu}^*(t) = \tilde{\boldsymbol{\nu}}/t$ dostávame

$$\psi(\boldsymbol{\lambda}^*(t), \boldsymbol{\nu}^*(t)) = \min_{\mathbf{x}} \left\{ f(\mathbf{x}) + \sum_{i=1}^{\ell} \frac{1}{-t \cdot g_i(\mathbf{x}^*(t))} \cdot g_i(\mathbf{x}) + \frac{1}{t} \cdot \tilde{\boldsymbol{\nu}}^T \cdot (\mathbb{A} \cdot \mathbf{x} - \mathbf{b}) \right\}.$$

Výraz vo vnútri zložených zátvoriek je pre dané $\boldsymbol{\lambda}^*(t)$ a $\boldsymbol{\nu}^*(t)$ funkcia v premennej \mathbf{x} . Táto funkcia je konvexná, pretože sú konvexné f, g_1, \dots, g_ℓ a $\boldsymbol{\lambda}^*(t) > \mathbf{0}$. Znamená to, že ak má táto funkcia kritický bod, tak tento bod je hľadané minimum. Po zderivovaní dostaneme rovnicu

$$\nabla f(\mathbf{x}) + \sum_{i=1}^{\ell} \frac{1}{-t \cdot g_i(\mathbf{x}^*(t))} \cdot \nabla g_i(\mathbf{x}) + \frac{1}{t} \cdot \mathbb{A}^T \cdot \tilde{\boldsymbol{\nu}} = \mathbf{0}.$$

Ked' ju porovnáme s (21) vidíme, že gradient je nulový ak $\mathbf{x} = \mathbf{x}^*(t)$. Preto je dané $(\boldsymbol{\lambda}^*(t), \boldsymbol{\nu}^*(t))$ duálne prípustný bod (18) zodpovedajúci $\mathbf{x}^*(t)$. \square

Podľa lemy vieme, že

$$\begin{aligned} \psi(\boldsymbol{\lambda}^*(t), \boldsymbol{\nu}^*(t)) &= L(\mathbf{x}^*(t), \boldsymbol{\lambda}^*(t), \boldsymbol{\nu}^*(t)) = \\ &= f(\mathbf{x}^*(t)) + \sum_{i=1}^{\ell} \lambda_i^*(t) \cdot g_i(\mathbf{x}^*(t)) + \boldsymbol{\nu}^*(t)^T \cdot (\mathbb{A} \cdot \mathbf{x}^*(t) - \mathbf{b}) = \\ &= f(\mathbf{x}^*(t)) + \sum_{i=1}^{\ell} -\frac{1}{t} + 0 = f(\mathbf{x}^*(t)) - \frac{\ell}{t}. \end{aligned}$$

Ked'že podľa slabej duality platí $\min f(\mathbf{x}) \geq \max \psi(\boldsymbol{\lambda}, \boldsymbol{\nu})$, dostávame dolnú hranicu na optimálne riešenie primárnej úlohy (18) a zároveň dobrú podmienku na ukončenie algoritmu. Ak $\frac{\ell}{t} < \varepsilon$, kde ε je presnosť, tak môžeme skončiť.

Z predchádzajúceho a z (21) dostávame KKT podmienky pre centrálnu cestu. Pre $\lambda_i = -\frac{1}{t \cdot g_i(\mathbf{x})}$ musí platiť

$$\begin{aligned} \nabla f(\mathbf{x}) + \sum_{i=1}^{\ell} \lambda_i \cdot \nabla g_i(\mathbf{x}) + \mathbb{A}^T \cdot \boldsymbol{\nu} &= \mathbf{0} \\ g_i(\mathbf{x}) < 0 &\quad 1 \leq i \leq \ell \\ \boldsymbol{\lambda} > \mathbf{0} & \\ \lambda_i \cdot g_i(\mathbf{x}) &= -\frac{1}{t} \quad 1 \leq i \leq \ell \\ \mathbb{A} \cdot \mathbf{x} &= \mathbf{b} \end{aligned} \tag{22}$$

Ked' tieto podmienky porovnáme s KKT podmienkami pre pôvodnú úlohu (18), ktoré sme uviedli na začiatku kapitoly, tak rozdiel je v podstate iba v tom, že tu máme $\lambda_i \cdot g_i(\mathbf{x}) = -\frac{1}{t}$ a nie $\lambda_i \cdot g_i(\mathbf{x}) = 0$, kde $1 \leq i \leq \ell$.

Priama metóda riešenia

Podľa predchádzajúceho, ak je ε požadovaná presnosť, tak stačí zvoliť $t = \ell/\varepsilon$ a vyriešiť úlohu

$$\begin{array}{c} \min \frac{\ell}{\varepsilon} \cdot f(\mathbf{x}) + \Phi(\mathbf{x}) \\ \hline \mathbb{A} \cdot \mathbf{x} = \mathbf{b} \\ \hline \mathbf{x} \in D \end{array}$$

pomocou Newtonovej metódy pre ohraničenia tvaru rovníc z predchádzajúcej kapitoly. Pre malé úlohy, vhodne zvolený počiatočný bod a presnosť (ε nesmie byť príliš malé) tento postup funguje dobre. Avšak v ostatných prípadoch nefunguje, pretože druhá derivácia bariérovej funkcie sa v blízkosti hranice mení príliš rýchlo. Preto sa priama Newtonova metóda nepoužíva.

Path-following metóda

Používa sa nasledujúci algoritmus, ktorý rieši postupnosť úloh, pričom zakaždým je koncový (výsledný) bod jednej úlohy začiatočným bodom ďalšej.

ALGORITMUS: Path-following bariérová metóda.

Vstup: $\mathbf{x}^0 \in D_{t \cdot f + \Phi}$, $t > 0$, $\mu > 1$, presnosť $\varepsilon > 0$.

Begin

Repeat Begin

CENTERING STEP

Začni v \mathbf{x}^0 a vypočítaj $\mathbf{x}^*(t)$ ako riešenie úlohy

$$\begin{array}{c} \min t \cdot f(\mathbf{x}) + \Phi(\mathbf{x}) \\ \hline \mathbb{A} \cdot \mathbf{x} = \mathbf{b} \\ \hline \mathbf{x} \in D \end{array}$$

algoritmom „Newtonova metóda pre rovnice z neprípustného bodu“ z Kapitoly 6;

$\mathbf{x}^* = \mathbf{x}^*(t)$;

Quit If $\ell/t < \varepsilon$;

$t = \mu \cdot t$; $\mathbf{x}^0 = \mathbf{x}^*$;

End;

End.

Ako vyzerá Newtonov krok v tomto algoritme? Podľa predchádzajúcej kapitoly je riešením sústavy

$$\begin{pmatrix} t \cdot \nabla^2 f(\mathbf{x}) + \nabla^2 \Phi(\mathbf{x}) & \mathbb{A}^T \\ \mathbb{A} & \mathbb{0} \end{pmatrix} \cdot \begin{pmatrix} \Delta \mathbf{x}_{nt} \\ \Delta \boldsymbol{\nu}_{nt} \end{pmatrix} = - \begin{pmatrix} t \cdot \nabla f(\mathbf{x}) + \nabla \Phi(\mathbf{x}) + \mathbb{A}^T \boldsymbol{\nu} \\ \mathbb{A} \cdot \mathbf{x} - \mathbf{b} \end{pmatrix}.$$

V tomto algoritme sa v kroku Centering step robí algoritmus z predchádzajúcej kapitoly, teda v tomto jednom kroku budeme mať veľa iterácií. V princípe nemusíme spočítať $\mathbf{x}^*(t)$ s presnosťou ε , avšak tým veľa neušetríme, pretože medzi „dostatočne dobrú aproximáciu $\mathbf{x}^*(t)$ “ a „presnou hodnotou $\mathbf{x}^*(t)$ “ je iba zopár krokov Newtonovho algoritmu.

Pre príliš malé μ budeme mať priveľa iterácií algoritmu, na druhej strane pre príliš veľké μ budeme mať priveľa iterácií v kroku Centering step. Vhodnou voľbou je $\mu \in \langle 10, 20 \rangle$.

Vhodnou voľbou na počiatočnú hodnotu t je riešenie úlohy

$$\min_{t, \boldsymbol{\nu}} \|t \cdot \nabla f(\mathbf{x}^0) + \nabla \Phi(\mathbf{x}^0) + \mathbb{A}^T \cdot \boldsymbol{\nu}\|.$$

Ak poznáme duálne prípustný bod $(\boldsymbol{\lambda}, \boldsymbol{\nu})$, tak vhodná počiatočná hodnota je

$$t = \frac{\ell}{f(\mathbf{x}^0) - \psi(\boldsymbol{\lambda}, \boldsymbol{\nu})}.$$

Prvá fáza, nájdenie $\mathbf{x} \in \mathcal{F}^o$

Zostáva opísť, ako nájdeme úvodné riešenie $\mathbf{x} \in \mathcal{F}^o$, v ktorom spustíme algoritmus. (V skutočnosti by nám stačilo nájsť $\mathbf{x} \in D_{t \cdot f + \Phi}$, čiže pre \mathbf{x} nemusí platiť $\mathbb{A} \cdot \mathbf{x} = \mathbf{b}$. Budeme tu však požadovať aj splnenie podmienky $\mathbb{A} \cdot \mathbf{x} = \mathbf{b}$, pretože chceme zistiť, či má zadaná úloha striktne prípustné riešenie. V prípade, keď úloha striktne prípustné riešenie nemá, by sme chceli zistiť, ktoré ohraničenia spôsobujú problémy.) V tejto časti navrhнемe novú úlohu, ktorej riešením bude hľadaný bod $\mathbf{x} \in \mathcal{F}^o$. Na riešenie novej úlohy použijeme algoritmus opísaný v tejto kapitole. Ukážeme si štyri možnosti ako nájsť $\mathbf{x} \in \mathcal{F}^o$, pričom zakaždým predpokladáme, že máme $\mathbf{x}^0 \in D = D_f \cap D_{g_1} \cap \dots \cap D_{g_\ell}$.

1. možnosť

Budeme riešiť úlohu

$$\begin{array}{c} \min s \\ \hline g_i(\mathbf{x}) \leq s & 1 \leq i \leq \ell \\ \mathbb{A} \cdot \mathbf{x} = \mathbf{b} \\ \hline \mathbf{x} \in D \end{array}$$

pričom na začiatku volíme s také, aby platilo $g_i(\mathbf{x}^0) < s$ pre všetky i , $1 \leq i \leq \ell$. Všimnime si, že ako sa s zmenšuje, bod \mathbf{x} sa blíži k množine prípustných riešení. Nech je s^* riešením tejto úlohy.

- a) Ak $s^* < 0$ tak úloha má striktne prípustné riešenie a algoritmus ho nájde.

- b) Ak $s^* > 0$ tak úloha nemá prípustné riešenie a nie je čo optimalizovať.
- c) Ak $s^* = 0$ a minimum sa dosahuje v \mathbf{x}^* , tak úloha má prípustné, ale nie striktne prípustné riešenie. Ak sa minimum nedosahuje, tak úloha prípustné riešenie nemá.

2. možnosť

Budeme riešiť úlohu

$$\begin{array}{c} \min s_1 + \cdots + s_\ell \\ \hline g_i(\mathbf{x}) \leq s_i \quad 1 \leq i \leq \ell \\ \mathbb{A} \cdot \mathbf{x} = \mathbf{b} \\ \hline s_1, \dots, s_\ell \geq 0 \\ \hline \mathbf{x} \in D \end{array}$$

Túto variáciu predchádzajúceho postupu môžeme použiť, ak zistíme, že úloha nemá prípustné riešenie. V takom prípade totiž obyčajne chceme získať riešenie, ktoré je čo najblížšie k prípustnému. Keď úloha prípustné riešenie nemá, touto úlohou vieme tiež identifikovať, ktoré ohraničenia sú problematické. Sú to tie, pre ktoré v optimálnom riešení platí $s_i^* > 0$.

3. možnosť

Budeme riešiť úlohu

$$\begin{array}{c} \min s \\ \hline f(\mathbf{x}) \leq M \\ g_i(\mathbf{x}) \leq s \quad 1 \leq i \leq \ell \\ \mathbb{A} \cdot \mathbf{x} = \mathbf{b} \\ \hline \mathbf{x} \in D \end{array}$$

kde M je väčšie ako $f(\mathbf{x}^0)$ aj $f(\mathbf{x}^*)$.

Táto úloha je malou variáciou úlohy v 1. možnosti, avšak umožňuje získať bod blízko centrálnej cesty, ktorý je vynikajúcou voľbou pre počiatočný bod nášho algoritmu. Bariérovú funkciu zostrojíme z ohraničení tvaru nerovníc, čiže

$$\Phi(\mathbf{x}) = -\ln(M - f(\mathbf{x})) - \sum_{i=1}^{\ell} \ln(s - g_i(\mathbf{x}))$$

Lagrangeova funkcia pre centrálnu cestu, tak ako sme ju opísali vyššie, je

$$L(s, \mathbf{x}, \boldsymbol{\nu}) = s \cdot \bar{t} - \ln(M - f(\mathbf{x})) - \sum_{i=1}^{\ell} \ln(s - g_i(\mathbf{x})) + \boldsymbol{\nu}^T \cdot (\mathbb{A} \cdot \mathbf{x} - \mathbf{b})$$

a $\nabla_{s,\mathbf{x}} L(s, \mathbf{x}, \boldsymbol{\nu}) = \mathbf{0}$ dá

$$\begin{aligned} \bar{t} &= \sum_{i=1}^{\ell} \frac{1}{s - g_i(\mathbf{x})} && \text{a} \\ \frac{1}{M - f(\mathbf{x})} \cdot \nabla f(\mathbf{x}) + \sum_{i=1}^{\ell} \frac{1}{s - g_i(\mathbf{x})} \cdot \nabla g_i(\mathbf{x}) + \mathbb{A}^T \cdot \boldsymbol{\nu} &= \mathbf{0}. \end{aligned}$$

Teraz keď algoritmus skončí pri $s = 0$, tak pre $t = \frac{1}{M - f(\mathbf{x})}$ dostávame

$$t \cdot \nabla f(\mathbf{x}) + \sum_{i=1}^{\ell} \frac{1}{-g_i(\mathbf{x})} \cdot \nabla g_i(\mathbf{x}) + \mathbb{A}^T \cdot \boldsymbol{\nu} = \mathbf{0},$$

čo je podľa (21) bod na centrálnej ceste pre úlohu (18).

4. možnosť

Newtonovou metódou začínajúcou v neprípustnom bode môžeme riešiť aj úlohu

$$\begin{array}{c} \min f(\mathbf{x}) \\ \hline g_i(\mathbf{x}) \leq s & 1 \leq i \leq \ell \\ \mathbb{A} \cdot \mathbf{x} = \mathbf{b} \\ \hline s = 0 \\ \hline \mathbf{x} \in D \end{array}$$

To znamená, že pri štarte v $\mathbf{x}^0 \in D$ a $s^0 > \max\{g_i(\mathbf{x}^0); 1 \leq i \leq \ell\}$ budeme riešiť úlohu

$$\begin{array}{c} \min t \cdot f(\mathbf{x}) - \sum_{i=1}^{\ell} \ln(s - g_i(\mathbf{x})) \\ \hline \mathbb{A} \cdot \mathbf{x} = \mathbf{b} \\ \hline s = 0 \\ \hline \mathbf{x} \in D \end{array}$$

Podobný trik môžeme použiť, ak nemáme $\mathbf{x}^0 \in D$. Vtedy riešime úlohu

$$\begin{array}{c} \min t \cdot f(\mathbf{x} - \mathbf{z}_0) - \sum_{i=1}^{\ell} \ln(s - g_i(\mathbf{x} - \mathbf{z}_i)) \\ \hline \mathbb{A} \cdot \mathbf{x} = \mathbf{b} \\ \hline s = 0 \\ \hline \mathbf{z}_i = \mathbf{0} & 0 \leq i \leq \ell \\ \hline \mathbf{x} \in \mathbb{R}^n \end{array}$$

pričom začneme s takým $\mathbf{z}_0, \mathbf{z}_1, \dots, \mathbf{z}_{\ell}$, pre ktoré je $\mathbf{x} + \mathbf{z}_0 \in D_f$, $\mathbf{x} + \mathbf{z}_1 \in D_{g_1}$, \dots , $\mathbf{x} + \mathbf{z}_{\ell} \in D_{g_{\ell}}$. Problém tohto prístupu je v tom, že ak úloha nemá prípustné riešenie, tak nemáme dobré kritérium na ukončenie algoritmu.

Cvičenia

CVIČENIE 7.1. Pomocou Bariérovej metódy vyriešte úlohu

$$\begin{aligned} \min f(\mathbf{x}) &= (x_1 + 1)^4 + 3(x_2 - 1)^2 + (x_3 - 1)^2 \\ &\quad (x_1 - 1)^2 + (x_2 + 2,5)^2 \leq 9 \\ &\quad (x_1 - 1)^2 + (x_2 - 2)^2 \leq 9 \\ &\quad x_1 \leq 2 - x_2^2 \\ &\quad x_1 - x_2 + x_3 = 0,3 \end{aligned}$$

Otvorte si (predpripravený) súbor, upravte ho a doplňte chýbajúce časti. Najprv si do zošita nakreslite, akú časť priestoru ohraničujú nerovnice (v dvojrozmernom reze rovinou $\overleftrightarrow{x_1 x_2}$).

1. Definície.

- a) Funkcie $g_1(\mathbf{x})$, $g_2(\mathbf{x})$ a $g_3(\mathbf{x})$ pre ohraničenia tvaru nerovníc.
 - b) Bariérová funkcia $\Phi(\mathbf{x}) = -\sum_{i=1}^3 \ln(-g_i(\mathbf{x}))$.
 - c) Modifikovaná účelová funkcia $F(\mathbf{x}, t_B) = f(\mathbf{x}) + \frac{1}{t_B} \Phi(\mathbf{x})$.
2. Upravte definície gradientu, Hessovej matice, reziduú a Jacobiho matice. Veličiny je vhodné definovať ako funkcie parametra t_B . Teda napríklad r je vhodné definovať ako $r[\{x_1, x_2, x_3, \nu\}, t_B]$.
3. Vykreslite bariérovú funkciu $\Phi(\mathbf{x})$ ako funkciu dvoch premenných x_1 a x_2 (keďže na x_3 nezávisí).
4. Vykreslite prípustnú množinu \mathcal{F} s hodnotami účelovej funkcie $f(\mathbf{x})$.
5. Modifikujte Backtracking algoritmus. Do zastavovacieho kritéria treba pridať podmineky na to, aby nerovnice zostali splnené.
6. Naprogramujte Bariérovú metódu (s Backtrackingom v modifikovanej podobe), pričom štartovací bod zvolíte tak, aby splňal nerovnice a nesplňal rovnicu.
7. Vizualizujte zvlášť (inou farbou) vonkajšie a vnútorné iterácie.
8. Naprogramujte prvú fázu Bariérovej metódy, ktorá môže štartovať z bodu \mathbf{x}^0 , pre ktorý $\mathbf{x}^0 \notin D_{f+\frac{1}{t_B}\phi}$, a nájdite prípustný bod.

Otázky.

- Ako vyzerá prípustná množina?
- Kde bude optimum?
- Aký systém rovníc sa bude riešiť?
- Čo sa bude ukladať a kedy?
- Ako treba nastaviť μ a $t_{B,0}$, aby kroky algoritmu čo najviac kopírovali centrálnu cestu?

CVIČENIE 7.2. Doplnenie Bariérovej metódy z Cvičenia 7.1.

1. Nastavte štartovací bod na $\mathbf{x}^0 = (1,5 ; 0 ; 0)$.
2. Vypisujte t v každom kroku.
3. Experimentujte s parametrami μ a $t_{B,0}$ a meňte ich hodnoty.
 - a) Nastavte $\mu = 2$ a meňte $t_{B,0} = 0,001; 0,1; 1; 10$.
 - b) Nastavte $t_{B,0} = 0,1$ a meňte $\mu = 1; 2; 5; 10$.

Ako a prečo sa mení správanie algoritmu? Argumentujte bariérovou funkciou.

4. Ako treba nastaviť μ a $t_{B,0}$, aby kroky algoritmu čo najviac kopírovali centrálnu cestu?

8 Primárno-duálne metódy

Primárno-duálne metódy sú často efektívnejšie ako Bariérové metódy a obyčajne fungujú, aj keď má problém iba prípustné a nie striktne prípustné riešenie. My však opíšeme algoritmus, ktorý využíva striktne prípustné riešenie. Predpokladáme teda, že $\mathcal{F}^o \neq \emptyset$.

Lineárna konvergencia

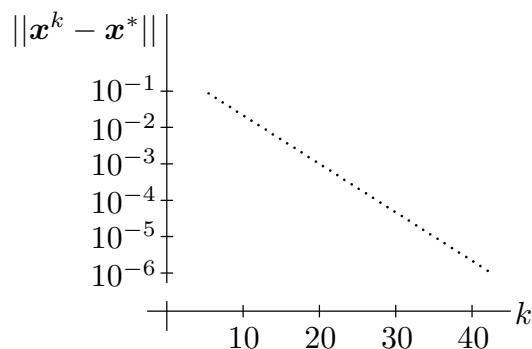
Primárno-duálne metódy často konvergujú rýchlejšie ako lineárne. Čo však znamená, že algoritmus konverguje lineárne?

DEFINÍCIA. Nech je $\{\mathbf{x}^k\}_{k=0}^{\infty}$ postupnosť bodov, získaných v jednotlivých iteráciách algoritmu, ktorá konverguje k optimálnemu riešeniu \mathbf{x}^* . Ak existuje ξ také, že $0 < \xi < 1$ a

$$\lim_{k \rightarrow \infty} \frac{\|\mathbf{x}^{k+1} - \mathbf{x}^*\|}{\|\mathbf{x}^k - \mathbf{x}^*\|} = \xi,$$

tak hovoríme, že $\{\mathbf{x}^k\}_{k=0}^{\infty}$ **konverguje lineárne** k \mathbf{x}^* .

Podľa definície, ak $\{\mathbf{x}^k\}_{k=0}^{\infty}$ konverguje lineárne, tak zlepšenie o rád trvá rovnako dlho, nezávisle od toho, ktorý rád to je. Lineárnu sa táto konvergencia nazýva preto, lebo keď zakreslíme hodnoty $\|\mathbf{x}^k - \mathbf{x}^*\|$ do obrázku s logaritmickou škálou na vertikálnej osi, zatiaľ čo na horizontálnej osi budeme mať číslo iterácie k , tak hodnoty $\|\mathbf{x}^k - \mathbf{x}^*\|$ budú pre veľké k ležať približne na priamke, pozri Obrázok 23.



Obrázok 23

Primárno-duálna metóda vnútorného bodu

Budeme riešiť úlohu konvexného programovania (18)

$$\begin{array}{c} \min f(\mathbf{x}) \\ \hline g_i(\mathbf{x}) \leq 0 & 1 \leq i \leq \ell \\ \mathbb{A} \cdot \mathbf{x} = \mathbf{b} \\ \hline \mathbf{x} \in D \end{array}$$

kde f, g_1, \dots, g_ℓ sú konvexné, dvakrát spojito diferencovateľné funkcie, pozri Kapitolu 7. Označíme

$$G(\mathbf{x}) = \begin{pmatrix} g_1(\mathbf{x}) \\ \vdots \\ g_\ell(\mathbf{x}) \end{pmatrix}, \quad DG(\mathbf{x}) = \begin{pmatrix} \nabla g_1(\mathbf{x})^T \\ \vdots \\ \nabla g_\ell(\mathbf{x})^T \end{pmatrix}, \quad \text{diag}(\mathbf{q}) = \begin{pmatrix} q_1 & & 0 \\ & \ddots & \\ 0 & & q_n \end{pmatrix},$$

kde \mathbf{q} je vektor dĺžky n . Podľa (22) z predchádzajúcej kapitoly, KKT podmienky pre centrálnu cestu sú

$$\begin{pmatrix} \nabla f(\mathbf{x}) + DG(\mathbf{x})^T \cdot \boldsymbol{\lambda} + \mathbb{A}^T \cdot \boldsymbol{\nu} \\ -\text{diag}(\boldsymbol{\lambda}) \cdot G(\mathbf{x}) - \frac{1}{t} \cdot \mathbf{1} \\ \mathbb{A} \cdot \mathbf{x} - \mathbf{b} \end{pmatrix} = \mathbf{0}, \quad (23)$$

kde $\mathbf{1} = (1, \dots, 1)^T$. Znamená to, že ak je $(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu})$ riešenie tejto rovnice, potom pre $\mathbf{x}^*(t) = \mathbf{x}$, $\boldsymbol{\lambda}^*(t) = \boldsymbol{\lambda}$ a $\boldsymbol{\nu}^*(t) = \boldsymbol{\nu}$ platí, že $\mathbf{x}^*(t)$ je bod na centrálnej ceste. Naviac, $\mathbf{x}^*(t)$ je prípustné riešenie primárnej úlohy (18) a $(\boldsymbol{\lambda}^*(t), \boldsymbol{\nu}^*(t))$ je zodpovedajúce riešenie úlohy duálnej k (18), pričom rozdiel medzi hodnotami účelových funkcií týchto riešení je ℓ/t .

Budeme hľadať riešenie (23) pomocou Newtonovej metódy. Označme

$$\begin{aligned} \mathbf{r}_D &= \nabla f(\mathbf{x}) + DG(\mathbf{x})^T \cdot \boldsymbol{\lambda} + \mathbb{A}^T \cdot \boldsymbol{\nu} && \text{duálne reziduum,} \\ \mathbf{r}_C &= -\text{diag}(\boldsymbol{\lambda}) \cdot G(\mathbf{x}) - \frac{1}{t} \cdot \mathbf{1} && \text{centralitné reziduum,} \\ \mathbf{r}_P &= \mathbb{A} \cdot \mathbf{x} - \mathbf{b} && \text{primárne reziduum,} \\ \mathbf{r}_T &= (\mathbf{r}_D, \mathbf{r}_C, \mathbf{r}_P)^T && \text{totálne reziduum.} \end{aligned}$$

Nech $\mathbf{y} = (\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu})^T$. Našou úlohou je vyriešiť rovnicu $\mathbf{r}_T(\mathbf{y}) = \mathbf{0}$. Opäť použijeme Newtonovu metódu. Ak označíme $F(\mathbf{y}) = \mathbf{r}_T(\mathbf{y})$, tak Newtonov algoritmus navrhuje posunúť sa z bodu \mathbf{y} v smere Newtonovho kroku $\Delta \mathbf{y}_{nt}$, pričom $\Delta \mathbf{y}_{nt}$ je riešením sústavy lineárnych rovníc (3) z Kapitoly 2

$$DF(\mathbf{y}) \cdot \Delta \mathbf{y}_{nt} = -F(\mathbf{y}),$$

kde $DF(\mathbf{y})$ je Jacobiho matica funkcie F v bode \mathbf{y} . Po rozpísaní z (23) dostávame, že Newtonov krok, ktorý teraz označíme $\Delta \mathbf{y}_{pd} = (\Delta \mathbf{x}_{pd}, \Delta \boldsymbol{\lambda}_{pd}, \Delta \boldsymbol{\nu}_{pd})^T$, je riešením sústavy lineárnych rovníc

$$\begin{pmatrix} \nabla^2 f(\mathbf{x}) + \sum_{i=1}^{\ell} \lambda_i \cdot \nabla^2 g_i(\mathbf{x}) & DG(\mathbf{x})^T & \mathbb{A}^T \\ -\text{diag}(\boldsymbol{\lambda}) \cdot DG(\mathbf{x}) & -\text{diag}(G(\mathbf{x})) & \mathbf{0} \\ \mathbb{A} & \mathbf{0} & \mathbf{0} \end{pmatrix} \cdot \begin{pmatrix} \Delta \mathbf{x}_{pd} \\ \Delta \boldsymbol{\lambda}_{pd} \\ \Delta \boldsymbol{\nu}_{pd} \end{pmatrix} = - \begin{pmatrix} \mathbf{r}_D \\ \mathbf{r}_C \\ \mathbf{r}_P \end{pmatrix}. \quad (24)$$

Budeme používať backtracking, preto iterácie \mathbf{x}^k a $(\boldsymbol{\lambda}^k, \boldsymbol{\nu}^k)$ nemusia byť prípustnými riešeniami P, respektíve D. Predpokladáme iba, že platí $G(\mathbf{x}^k) < \mathbf{0}$ a $\boldsymbol{\lambda}^k > \mathbf{0}$. Hodnotu $\hat{\eta} = -G(\mathbf{x}^k)^T \cdot \boldsymbol{\lambda}^k = \sum_{i=1}^{\ell} g_i(\mathbf{x}^k) \cdot \lambda_i^k$, ktorá vyjadruje do akej miery sú narušené podmienky komplementarity, nazývame **surrogate gap** (náhradný rozdiel). Ak budú \mathbf{x}^k a $(\boldsymbol{\lambda}^k, \boldsymbol{\nu}^k)$ prípustné a duálne prípustné, čiže ak $\mathbf{r}_P = \mathbf{0}$ a $\mathbf{r}_D = \mathbf{0}$, tak zo surrogate gap získame **duality gap**. Ak $\mathbf{r}_C = 0$, tak podľa definície centralitného rezidua z predchádzajúcej strany platí $t = \ell/\hat{\eta}$, čo budeme využívať v algoritme.

ALGORITMUS: Primárno-duálna metóda vnútorného bodu.

Vstup: \mathbf{x}^0 splňajúce $G(\mathbf{x}^0) < \mathbf{0}$, $\boldsymbol{\lambda} > \mathbf{0}$, $\mu > 1$ a presnosti $\varepsilon_{pd} > 0$ a $\varepsilon > 0$.

Begin

Repeat

$$\hat{\eta} = -G(\mathbf{x})^T \cdot \boldsymbol{\lambda}; \quad t = \mu \cdot \ell/\hat{\eta};$$

Urči $\Delta \mathbf{y}_{pd}$ podľa vzťahov (24);

Backtrackingom urči násobok s ;

$$\mathbf{y} = \mathbf{y} + s \cdot \Delta \mathbf{y}_{pd};$$

Until $\left(\|\mathbf{r}_P\| \leq \varepsilon_{pd} \text{ And } \|\mathbf{r}_D\| \leq \varepsilon_{pd} \text{ And } \hat{\eta} \leq \varepsilon \right)$;

End.

Poznamenajme, že t určené na začiatku slučky vystupuje v \mathbf{r}_C a μ tu hrá rolu centralizačného parametra.

Vhodná voľba pre μ je $\mu \doteq 10$. Hodnoty ε_{pd} a ε volíme maličké, pretože primárno-duálne metódy vnútorného bodu obyčajne konvergujú rýchlejšie ako lineárne.

Hodnotu s určíme backtrackingom, najprv však potrebujeme zabezpečiť, aby pre nové \mathbf{x} a $\boldsymbol{\lambda}$ platilo $G(\mathbf{x}) < \mathbf{0}$ a $\boldsymbol{\lambda} > \mathbf{0}$, čiže aby sme zostali vnútri ohraničení tvaru nerovností. Označme

$$\mathbf{x}^n = \mathbf{x} + s \cdot \Delta \mathbf{x}_{pd}, \quad \boldsymbol{\lambda}^n = \boldsymbol{\lambda} + s \cdot \Delta \boldsymbol{\lambda}_{pd}, \quad \boldsymbol{\nu}^n = \boldsymbol{\nu} + s \cdot \Delta \boldsymbol{\nu}_{pd}.$$

Najprv zabezpečíme, aby platilo $\boldsymbol{\lambda}^n > \mathbf{0}$. Nech

$$\begin{aligned} s^{\max} &= \max \left\{ s \in (0, 1) ; \boldsymbol{\lambda} + s \cdot \Delta \boldsymbol{\lambda}_{pd} \geq \mathbf{0} \right\} \\ &= \min \left\{ 1, \min \left\{ \frac{-\lambda_i}{\Delta \lambda_i} ; \Delta \lambda_i < 0 \right\} \right\}. \end{aligned}$$

Teraz začneme backtracking s hodnotou $s = 0,99 \cdot s^{\max}$. Tým sme zabezpečili, že platí $\boldsymbol{\lambda}^n > \mathbf{0}$. Násobíme s číslom $\beta \in (0, 1)$ až kým nezískame $G(\mathbf{x}^n) < \mathbf{0}$. Pokračujeme v násobení číslom β až kým nezískame

$$\|\mathbf{r}_T(\mathbf{x}^n, \boldsymbol{\lambda}^n, \boldsymbol{\nu}^n)\| \leq (1 - \alpha s) \cdot \|\mathbf{r}_T(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu})\|.$$

Obyčajne volíme $\alpha \in \langle 0,01; 0,1 \rangle$ a $\beta \in \langle 0,3; 0,8 \rangle$.

Dostávame nasledujúci algoritmus.

ALGORITMUS: Backtracking primárno-duálnej metódy.

Vstup: \boldsymbol{x} , $\boldsymbol{\lambda}$, $\boldsymbol{\nu}$, $\Delta\boldsymbol{x}_{pd}$, $\Delta\boldsymbol{\lambda}_{pd}$, $\Delta\boldsymbol{\nu}_{pd}$, $\alpha \in \langle 0,01; 0,1 \rangle$ a $\beta \in \langle 0,3; 0,8 \rangle$.

Begin

$$s^{\max} = \min \left\{ 1, \min \left\{ \frac{-\lambda_i}{\Delta\lambda_i}; \Delta\lambda_i < 0 \right\} \right\};$$

$$s = 0,99 \cdot s^{\max};$$

Do

$$s = \beta \cdot s;$$

Until $G(\boldsymbol{x} + s \cdot \Delta\boldsymbol{x}) < 0$;

While $\|\boldsymbol{r}_T(\boldsymbol{x} + s \cdot \Delta\boldsymbol{x}_{pd}, \boldsymbol{\lambda} + s \cdot \Delta\boldsymbol{\lambda}_{pd}, \boldsymbol{\nu} + s \cdot \Delta\boldsymbol{\nu}_{pd})\| > (1 - \alpha s) \cdot \|r(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\nu})\|$ Do

Begin

$$s = \beta \cdot s;$$

End;

End.

Porovnanie s path-following metódami pre lineárne programovanie

V lineárnom programovaní sme riešili úlohu

$$\begin{array}{c} \min \boldsymbol{c}^T \cdot \boldsymbol{x} \\ \hline \boldsymbol{A} \cdot \boldsymbol{x} = \boldsymbol{b} \\ \hline \boldsymbol{x} \geq \boldsymbol{0} \end{array}$$

Kedť túto úlohu prepíšeme do tvaru, aký používame pre úlohy konvexného programovania, dostávame

$$\begin{array}{c} \min \boldsymbol{c}^T \cdot \boldsymbol{x} \\ \hline -\boldsymbol{x} \leq \boldsymbol{0} \\ \hline \boldsymbol{A} \cdot \boldsymbol{x} = \boldsymbol{b} \\ \hline \boldsymbol{x} \in \mathbb{R}^n \end{array}$$

Ako vyzerá pre túto úlohu rovnica $D\boldsymbol{r}_T(\boldsymbol{y}) \cdot \Delta\boldsymbol{y}_{pd} = -\boldsymbol{r}_T(\boldsymbol{y})$, kde \boldsymbol{r}_T je totálne reziduum? Z (24) dostávame, že musí platiť

$$\begin{pmatrix} \emptyset & -\mathbb{I} & \mathbb{A}^T \\ \text{diag}(\boldsymbol{\lambda}) & \text{diag}(\boldsymbol{x}) & \emptyset \\ \mathbb{A} & \emptyset & \emptyset \end{pmatrix} \cdot \begin{pmatrix} \Delta\boldsymbol{x}_{pd} \\ \Delta\boldsymbol{\lambda}_{pd} \\ \Delta\boldsymbol{\nu}_{pd} \end{pmatrix} = - \begin{pmatrix} \boldsymbol{c} - \boldsymbol{\lambda} + \mathbb{A}^T \cdot \boldsymbol{\nu} \\ \lambda_1 x_1 - \frac{1}{t} \\ \vdots \\ \lambda_n x_n - \frac{1}{t} \\ \mathbb{A} \cdot \boldsymbol{x} - \boldsymbol{b} \end{pmatrix}.$$

Teraz kedť zavedieme substitúciu $\mu = \frac{1}{t}$, $s = \boldsymbol{\lambda}$ a $\boldsymbol{y} = -\boldsymbol{\nu}$, pričom $\Delta\boldsymbol{x} = \Delta\boldsymbol{x}_{pd}$,

$\Delta \mathbf{s} = \Delta \boldsymbol{\lambda}_{pd}$ a $\Delta \mathbf{y} = -\Delta \boldsymbol{\nu}_{pd}$, získame

$$\begin{pmatrix} 0 & -\mathbb{I} & -\mathbb{A}^T \\ \mathbb{S} & \mathbb{X} & 0 \\ \mathbb{A} & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} \Delta \mathbf{x} \\ \Delta \mathbf{s} \\ \Delta \mathbf{y} \end{pmatrix} = - \begin{pmatrix} \mathbf{c} - \mathbf{s} - \mathbb{A}^T \cdot \mathbf{y} \\ \lambda_1 x_1 - \mu \\ \vdots \\ \lambda_n x_n - \mu \\ \mathbb{A} \cdot \mathbf{x} - \mathbf{b} \end{pmatrix},$$

kde $\mathbb{S} = \text{diag}(\mathbf{s})$ a $\mathbb{X} = \text{diag}(\mathbf{x})$. Zmenou poradia premenných a prenásobením prvého riadku blokového rozkladu číslom -1 dostávame

$$\begin{pmatrix} 0 & \mathbb{A}^T & \mathbb{I} \\ \mathbb{A} & 0 & 0 \\ \mathbb{S} & 0 & \mathbb{X} \end{pmatrix} \cdot \begin{pmatrix} \Delta \mathbf{x} \\ \Delta \mathbf{y} \\ \Delta \mathbf{s} \end{pmatrix} = - \begin{pmatrix} \mathbb{A}^T \cdot \mathbf{y} + \mathbf{s} - \mathbf{c} \\ \mathbb{A} \cdot \mathbf{x} - \mathbf{b} \\ \lambda_1 x_1 - \mu \\ \vdots \\ \lambda_n x_n - \mu \end{pmatrix},$$

čo je rovnica pre centrálnu cestu pri lineárnom programovaní, pozri (4) v Kapitole 2. Teraz keď je \mathbf{x} prípustným riešením (P) a (\mathbf{y}, \mathbf{s}) je prípustným riešením (D), tak nahradením μ výrazom τ získame Short-step path-following algoritmus. Znamená to, že teraz môžeme pozmeniť algoritmus primárno-duálnej metódy vnútorného bodu podobne, ako sme pozmenili Short-step path-following algoritmus, keď sme dostali ďalšie algoritmy lineárneho programovania. A môžeme očakávať, že minimálne v niektorých prípadoch získame rýchlejšie algoritmy.

Uzawova metóda

Na záver kapitoly si na účely porovnania pripomeňme gradientnú primárno-duálnu Uzawovu metódu, pozri [10], respektíve [4]. Najprv ju však upravíme, aby riešila úlohy tu spomínaného typu. Budeme riešiť úlohu

$$\begin{array}{ll} \min f(\mathbf{x}) \\ \frac{g_i(\mathbf{x}) \leq 0}{\mathbf{x} \in \mathbb{R}^n} & 1 \leq i \leq \ell \end{array}$$

Lagrangeova funkcia je

$$L(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \sum_{i=1}^{\ell} \lambda_i \cdot g_i(\mathbf{x}).$$

ALGORITMUS: Uzawova metóda.

Vstup: $\boldsymbol{\lambda}(\rho) = \mathbf{0}$, presnosť $\varepsilon > 0$.

Begin

Repeat Begin

Nájdi \mathbf{x} a ρ riešiace $\min \{L(\mathbf{x}, \boldsymbol{\lambda}(\rho)); \mathbf{x} \in \mathbb{R}^n \text{ a } \rho \in \mathbb{R}_+\}$;

Polož $\boldsymbol{\lambda} = \boldsymbol{\lambda}(\rho)$;

Quit if $\|\mathbf{x} - \mathbf{x}^s\| < \varepsilon$; { Tento krok pri prvom prechode algoritmom preskoč }
 Polož $\boldsymbol{\lambda}(\rho) = (\max\{\lambda_1 + \rho \cdot g_1(\mathbf{x}), 0\}, \dots, \max\{\lambda_\ell + \rho \cdot g_\ell(\mathbf{x}), 0\})^T$;
 $\mathbf{x}^s = \mathbf{x}$;
 End;
 End.

Výhodou Uzawovho algoritmu je, že prevádzza úlohu s ohraničeniami na postupnosť úloh bez ohraničení.

PRÍKLAD. Uzawovou metódou vyriešte

$$\begin{array}{c} \min x_1^2 + 2x_2^2 \\ \hline 3 - x_1 - x_2 \leq 0 \\ \hline x_1 - 1 \leq 0 \\ \hline x_1, x_2 \in \mathbb{R} \end{array}$$

RIEŠENIE. Lagrangeova funkcia je

$$L(x_1, x_2, \lambda_1, \lambda_2) = x_1^2 + 2x_2^2 + \lambda_1(3 - x_1 - x_2) + \lambda_2(x_1 - 1).$$

Spravíme päť iterácií algoritmu a zostrojíme \mathbf{x}^1 až \mathbf{x}^5 .

1° Kedže na začiatku $\boldsymbol{\lambda}^0 = \mathbf{0}$, máme nájst

$$\min x_1^2 + 2x_2^2.$$

Z rovníc

$$\begin{aligned} \frac{\partial}{\partial x_1} L(\mathbf{x}, \boldsymbol{\lambda}) &= 2x_1 = 0 \\ \frac{\partial}{\partial x_2} L(\mathbf{x}, \boldsymbol{\lambda}) &= 4x_2 = 0 \end{aligned}$$

dostávame $\mathbf{x}^1 = (0, 0)^T$. Teraz by sme mali určiť $\boldsymbol{\lambda}^0 = \boldsymbol{\lambda}(\rho)$, no pri prvom prechode algoritmom je $\boldsymbol{\lambda}^0 = (0, 0)^T$ a ρ tu nevystupuje. Kedže $g_1(\mathbf{x}^1) = 3$ a $g_2(\mathbf{x}^1) = -1$, dostávame

$$\boldsymbol{\lambda}(\rho) = (\max\{0 + \rho \cdot 3, 0\}, \max\{0 + \rho \cdot (-1), 0\})^T = (3\rho, 0)^T.$$

2° Máme nájst

$$\min x_1^2 + 2x_2^2 + 3\rho(3 - x_1 - x_2).$$

Z rovníc

$$\begin{aligned} \frac{\partial}{\partial x_1} L(\mathbf{x}, \boldsymbol{\lambda}(\rho)) &= 2x_1 - 3\rho = 0 \\ \frac{\partial}{\partial x_2} L(\mathbf{x}, \boldsymbol{\lambda}(\rho)) &= 4x_2 - 3\rho = 0 \\ \frac{\partial}{\partial \rho} L(\mathbf{x}, \boldsymbol{\lambda}(\rho)) &= 3(3 - x_1 - x_2) = 0 \end{aligned}$$

dostávame $\mathbf{x}^2 = (2, 1)^T$ a $\rho = \frac{4}{3}$. Z toho $\boldsymbol{\lambda}^1 = (4, 0)^T$. Teraz $g_1(\mathbf{x}^2) = 0$ a $g_2(\mathbf{x}^2) = 1$, teda

$$\boldsymbol{\lambda}(\rho) = (\max\{4 + \rho \cdot 0, 0\}, \max\{0 + \rho \cdot 1, 0\})^T = (4, \rho)^T.$$

3° Máme nájst

$$\min x_1^2 + 2x_2^2 + 4(3 - x_1 - x_2) + \rho(x_1 - 1).$$

Z rovníc

$$\begin{aligned}\frac{\partial}{\partial x_1} L(\boldsymbol{x}, \boldsymbol{\lambda}(\rho)) &= 2x_1 - 4 + \rho = 0 \\ \frac{\partial}{\partial x_2} L(\boldsymbol{x}, \boldsymbol{\lambda}(\rho)) &= 4x_2 - 4 = 0 \\ \frac{\partial}{\partial \rho} L(\boldsymbol{x}, \boldsymbol{\lambda}(\rho)) &= x_1 - 1 = 0\end{aligned}$$

dostávame $\boldsymbol{x}^3 = (1, 1)^T$ a $\rho = 2$. Z toho $\boldsymbol{\lambda}^2 = (4, 2)^T$. Teraz $g_1(\boldsymbol{x}^2) = 1$ a $g_2(\boldsymbol{x}^2) = 0$, teda

$$\boldsymbol{\lambda}(\rho) = (\max\{4 + \rho \cdot 1, 0\}, \max\{2 + \rho \cdot 0, 0\})^T = (4 + \rho, 2)^T.$$

4° Máme nájst

$$\min x_1^2 + 2x_2^2 + (4 + \rho)(3 - x_1 - x_2) + 2(x_1 - 1).$$

Z rovníc

$$\begin{aligned}\frac{\partial}{\partial x_1} L(\boldsymbol{x}, \boldsymbol{\lambda}(\rho)) &= 2x_1 - 4 - \rho + 2 = 0 \\ \frac{\partial}{\partial x_2} L(\boldsymbol{x}, \boldsymbol{\lambda}(\rho)) &= 4x_2 - 4 - \rho = 0 \\ \frac{\partial}{\partial \rho} L(\boldsymbol{x}, \boldsymbol{\lambda}(\rho)) &= 3 - x_1 - x_2 = 0\end{aligned}$$

dostávame $\boldsymbol{x}^4 = \left(\frac{5}{3}, \frac{4}{3}\right)^T$ a $\rho = \frac{4}{3}$. Z toho $\boldsymbol{\lambda}^3 = \left(\frac{16}{3}, 2\right)^T$. Teraz $g_1(\boldsymbol{x}^2) = 0$ a $g_2(\boldsymbol{x}^2) = \frac{2}{3}$, teda

$$\boldsymbol{\lambda}(\rho) = (\max\{\frac{16}{3} + \rho \cdot 0, 0\}, \max\{2 + \rho \cdot \frac{2}{3}, 0\})^T = \left(\frac{16}{3}, 2 + \frac{2}{3}\rho\right)^T.$$

5° Máme nájst

$$\min x_1^2 + 2x_2^2 + \frac{16}{3}(3 - x_1 - x_2) + (2 + \frac{2}{3}\rho)(x_1 - 1).$$

Z rovníc

$$\begin{aligned}\frac{\partial}{\partial x_1} L(\boldsymbol{x}, \boldsymbol{\lambda}(\rho)) &= 2x_1 - \frac{16}{3} + 2 + \frac{2}{3}\rho = 0 \\ \frac{\partial}{\partial x_2} L(\boldsymbol{x}, \boldsymbol{\lambda}(\rho)) &= 4x_2 - \frac{16}{3} = 0 \\ \frac{\partial}{\partial \rho} L(\boldsymbol{x}, \boldsymbol{\lambda}(\rho)) &= x_1 - 1 = 0\end{aligned}$$

dostávame $\boldsymbol{x}^5 = \left(1, \frac{4}{3}\right)^T$ a $\rho = 2$. Z toho $\boldsymbol{\lambda}^4 = \left(\frac{16}{3}, \frac{10}{3}\right)^T$.

Teraz určíme optimálne riešenie priamo z KKT podmienok. Dostávame

$$\begin{aligned}\frac{\partial}{\partial x_1} L(\boldsymbol{x}, \boldsymbol{\lambda}) &= 2x_1 - \lambda_1 + \lambda_2 = 0 \\ \frac{\partial}{\partial x_2} L(\boldsymbol{x}, \boldsymbol{\lambda}) &= 4x_2 - \lambda_1 = 0 \\ \lambda_1(3 - x_1 - x_2) &= 0 \\ \lambda_2(x_1 - 1) &= 0\end{aligned}$$

pričom má platiť $g_1(\mathbf{x}) \leq 0$, $g_2(\mathbf{x}) \leq 0$ a $\boldsymbol{\lambda} \geq \mathbf{0}$. Kedže posledné dve rovnice sú nelineárne, budeme uvažovať štyri prípady podľa toho, či sú alebo nie sú λ_1 a λ_2 rovné 0.

1. $\lambda_1 = \lambda_2 = 0$.

Tu z prvých dvoch rovníc dostaneme $\mathbf{x} = (0, 0)^T$. Kedže $g_1(0, 0) = 3 > 0$, toto riešenie nespĺňa KKT podmienky.

2. $\lambda_1 = 0$ a $\lambda_2 > 0$.

Z prvých dvoch rovníc a $(x_1 - 1) = 0$ dostaneme $\mathbf{x} = (1, 0)^T$. Kedže $g_1(1, 0) = 2 > 0$, ani toto riešenie nespĺňa KKT podmienky.

3. $\lambda_1 > 0$ a $\lambda_2 = 0$.

Z prvých dvoch rovníc a $(3 - x_1 - x_2) = 0$ dostaneme $\mathbf{x} = (2, 1)^T$. Kedže $g_2(2, 1) = 1 > 0$, ani toto riešenie nespĺňa KKT podmienky.

4. $\lambda_1 > 0$ a $\lambda_2 > 0$.

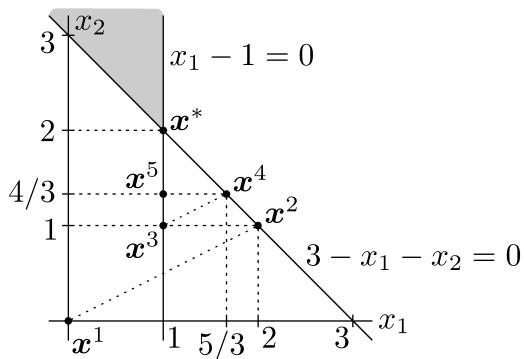
Z prvých dvoch rovníc, $(3 - x_1 - x_2) = 0$ a $(x_1 - 1) = 0$ dostaneme $\mathbf{x} = (1, 2)^T$ a $\boldsymbol{\lambda} = (8, 6)^T$. Kedže $g_1(1, 2) = g_2(1, 2) = 0$, toto riešenie spĺňa KKT podmienky.

Kedže účelová funkcia je konvexná a rovnako sú konvexné (dokonca lineárne) aj ohraničenia, tak podľa Vety z Kapitoly 4 sú pre body spĺňajúce LICQ podmienku KKT podmienky nielen nutné, ale aj postačujúce na existenciu globálneho minima, keďže $D = \mathbb{R}^2$. Znamená to, že $\mathbf{x}^* = (1, 2)^T$ je riešením našej úlohy.

Na Obrázku 24 máme zaznačené body $\mathbf{x}^1, \dots, \mathbf{x}^5$ a \mathbf{x}^* , pričom sivou farbou je znázornená množina prípustných riešení. Všimnime si, že ak budú ďalšie body na priamkach $3 - x_1 - x_2 = 0$ a $x_1 - 1 = 0$ získané analogicky, tak platí

$$\lim_{k \rightarrow \infty} \frac{\|\mathbf{x}^{k+2} - \mathbf{x}^*\|}{\|\mathbf{x}^k - \mathbf{x}^*\|} = \frac{2}{3},$$

čiže v tom prípade bude konvergencia Uzawovej metódy v podstate lineárna.



Obrázok 24

Cvičenia

CVIČENIE 8.1. Primárno-duálnou metódou vyriešte úlohu

$$\begin{aligned} \min f(\boldsymbol{x}) &= (x_1 + 1)^4 + 3(x_2 - 1)^2 + (x_3 - 1)^2 \\ (x_1 - 1)^2 + (x_2 + 2,5)^2 &\leq 9 \\ (x_1 - 1)^2 + (x_2 - 2)^2 &\leq 9 \\ x_1 &\leq 2 - x_2^2 \\ x_1 - x_2 + x_3 &= 0,3 \end{aligned}$$

(porovnajte s Cvičením 7.1). Začnite zo súboru z Cvičenia 7.1, upravte ho a doplňte chýbajúce časti. Nepotrebné časti (definície bariérovej a modifikovanej účelovej funkcie) môžete vypustiť.

1. Definujte
 - a) Gradienty a Hessove matice funkcií $g_i(\boldsymbol{x})$.
 - b) Reziduum komplementarity $r_C(\boldsymbol{x}, \boldsymbol{y})$.
2. Upravte definície reziduú a Jacobiho matice. Kedže medzi veličiny pribudne vektor duálnych premenných $\boldsymbol{\lambda}$, tak napríklad reziduum môžeme definovať ako $r[\{x_{1-}, x_{2-}, x_{3-}, \lambda_{1-}, \lambda_{2-}, \lambda_{3-}, \nu_{-}\}, t_{B-}]$.
3. Modifikujte Backtracking algoritmus.
4. Naprogramujte Primárno-duálnu metódu (aj s Backtracking algoritmom). Štartovací bod \boldsymbol{x}^0 zvolte tak, aby splňal nerovnice a nespĺňal rovnicu. Rozmyslite si, aká je vhodná voľba $\boldsymbol{\lambda}_0$. (Návod: Aké je vhodné $t_{B,0}$? Ako súvisí $t_{B,0}$ s $\boldsymbol{\lambda}_0$?)

9 Metóda projekcie gradientu

Uvažujme úlohu

$$\begin{array}{c} \min f(\mathbf{x}) \\ \hline g_i(\mathbf{x}) \leq 0 & 1 \leq i \leq \ell \\ \mathbf{x} \in D \end{array}$$

kde f, g_1, \dots, g_ℓ sú spojito diferencovateľné funkcie. V metóde projekcie gradientu predpokladáme, že $\Delta\mathbf{x}$ leží v priestore dotykovom k aktívnym ohraničeniam (čiže k tým, v ktorých sa dosahuje rovnosť). Najprv budeme uvažovať prípad, keď sú všetky funkcie g_i lineárne, neskôr postup zovšeobecníme. Pre jednoduchosť budeme predpokladať, že $D = \mathbb{R}^n$.

Lineárne ohraničenia

Budeme teda uvažovať úlohu

$$\begin{array}{ccc} \min f(\mathbf{x}) & & \min f(\mathbf{x}) \\ \hline \mathbb{A} \cdot \mathbf{x} \leq \mathbf{b} & \text{čiže} & \mathbf{a}_i \cdot \mathbf{x} - b_i \leq 0 & 1 \leq i \leq \ell \\ \mathbf{x} \in \mathbb{R}^n & & \mathbf{x} \in \mathbb{R}^n \end{array}$$

kde \mathbb{A} je matica typu $(\ell \times n)$ a f je konvexná funkcia.

Nech je \mathbf{x}^k z množiny prípustných riešení, čiže $\mathbf{x}^k \in \mathcal{F}$, a nech \mathbb{A}_p obsahuje tie riadky matice \mathbb{A} , pre ktoré $\mathbf{a}_i \cdot \mathbf{x}^k - b_i = 0$. Matica \mathbb{A}_p teda obsahuje tie ohraničenia, ktoré sú pre \mathbf{x}^k splnené ako rovnice. Nech je \mathbb{A}_p typu $(r \times n)$. Predpokladáme, že riadky \mathbb{A}_p sú lineárne nezávislé.

Pre nové riešenie $\mathbf{x}^{k+1} = \mathbf{x}^k + \Delta\mathbf{x}$ (ktoré neskôr spresníme na $\mathbf{x}^{k+1} = \mathbf{x}^k + s \cdot \Delta\mathbf{x}$) požadujeme, aby platilo $\mathbb{A}_p \cdot \mathbf{x}^{k+1} - \mathbf{b}_p = \mathbf{0}$, z čoho dostávame $\mathbb{A}_p \cdot \Delta\mathbf{x} = \mathbf{0}$. Vektorov splňajúcich $\mathbb{A}_p \cdot \Delta\mathbf{x} = \mathbf{0}$ bude viac, preto nájdeme ten, pre ktorý má f najstrmší spád. Z Taylorovho rozvoja prvého rádu dostávame

$$f(\mathbf{x}^{k+1}) = f(\mathbf{x}^k + \Delta\mathbf{x}) \approx f(\mathbf{x}^k) + \nabla f(\mathbf{x}^k)^T \cdot \Delta\mathbf{x},$$

pričom požadujeme $\Delta\mathbf{x}^T \cdot \Delta\mathbf{x} = 1$, teda chceme, aby mal vektor $\Delta\mathbf{x}$ veľkosť 1. (V opačnom prípade by úloha hľadajúca minimum lineárnej funkcie $\Delta\mathbf{x}^T \cdot \nabla f(\mathbf{x}^k)$ nemusela mať optimálne riešenie.) Teda potrebujeme vyriešiť úlohu

$$\begin{array}{c} \min \Delta \mathbf{x}^T \cdot \nabla f(\mathbf{x}^k) \\ \hline \mathbb{A}_p \cdot \Delta \mathbf{x} = \mathbf{0} \\ \Delta \mathbf{x}^T \cdot \Delta \mathbf{x} \leq 1 \\ \hline \mathbf{x} \in \mathbb{R}^n \end{array}$$

Podmienku $\Delta \mathbf{x}^T \cdot \Delta \mathbf{x} = 1$ sme nahradili podmienkou $\Delta \mathbf{x}^T \cdot \Delta \mathbf{x} \leq 1$ preto, aby sme dostali úlohu konvexného programovania. Keď sú ohraničenia splnené pre $\Delta \mathbf{x}$, tak sú splnené aj pre $-\Delta \mathbf{x}$, z čoho plynie, že minimum nie je kladné. Ohraničenia tvaru rovníc sú však splnené pre ľubovoľný násobok $\Delta \mathbf{x}$, preto sa minimum dosahuje keď $\Delta \mathbf{x}^T \cdot \Delta \mathbf{x} = 1$. Keďže $\mathbf{0}$ je vnútorný bod množiny prípustných riešení, táto úloha spĺňa Slaterovu podmienku, čiže platí silná dualita, z čoho plynie, že optimálne riešenie $\Delta \mathbf{x}$ splňa KKT podmienky.

Zostrojíme Lagrangeovu funkciu

$$L(\Delta \mathbf{x}, \boldsymbol{\nu}, \lambda) = \Delta \mathbf{x}^T \cdot \nabla f(\mathbf{x}^k) + \boldsymbol{\nu}^T \cdot \mathbb{A}_p \cdot \Delta \mathbf{x} + \lambda(\Delta \mathbf{x}^T \cdot \Delta \mathbf{x} - 1).$$

Tu $\boldsymbol{\nu}^T \cdot \mathbb{A}_p \cdot \Delta \mathbf{x} = \Delta \mathbf{x}^T \cdot \mathbb{A}_p^T \cdot \boldsymbol{\nu}$, keďže $\boldsymbol{\nu}^T \cdot \mathbb{A}_p \cdot \Delta \mathbf{x}$ je číslo. Podľa KKT podmienok pre optimálne riešenie $\Delta \mathbf{x}$ platí $\nabla_{\Delta \mathbf{x}} L(\Delta \mathbf{x}, \boldsymbol{\nu}, \lambda) = \mathbf{0}$, z čoho dostávame (všimnime si, že $\Delta \mathbf{x}^T \cdot \Delta \mathbf{x} = \Delta x_1^2 + \dots + \Delta x_n^2$)

$$\begin{aligned} \nabla f(\mathbf{x}^k) + \mathbb{A}_p^T \cdot \boldsymbol{\nu} + 2\lambda \cdot \Delta \mathbf{x} &= \mathbf{0} && / \mathbb{A}_p \cdot \\ \mathbb{A}_p \cdot \nabla f(\mathbf{x}^k) + \mathbb{A}_p \cdot \mathbb{A}_p^T \cdot \boldsymbol{\nu} &= \mathbf{0} \end{aligned} \quad (25)$$

keďže $\mathbb{A}_p \cdot \Delta \mathbf{x} = \mathbf{0}$. Z posledného vzťahu dostávame

$$\boldsymbol{\nu} = -(\mathbb{A}_p \cdot \mathbb{A}_p^T)^{-1} \cdot \mathbb{A}_p \cdot \nabla f(\mathbf{x}^k)$$

a dosadením do (25) získame

$$\begin{aligned} (\mathbb{I} - \mathbb{A}_p^T \cdot (\mathbb{A}_p \cdot \mathbb{A}_p^T)^{-1} \cdot \mathbb{A}_p) \cdot \nabla f(\mathbf{x}^k) + 2\lambda \cdot \Delta \mathbf{x} &= \mathbf{0} && \text{čiže} \\ \Delta \mathbf{x} &= \frac{-1}{2\lambda} \cdot (\mathbb{I} - \mathbb{A}_p^T \cdot (\mathbb{A}_p \cdot \mathbb{A}_p^T)^{-1} \cdot \mathbb{A}_p) \cdot \nabla f(\mathbf{x}^k) = \frac{-1}{2\lambda} \cdot \mathbb{P} \cdot \nabla f(\mathbf{x}^k). \end{aligned}$$

Maticu $\mathbb{P} = \mathbb{I} - \mathbb{A}_p^T \cdot (\mathbb{A}_p \cdot \mathbb{A}_p^T)^{-1} \cdot \mathbb{A}_p$ nazývame **projektor**. Keďže na veľkosťi $\Delta \mathbf{x}$ nezáleží, pretože neskôr ešte určíme násobok s , tak kladieme

$$\Delta \mathbf{x} = -\mathbb{P} \cdot \nabla f(\mathbf{x}^k).$$

Obyčajne \mathbb{P} nepočítame podľa vzťahu $\mathbb{P} = \mathbb{I} - \mathbb{A}_p^T \cdot (\mathbb{A}_p \cdot \mathbb{A}_p^T)^{-1} \cdot \mathbb{A}_p$, lebo to nie je najefektívnejší spôsob. Dá sa ukázať, že $\mathbb{P} = \mathbb{Q}_2^T \cdot \mathbb{Q}_2$, kde \mathbb{Q}_2 pozostáva z posledných $n - r$ riadkov \mathbb{Q} -člena v $\mathbb{Q} - \mathbb{R}$ rozklade matice \mathbb{A}_p^T . Pripomeňme, že $\mathbb{Q} - \mathbb{R}$ rozklad \mathbb{A}_p^T je rozklad \mathbb{A}_p^T na súčin $\mathbb{A}_p^T = \mathbb{Q} \cdot \mathbb{R}$, kde \mathbb{A}_p^T , \mathbb{Q} a \mathbb{R} sú po rade typov $(n \times r)$, $(n \times n)$ a $(n \times r)$, \mathbb{Q} je ortogonálna a \mathbb{R} je horná trojuholníková matica. Tento rozklad môžeme získať pomocou Gram-Schmidtovoho ortogonalizačného algoritmu.

Ukážeme, že \mathbb{P} je projektor do priestoru dotykového k aktívnym ohraničeniam. Na to potrebujeme ukázať dve veci. Jednak, že $\mathbb{P}^2 = \mathbb{P}$, a tiež, že sú riadky \mathbb{A}_p (ktoré

sú gradienty ohraničení, čiže sú to vektory kolmé na nadroviny $\mathbf{a}_i \cdot \mathbf{x}^k - b_i = 0$ kolmé na $\mathbb{P} \cdot \mathbf{w}$ pre ľubovoľný vektor \mathbf{w} . Kedže však

$$\mathbb{A}_p \cdot \mathbb{P} \cdot \mathbf{w} = \mathbb{A}_p \cdot (\mathbb{I} - \mathbb{A}_p^T \cdot (\mathbb{A}_p \cdot \mathbb{A}_p^T)^{-1} \cdot \mathbb{A}_p) \cdot \mathbf{w} = (\mathbb{A}_p - \mathbb{A}_p) \cdot \mathbf{w} = \mathbf{0},$$

tak druhá vlastnosť je splnená. Teraz ukážeme prvú. Platí

$$\begin{aligned} \mathbb{P}^2 &= (\mathbb{I} - \mathbb{A}_p^T \cdot (\mathbb{A}_p \cdot \mathbb{A}_p^T)^{-1} \cdot \mathbb{A}_p)^2 \\ &= \mathbb{I} - 2 \cdot \mathbb{A}_p^T \cdot (\mathbb{A}_p \cdot \mathbb{A}_p^T)^{-1} \cdot \mathbb{A}_p + \mathbb{A}_p^T \cdot (\mathbb{A}_p \cdot \mathbb{A}_p^T)^{-1} \cdot \mathbb{A}_p \cdot \mathbb{A}_p^T \cdot (\mathbb{A}_p \cdot \mathbb{A}_p^T)^{-1} \cdot \mathbb{A}_p \\ &= \mathbb{I} - 2 \cdot \mathbb{A}_p^T \cdot (\mathbb{A}_p \cdot \mathbb{A}_p^T)^{-1} \cdot \mathbb{A}_p + \mathbb{A}_p^T \cdot (\mathbb{A}_p \cdot \mathbb{A}_p^T)^{-1} \cdot \mathbb{A}_p \\ &= \mathbb{I} - \mathbb{A}_p^T \cdot (\mathbb{A}_p \cdot \mathbb{A}_p^T)^{-1} \cdot \mathbb{A}_p = \mathbb{P} \end{aligned}$$

Naviac, \mathbb{P} je symetrická. Aby sme to ukázali, označme $\mathbb{T} = \mathbb{A}_p^T \cdot (\mathbb{A}_p \cdot \mathbb{A}_p^T)^{-1} \cdot \mathbb{A}_p$. Kedže je $\mathbb{A}_p \cdot \mathbb{A}_p^T$ symetrická, tak je symetrická aj $(\mathbb{A}_p \cdot \mathbb{A}_p^T)^{-1}$. Čiže $\mathbb{T} = \mathbb{A}_p^T \cdot \mathbb{S} \cdot \mathbb{A}_p$, kde \mathbb{S} je symetrická matica. Kedže

$$\mathbb{T}^T = (\mathbb{A}_p^T \cdot \mathbb{S} \cdot \mathbb{A}_p)^T = \mathbb{A}_p^T \cdot \mathbb{S}^T \cdot \mathbb{A}_p = \mathbb{A}_p^T \cdot \mathbb{S} \cdot \mathbb{A}_p = \mathbb{T},$$

tak je symetrická aj \mathbb{T} a následne \mathbb{P} .

Teraz ukážeme, že \mathbb{P} je kladne semidefinitná. Nech je \mathbf{w} ľubovoľný vektor. Potom

$$\mathbf{w}^T \cdot \mathbb{P} \cdot \mathbf{w} = \mathbf{w}^T \cdot \mathbb{P} \cdot \mathbb{P} \cdot \mathbf{w} = \mathbf{w}^T \cdot \mathbb{P}^T \cdot \mathbb{P} \cdot \mathbf{w} = (\mathbb{P} \cdot \mathbf{w})^T \cdot (\mathbb{P} \cdot \mathbf{w}) \geq 0.$$

Znamená to, že ak je α uhol medzi \mathbf{w} a $\mathbb{P} \cdot \mathbf{w}$, tak

$$\cos \alpha = \frac{\mathbf{w}^T \cdot (\mathbb{P} \cdot \mathbf{w})}{\|\mathbf{w}\| \cdot \|\mathbb{P} \cdot \mathbf{w}\|},$$

čiže $\alpha \in \langle 0, 90^\circ \rangle$. Kedže $\nabla f(\mathbf{x}^k)$ je vektor najväčšieho rastu, tak najväčšie klesanie sa dosahuje pre $-\nabla f(\mathbf{x}^k)$, a preto $\mathbf{x}^{k+1} = \mathbf{x}^k + s \cdot \Delta \mathbf{x} = \mathbf{x}^k + s \cdot (-\mathbb{P} \cdot \nabla f(\mathbf{x}^k))$ pre nezáporné s .

Teraz určíme násobok s . Kedže $\mathbf{x}^{k+1} = \mathbf{x}^k + s \cdot \Delta \mathbf{x}$, tak pre ohraničenia, ktoré nie sú aktívne, potrebujeme skontrolovať, či platí $\mathbf{a}_i(\mathbf{x}^k + s \cdot \Delta \mathbf{x}) - b_i \leq 0$. Ak $\mathbf{a}_i \cdot \Delta \mathbf{x} \leq 0$, tak s môže byť ľubovoľne veľké. No ak $\mathbf{a}_i \cdot \Delta \mathbf{x} > 0$, vtedy

$$\begin{aligned} \mathbf{a}_i \cdot \mathbf{x}^k - b_i + s \cdot \mathbf{a}_i \cdot \Delta \mathbf{x} &\leq 0 \quad \text{čiže} \\ s &\leq -(\mathbf{a}_i \cdot \mathbf{x}^k - b_i) / (\mathbf{a}_i \cdot \Delta \mathbf{x}). \end{aligned}$$

Tieto ohraničenia sú pre rôzne i rôzne, preto

$$\bar{s} = \min_{\mathbf{a}_i \cdot \Delta \mathbf{x} > 0} \left\{ \frac{-(\mathbf{a}_i \cdot \mathbf{x}^k - b_i)}{\mathbf{a}_i \cdot \Delta \mathbf{x}} \right\}.$$

Pripomeňme, že pri výpočte \bar{s} uvažujeme iba neaktívne ohraničenia.

Keď máme hornú hranicu na s , vyriešime jednorozmernú úlohu

$$\begin{array}{c} \min f(\mathbf{x}^k + s \cdot \Delta \mathbf{x}) \\ \hline 0 \leq s \leq \bar{s} \end{array}$$

Položíme $\mathbf{x}^{k+1} = \mathbf{x}^k + s \cdot \Delta \mathbf{x}$ a možno budú aktívne iné ohraničenia. Dostaneme inú maticu \mathbb{A}_p a pokračujeme v algoritme.

Proces opakujeme pokiaľ $\Delta \mathbf{x} \neq \mathbf{0}$. Ak $\Delta \mathbf{x} = \mathbf{0}$, tak sme v kritickom bode L , preto skontrolujeme KKT podmienky. V podstate stačí určiť Lagrangeove multiplikátory. Máme

$$L(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \sum_{i=1}^{\ell} \lambda_i \cdot (\mathbf{a}_i \cdot \mathbf{x} - b_i) = f(\mathbf{x}) + (\mathbf{x}^T \cdot \mathbb{A}^T - \mathbf{b}) \cdot \boldsymbol{\lambda}$$

a ak sú KKT podmienky splnené, tak $\nabla_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\lambda}) = \mathbf{0}$, čiže

$$\nabla_{\mathbf{x}} f(\mathbf{x}) + \mathbb{A}^T \cdot \boldsymbol{\lambda} = \mathbf{0}.$$

Okrem toho má platiť $\lambda_i \cdot (\mathbf{a}_i \cdot \mathbf{x} - b_i) = 0$. Teda pre ohraničenia, ktoré nie sú splnené presne, bude $\lambda_i = 0$. Preto stačí určiť λ_i pre tie ohraničenia, ktoré sú aktívne, čiže pre tie, v ktorých sa dosahuje rovnosť. Označme $\boldsymbol{\lambda}_p$ tú časť vektora $\boldsymbol{\lambda}$, ktorá zodpovedá aktívnym ohraničeniam. Potom

$$\begin{aligned} \nabla_{\mathbf{x}} f(\mathbf{x}) + \mathbb{A}_p^T \cdot \boldsymbol{\lambda}_p &= \mathbf{0} && / \mathbb{A}_p \\ \mathbb{A}_p \cdot \nabla_{\mathbf{x}} f(\mathbf{x}) + \mathbb{A}_p \cdot \mathbb{A}_p^T \cdot \boldsymbol{\lambda}_p &= \mathbf{0} && \text{čiže} \\ \boldsymbol{\lambda}_p &= -(\mathbb{A}_p \cdot \mathbb{A}_p^T)^{-1} \cdot \mathbb{A}_p \cdot \nabla_{\mathbf{x}} f(\mathbf{x}) \end{aligned}$$

Teraz ak $\boldsymbol{\lambda}_p \geq \mathbf{0}$, tak sú KKT podmienky splnené a bod \mathbf{x} je optimálne riešenie našej úlohy. Avšak ak $\boldsymbol{\lambda}_p \geq \mathbf{0}$ neplatí, vyhodíme ohraničenie s najmenšou hodnotou λ_i a pokračujeme s redukovanou maticou \mathbb{A}_p .

Práve opísaný postup si ukážeme na príklade.

PRÍKLAD 1. Vyriešte úlohu

$$\begin{array}{c} \min (x_1 - 2)^2 + (x_2 - 3)^2 \\ \hline x_1 + x_2 - 6 \leq 0 \\ x_1 + 2x_2 - 10 \leq 0 \\ x_1 - 4 \leq 0 \\ -x_1 \leq 0 \\ -x_2 \leq 0 \\ \hline x_1, x_2 \in \mathbb{R} \end{array}$$

metódou projekcie gradientu. Začnite v bode $\mathbf{x}^0 = (0, 0)^T \in \mathcal{F}$.

RIEŠENIE. Máme

$$\nabla_{\mathbf{x}} f(\mathbf{x}) = \begin{pmatrix} 2(x_1 - 2) \\ 2(x_2 - 3) \end{pmatrix}.$$

Budeme robiť jednotlivé iterácie algoritmu až kým nenájdeme optimálne riešenie.

1. iterácia. Pre $\mathbf{x}^0 = (0, 0)^T$ sú aktívne ohraničenia 4 a 5, čiže

$$\mathbb{A}_p = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}.$$

Preto

$$\begin{aligned} \mathbb{P} &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} \cdot \left(\begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} \cdot \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} \right)^{-1} \cdot \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} = \\ &= \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}. \end{aligned}$$

Vo všeobecnosti, ak je \mathbb{A}_p regulárna štvorcová matica, tak $\mathbb{P} = \mathbb{0}$. To preto, lebo vtedy

$$\mathbb{I} - \mathbb{A}_p^T \cdot (\mathbb{A}_p \cdot \mathbb{A}_p^T)^{-1} \cdot \mathbb{A}_p = \mathbb{I} - \mathbb{A}_p^{-1} \cdot \mathbb{A}_p \cdot \mathbb{A}_p^T \cdot (\mathbb{A}_p \cdot \mathbb{A}_p^T)^{-1} \cdot \mathbb{A}_p = \mathbb{I} - \mathbb{A}_p^{-1} \cdot \mathbb{A}_p = \mathbb{0}.$$

Preto $\Delta \mathbf{x} = -\mathbb{P} \cdot \nabla f(\mathbf{x}^0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ a potrebujeme určiť KKT podmienky.

$$\boldsymbol{\lambda}_p = - \left(\begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} \cdot \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} \right)^{-1} \cdot \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} \cdot \begin{pmatrix} -4 \\ -6 \end{pmatrix} = \begin{pmatrix} -4 \\ -6 \end{pmatrix}.$$

Zjavne neplatí $\boldsymbol{\lambda} \geq \mathbf{0}$, čiže súčasný bod nie je optimálne riešenie. Podľa návodu vypúšťame z \mathbb{A}_p ohraničenie 5 a v ďalšom kroku ho nebudeme uvažovať.

2. iterácia. Teda pre $\mathbf{x}^0 = (0, 0)^T$ je aktívne ohraničenie 4, čiže $\mathbb{A}_p = (-1 \ 0)$.

Preto

$$\begin{aligned} \mathbb{P} &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} -1 \\ 0 \end{pmatrix} \cdot \left((-1 \ 0) \cdot \begin{pmatrix} -1 \\ 0 \end{pmatrix} \right)^{-1} \cdot (-1 \ 0) = \\ &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} -1 \\ 0 \end{pmatrix} \cdot (-1 \ 0) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}. \end{aligned}$$

Kedže $\nabla_{\mathbf{x}} f(\mathbf{x}^0) = (-4, -6)^T$, získavame

$$\Delta \mathbf{x} = - \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} -4 \\ -6 \end{pmatrix} = \begin{pmatrix} 0 \\ 6 \end{pmatrix}.$$

Neaktívne sú ohraničenia 1, 2, 3 a 5, takže pre tieto ohraničenia určíme hornú hranicu na s .

$$\mathbf{a}_1 \cdot \Delta \mathbf{x} = (1 \ 1) \cdot \begin{pmatrix} 0 \\ 6 \end{pmatrix} = 6 \quad \text{a} \quad s_1 = -\frac{-6}{6} = 1$$

$$\mathbf{a}_2 \cdot \Delta \mathbf{x} = (1 \ 2) \cdot \begin{pmatrix} 0 \\ 6 \end{pmatrix} = 12 \quad \text{a} \quad s_2 = -\frac{-10}{12} = \frac{5}{6}$$

$$\mathbf{a}_3 \cdot \Delta \mathbf{x} = (1 \ 0) \cdot \begin{pmatrix} 0 \\ 6 \end{pmatrix} = 0 \quad \text{takže toto ohraničenie netreba uvažovať}$$

$$\mathbf{a}_5 \cdot \Delta \mathbf{x} = (0 \ -1) \cdot \begin{pmatrix} 0 \\ 6 \end{pmatrix} = -6 \quad \text{teda ani toto ohraničenie netreba uvažovať.}$$

Dostali sme $\bar{s} = \frac{5}{6}$, čo znamená, že potrebujeme vyriešiť úlohu

$$\begin{array}{c} \min f(s) = (0 - 2)^2 + (6s - 3)^2 \\ 0 \leq s \leq \frac{5}{6} \end{array}$$

Kedže $f'(s) = 2 \cdot (6s - 3) \cdot 6 = 0$ dá $s = \frac{1}{2}$ a $0 \leq \frac{1}{2} \leq \frac{5}{6}$, tak $s = \frac{1}{2}$. Preto

$$\mathbf{x}^1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix} + \frac{1}{2} \begin{pmatrix} 0 \\ 6 \end{pmatrix} = \begin{pmatrix} 0 \\ 3 \end{pmatrix}.$$

3. iterácia. Pre $\mathbf{x}^1 = (0, 3)^T$ je aktívne iba ohraničenie 4. Preto $\mathbb{A}_p = (-1 \ 0)$ a tak ako vyššie, dostaneme

$$\mathbb{P} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}.$$

Kedže $\nabla_{\mathbf{x}} f(\mathbf{x}^1) = (-4, 0)^T$, získavame

$$\Delta \mathbf{x} = - \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} -4 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Teda potrebujeme určiť KKT podmienky.

$$\boldsymbol{\lambda}_p = - \left((-1 \ 0) \cdot \begin{pmatrix} -1 \\ 0 \end{pmatrix} \right)^{-1} \cdot (-1 \ 0) \cdot \begin{pmatrix} -4 \\ 0 \end{pmatrix} = -4,$$

preto súčasný bod nie je optimálne riešenie. Vypustíme z \mathbb{A}_p ohraničenie 4 a v ďalšom ho nebudeme uvažovať.

4. iterácia. Máme $\mathbf{x}^1 = (0, 3)^T$ a $\mathbb{A}_p = \emptyset$. Preto je \mathbb{P} jednotková matica \mathbb{I} . Kedže $\nabla_{\mathbf{x}} f(\mathbf{x}^1) = (-4, 0)^T$, tak

$$\Delta \mathbf{x} = - \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} -4 \\ 0 \end{pmatrix} = \begin{pmatrix} 4 \\ 0 \end{pmatrix}.$$

Neaktívne sú ohraničenia 1, 2, 3, 4 a 5.

$$\mathbf{a}_1 \cdot \Delta \mathbf{x} = (1 \ 1) \cdot \begin{pmatrix} 4 \\ 0 \end{pmatrix} = 4 \quad \text{a} \quad s_1 = -\frac{-3}{4} = \frac{3}{4}$$

$$\mathbf{a}_2 \cdot \Delta \mathbf{x} = (1 \ 2) \cdot \begin{pmatrix} 4 \\ 0 \end{pmatrix} = 4 \quad \text{a} \quad s_2 = -\frac{-4}{4} = 1$$

$$\mathbf{a}_3 \cdot \Delta \mathbf{x} = (1 \ 0) \cdot \begin{pmatrix} 4 \\ 0 \end{pmatrix} = 4 \quad \text{a} \quad s_3 = -\frac{-4}{4} = 1$$

$$\mathbf{a}_5 \cdot \Delta \mathbf{x} = (-1 \ 0) \cdot \begin{pmatrix} 4 \\ 0 \end{pmatrix} = -4 \quad \text{takže toto ohraničenie netreba uvažovať}$$

$$\mathbf{a}_5 \cdot \Delta \mathbf{x} = (0 \ -1) \cdot \begin{pmatrix} 4 \\ 0 \end{pmatrix} = 0 \quad \text{teda ani toto ohraničenie netreba uvažovať.}$$

Dostali sme $\bar{s} = \frac{3}{4}$, čo znamená, že potrebujeme vyriešiť úlohu

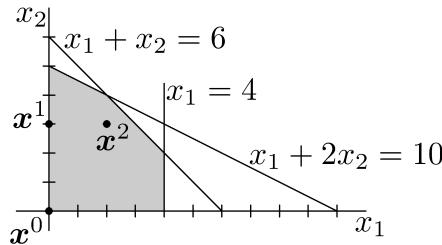
$$\begin{array}{c} \min f(s) = (4s - 2)^2 + (3 - 3)^2 \\ 0 \leq s \leq \frac{3}{4} \end{array}$$

Kedže $f'(s) = 2 \cdot (4s - 2) \cdot 4 = 0$ dá $s = \frac{1}{2}$ a $0 \leq \frac{1}{2} \leq \frac{3}{4}$, tak $s = \frac{1}{2}$. Preto

$$\mathbf{x}^2 = \begin{pmatrix} 0 \\ 3 \end{pmatrix} + \frac{1}{2} \begin{pmatrix} 4 \\ 0 \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \end{pmatrix}.$$

5. iterácia. Máme $\mathbf{x}^2 = (2, 3)^T$ a $\mathbb{A}_p = \emptyset$, lebo žiadne ohraničenie nie je aktívne. Teda $\mathbb{P} = \mathbb{I}$. Kedže $\nabla_{\mathbf{x}} f(\mathbf{x}^2) = (0, 0)^T$, dostávame $\Delta \mathbf{x} = (0, 0)^T$, takže potrebujeme určiť KKT podmienky. Kedže žiadne ohraničenie nie je aktívne, $\lambda_p = \emptyset$ a $\lambda = \mathbf{0}$. KKT podmienky sú splnené triviálne a algoritmus končí. Optimálne riešenie je $\mathbf{x}^2 = (2, 3)^T$.

Porovnajte ohraničenia predchádzajúceho príkladu s príkladom, ktorý sme riešili simplexovým algoritmom v Kapitole 1. Množina prípustných riešení spolu s bodmi \mathbf{x}^0 , \mathbf{x}^1 a \mathbf{x}^2 sú znázornené na Obrázku 25.



Obrázok 25

Nelineárne ohraničenia

Metóda projekcie gradientu bola zovšeobecnená na úlohy s nelineárnymi ohraničeniami. Budeme teda riešiť úlohu

$$\begin{array}{c} \min f(\mathbf{x}) \\ g_i(\mathbf{x}) \leq 0 \quad 1 \leq i \leq \ell \\ \mathbf{x} \in \mathbb{R}^n \end{array}$$

kde f, g_1, \dots, g_ℓ sú spojito diferencovateľné funkcie. Ak sú g_{i_1}, \dots, g_{i_r} aktívne ohraničenia, tak priestor dotykový k aktívnym ohraničeniam tvorí

$$\mathbb{A}_p = (\nabla g_{i_1}(\mathbf{x}), \dots, \nabla g_{i_r}(\mathbf{x}))^T.$$

Nasledujúce zovšeobecnenie navrhol Rosen a neskôr ho vylepšili Haug a Arora.

Zvolíme si, akú redukciu by sme chceli v ďalšom kroku dosiahnuť. Ak 5% , tak $\gamma = 0,05$, čo znamená, že očakávame $f(\mathbf{x}^k) - f(\mathbf{x}^{k+1}) \approx \gamma \cdot f(\mathbf{x}^k)$. Potom určíme

$$s^* = \frac{-\gamma \cdot f(\mathbf{x}^k)}{\Delta \mathbf{x}^k \cdot \nabla f(\mathbf{x}^k)},$$

kde $\Delta \mathbf{x}$ získame ako v lineárnej metóde projekcie gradientu, čiže $\Delta \mathbf{x} = -\mathbb{P} \cdot \nabla f(\mathbf{x}^k)$. Všimnime si, že s^* nezávisí od ohraničení, ale iba od účelovej funkcie. Preto sa ohraničenia môžu narušiť. Aby sa nenarušili príliš, volíme γ opatrne. Teraz položíme

$$\mathbf{x}^{k+1} = \mathbf{x}^k + s^* \cdot \Delta \mathbf{x}^k - \mathbb{A}_p^T \cdot (\mathbb{A}_p \cdot \mathbb{A}_p^T)^{-1} \cdot G_p(\mathbf{x}^k),$$

kde G_p je r -tica hodnôt funkcií g pre aktívne ohraničenia. Ak aktívne ohraničenia v \mathbf{x}^k nie sú narušené, posledný (opravný) člen vynechávame, pretože by vyšiel nulový.

Postup si ukážeme na dvoch príkladoch

PRÍKLAD 2. Riešte úlohu

$$\begin{array}{ll} \min & x_1^2 + x_2^2 + x_3^2 + x_4^2 - 2x_1 - 3x_4 \\ \text{---} & -2x_1 - x_2 - x_3 - 4x_4 + 7 \leq 0 \\ & -x_1 - x_2 - x_3^2 - x_4 + 5,1 \leq 0 \\ \text{---} & -x_1 \leq 0 \quad -x_2 \leq 0 \quad -x_3 \leq 0 \quad -x_4 \leq 0 \\ & x_1, x_2, x_3, x_4 \in \mathbb{R} \end{array}$$

Začnite v bode $\mathbf{x}^0 = (2, 2, 1, 0)^T$ a zlepšite hodnotu riešenia o 10% .

RIEŠENIE. Máme $f(\mathbf{x}^0) = 5$. V úlohe je šesť ohraničení. Pre \mathbf{x}^0 sú splnené ohraničenia 1, 2 a 6, avšak nelineárne ohraničenie 2 je trochu narušené. Keďže $\mathbb{A}_p = (\nabla g_1(\mathbf{x}^0), \nabla g_2(\mathbf{x}^0), \nabla g_6(\mathbf{x}^0))^T$, tak

$$\begin{aligned} \mathbb{A}_p &= \begin{pmatrix} -2 & -1 & -1 & -4 \\ -1 & -1 & -2 & -1 \\ 0 & 0 & 0 & -1 \end{pmatrix} & \mathbb{A}_p \cdot \mathbb{A}_p^T &= \begin{pmatrix} 22 & 9 & 4 \\ 9 & 7 & 1 \\ 4 & 1 & 1 \end{pmatrix} \\ (\mathbb{A}_p \cdot \mathbb{A}_p^T)^{-1} &= \frac{1}{11} \begin{pmatrix} 6 & -5 & -19 \\ -5 & 6 & 14 \\ -19 & 14 & 73 \end{pmatrix}. \end{aligned}$$

Ďalej $\mathbb{P} = \mathbb{I} - \mathbb{A}_p^T \cdot (\mathbb{A}_p \cdot \mathbb{A}_p^T)^{-1} \cdot \mathbb{A}_p$, čo dá

$$\mathbb{P} = \frac{1}{11} \begin{pmatrix} 1 & -3 & 1 & 0 \\ -3 & 9 & -3 & 0 \\ 1 & -3 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad \text{a} \quad \nabla f(\mathbf{x}^0) = \begin{pmatrix} 2 \\ 4 \\ 2 \\ -3 \end{pmatrix}.$$

Z toho dostávame $\Delta \mathbf{x} = -\mathbb{P} \cdot \nabla f(\mathbf{x}^0) = \left(\frac{8}{11}, -\frac{24}{11}, \frac{8}{11}, 0 \right)^T$, no keďže na veľkosti $\Delta \mathbf{x}$ nezáleží, pre jednoduchosť si zvolíme $\Delta \mathbf{x} = (1, -3, 1, 0)^T$. Ďalej určíme s^*

$$s^* = \frac{-0,1 \cdot f(\mathbf{x}^0)}{\Delta \mathbf{x}^T \cdot \nabla f(\mathbf{x}^0)} = \frac{-0,5}{-8} = 0,0625$$

(všimnime si, že $\Delta\mathbf{x}$ je v menovateli pre s^* , čiže na škálovaní $\Delta\mathbf{x}$ ozaj nezáleží). Pre korekciu máme $G_p(\mathbf{x}^0) = (0; 0,1; 0)^T$, preto je korekcia

$$-\mathbb{A}_p^T \cdot (\mathbb{A}_p \cdot \mathbb{A}_p^T)^{-1} \cdot G_p(\mathbf{x}^0) = \frac{-1}{110} \begin{pmatrix} 4 \\ -1 \\ -7 \\ 0 \end{pmatrix}.$$

Dostávame

$$\mathbf{x}^1 = \begin{pmatrix} 2 \\ 2 \\ 1 \\ 0 \end{pmatrix} + 0,0625 \begin{pmatrix} 1 \\ -3 \\ 1 \\ 0 \end{pmatrix} - \frac{1}{110} \begin{pmatrix} 4 \\ -1 \\ -7 \\ 0 \end{pmatrix} = \begin{pmatrix} 2,0261 \\ 1,8216 \\ 1,1261 \\ 0 \end{pmatrix}.$$

Máme $f(\mathbf{x}^1) = 4,6392$, $g_1(\mathbf{x}^1) = 0$ a $g_2(\mathbf{x}^1) = -0,0158$. V bode \mathbf{x}^1 sú teda splnené všetky ohraničenia. Kvôli nelinearite f a korekcii sme nedostali zlepšenie o 10 %, ale iba o čosi viac, ako 7 %.

PRÍKLAD 3. Riešte úlohu

$$\begin{array}{ll} \min & 3x_1 + \sqrt{3}x_2 \\ \hline & \frac{18}{x_1} + \frac{6\sqrt{3}}{x_2} - 3 \leq 0 \\ & 5,75 - x_1 \leq 0 \\ & 7,17 - x_2 \leq 0 \\ \hline & x_1, x_2 \in \mathbb{R} \end{array}$$

začnite v bode \mathbf{x}^0 , pre ktorý $g_1(\mathbf{x}^0) = g_3(\mathbf{x}^0) = 0$.

RIEŠENIE. Budeme postupovať podľa vyššie opísaného algoritmu.

1. iterácia. Z $g_1(\mathbf{x}^0) = g_3(\mathbf{x}^0) = 0$ dostávame $\mathbf{x}^0 = (11,6085; 7,17)^T$ s hodnotou účelovej funkcie $f(\mathbf{x}^0) = 47,2443$. Kedže

$$\nabla g_1(\mathbf{x}^0) = \begin{pmatrix} -0,1336 \\ -0,2021 \end{pmatrix} \quad \text{a} \quad \nabla g_3(\mathbf{x}^0) = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$$

tak

$$\mathbb{A}_p = \begin{pmatrix} -0,1336 & -0,2021 \\ 0 & -1 \end{pmatrix}.$$

Matica \mathbb{A}_p je regulárna, čo dá $\mathbb{P} = \emptyset$ a následne $\Delta\mathbf{x} = \mathbf{0}$. Riešenie sa teda nedá zlepšiť. Potrebujeme určiť KKT podmienky, čiže λ_p . Kedže $\nabla_{\mathbf{x}} f(\mathbf{x}^0) = (3, \sqrt{3})^T$, tak

$$\lambda_p = -(\mathbb{A}_p \cdot \mathbb{A}_p^T)^{-1} \cdot \mathbb{A}_p \cdot \nabla f(\mathbf{x}^0) = \begin{pmatrix} 22,47 \\ -2,80 \end{pmatrix}.$$

Preto vypustíme ohraničenie 3 a v ďalšom ho nebudeme uvažovať.

2. iterácia. Teraz máme $\mathbb{A}_p = (-0,1336 \ -0,2021)^T$, čo dá

$$\mathbb{P} = \begin{pmatrix} 0,6959 & -0,4600 \\ -0,4600 & 0,3041 \end{pmatrix} \quad \text{a} \quad \Delta\mathbf{x} = -\mathbb{P} \cdot \nabla f(\mathbf{x}^0) = \begin{pmatrix} -1,2910 \\ 0,8533 \end{pmatrix}.$$

Pokúsime sa zredukovať $f(\mathbf{x}^0)$ o 5 %. To dá

$$s^* = \frac{-0,05 \cdot 47,2443}{(-1,2910; 0,8533) \cdot (3, \sqrt{3})^T} = 0,9863.$$

Kedže žiadne ohraničenie nie je narušené, opravný člen vyniecháme (vyšiel by nulový). Dostávame

$$\mathbf{x}^1 = \mathbf{x}^0 + s^* \cdot \Delta \mathbf{x} = \begin{pmatrix} 11,6085 \\ 7,17 \end{pmatrix} + 0,9863 \cdot \begin{pmatrix} -1,2910 \\ 0,8533 \end{pmatrix} = \begin{pmatrix} 10,3352 \\ 8,0116 \end{pmatrix}.$$

V \mathbf{x}^1 ohraničenie 2 nie je narušené. Keby bolo, vrátili by sme sa na určenie s^* a ohraničili by sme ho hodnotou $-g_2(\mathbf{x}^0)/(\nabla g_2(\mathbf{x}^0)^T \cdot \Delta \mathbf{x})$ analogicky ako v prípade lineárnych ohraničení.

3. iterácia. V \mathbf{x}^1 je $f(\mathbf{x}^1) = 44,8821$ a $g_1(\mathbf{x}^1) = 0,0388$. Opäť je teda aktívne ohraničenie 1, ktoré je trochu narušené. Máme $\mathbb{A}_p = (-0,1685; -0,1619)$, čo dá

$$\mathbb{P} = \begin{pmatrix} 0,4800 & -0,4996 \\ -0,4996 & 0,5200 \end{pmatrix} \quad \text{a} \quad \Delta \mathbf{x} = -\mathbb{P} \cdot \nabla f(\mathbf{x}^1) = \begin{pmatrix} -0,5747 \\ 0,5981 \end{pmatrix}.$$

Pokúsime sa zredukovať $f(\mathbf{x}^1)$ o 2,5 %. To dá

$$s^* = \frac{-0,025 \cdot 44,8821}{(-0,5747; 0,5981) \cdot (3, \sqrt{3})^T} = 1,6305.$$

Teraz potrebujeme korekciu kvôli narušeniu g_1 .

$$-\mathbb{A}_p \cdot (\mathbb{A}_p \cdot \mathbb{A}_p^T)^{-1} \cdot G_p(\mathbf{x}^1) = \begin{pmatrix} 0,1179 \\ 0,1133 \end{pmatrix}.$$

Dostávame

$$\begin{aligned} \mathbf{x}^2 &= \mathbf{x}^1 + s^* \cdot \Delta \mathbf{x} - \mathbb{A}_p^T \cdot (\mathbb{A}_p \cdot \mathbb{A}_p^T)^{-1} \cdot G_p(\mathbf{x}^1) = \\ &= \begin{pmatrix} 10,3352 \\ 8,0116 \end{pmatrix} + 1,6305 \begin{pmatrix} -0,5747 \\ 0,5981 \end{pmatrix} + \begin{pmatrix} 0,1179 \\ 0,1133 \end{pmatrix} = \begin{pmatrix} 9,5161 \\ 9,1028 \end{pmatrix}. \end{aligned}$$

Tu $f(\mathbf{x}^2) = 44,3148$ a $g_1(\mathbf{x}^2) = 0,0332$, čiže ohraničenie 1 je opäť trochu narušené.

Pre optimálne riešenie predchádzajúceho príkladu platí $\mathbf{x}^* = (9,4641; 9,4641)^T$ a $f(\mathbf{x}^*) = 44,7846$. Všimnime si, že $f(\mathbf{x}^*) > f(\mathbf{x}^2)$, čo je spôsobené narušením g_1 v \mathbf{x}^2 .

Pre záujemcov o hlbšie štúdium teórie a algoritmov matematického programovania odporúčame špecializované monografie [2,7,9].

Cvičenia

CVIČENIE 9.1. Metódou projekcie gradientu vyriešte úlohu

$$\begin{array}{ll} \min f(\boldsymbol{x}) = e^{(x_1 - 3)^2} + (x_1 + x_2 - 6)^2 \\ \hline x_1 + x_2 \leq 5 \\ x_1 \leq 4 \\ -x_1 + x_2 \leq 2 \end{array}$$

1. Zakreslite množinu prípustných riešení a optimum úlohy bez ohraničení.
2. Naprogramujte algoritmus a spusťte ho z rôznych bodov \boldsymbol{x}^0 . Zakaždým do obrázku zakreslite jednotlivé iterácie.

CVIČENIE 9.2. V Príklade 3 sme dostali nerovnicu $f(\boldsymbol{x}^*) > f(\boldsymbol{x}^2)$. Ako sa zmení \boldsymbol{x}^2 ak v tretej iterácii zredukujeme $f(\boldsymbol{x}^1)$ o menej ako 2,5 %?

CVIČENIE 9.3. Spravte v Príklade 3 ďalšiu iteráciu, pričom sa pokúsíte zlepšiť $f(\boldsymbol{x}^2)$ o veľmi malú hodnotu.

CVIČENIE 9.4. Naprogramujte algoritmus riešiaci úlohu z Príkladu 3. Algoritmus spravte tak, aby vykresľoval jednotlivé iterácie, pričom po každej iterácii sa vypíše či (a ako) sú jednotlivé ohraničenia narušené.

Pre prípad, že by boli ohraničenia narušené príliš, implementujte možnosť vrátiť sa krok späť.

10 Stochasticke optimalizačné metódy

Budeme riešiť úlohu bez ohraničení

$$\frac{\min f(\mathbf{x})}{\mathbf{x} \in D_f}$$

kde $f(\mathbf{x})$ môže byť veľmi komplikovaná funkcia. Budeme sa snažiť nájsť buď globálne minimum, alebo jeho dobrú approximáciu. Metódy sa dajú rozdeliť do dvoch skupín.

- *Deterministické (presné) metódy.* Tieto sa dajú aplikovať, ak je priestor riešení malý, respektíve keď je f pomerne jednoduchá funkcia (napríklad diferencovateľná).
- *Stochasticke metódy.* Tieto metódy nájdu iba approximáciu globálneho minima a práve nimi sa budeme v posledných dvoch kapitolách zaoberať.

Medzi základné stochasticke metódy patria:

- *Metóda Monte Carlo,*
- *Horolezecký (hill-climbing) algoritmus,*
- *Simulované žihanie,*
- *Evolučné algoritmy.*

Metóda Monte Carlo

V základnej forme tátó metóda generuje náhodne prvky $\mathbf{x} \in D_f$ a určuje $f(\mathbf{x})$. Ten prvak \mathbf{x} , pre ktorý je po istom počte opakovania $f(\mathbf{x})$ najmenšie, prehlásime za riešenie.

Aby tátó metóda fungovala, potrebujeme dobrý generátor náhodných (presnejšie pseudonáhodných) čísel. V minulosti sa používali tabuľky náhodných čísel, v súčasnosti sa počítačom generujú pseudonáhodné čísla. Generátory musia prejsť viacerými testami, pričom sa od nich požaduje:

- a) *Dlhá períoda.* To znamená, že sa postupnosti generovaných čísel nesmú opakovať príliš skoro.
- b) *Efektívnosť.* Výpočet musí byť rýchly.
- c) *Opakovateľnosť.* Generátory sú deterministické, takže pri rovnakých vstupných podmienkach musia vygenerovať rovnaké postupnosti čísel.
- d) *Prenosnosť.* Vyžadujeme, aby bola vygenerovaná postupnosť rovnaká na rôznych počítačoch.
- e) *Nevypočítateľnosť.* Požadujeme, aby zo znalosti niekoľkých posledných, povedzme k vygenerovaných čísel nebolo možné určiť ďalšie.
- f) *Úspešnosť v empirických testoch.*

Na ukážku si uvedieme tri generátory pseudonáhodných čísel.

Metóda stredu mocniny

Začneme so štvorciferným číslom. Zostrojíme jeho druhú mocninu a cifry na mieste 10^2 až 10^5 budú tvoriť novovygenerované číslo. Napríklad

$$\begin{aligned}x_0 &= 8219 : \quad 8219^2 = 67\overline{5519}61 \\x_1 &= 5519 : \quad 5519^2 = 30\overline{4593}61 \\x_2 &= 4593 : \quad 4593^2 = 21\overline{0956}49 \quad \dots\end{aligned}$$

Tento generátor niekedy generuje postupnosti s veľmi krátkou periódou. Avšak keď sa tak stane, obyčajne sa dá problém rozpoznať rýchlo. Napríklad

$$x_0 = 3792 : \quad 3792^2 = 14\overline{3792}64$$

v tomto prípade získame postupnosť s periódou dĺžky 1. Respektíve

$$\begin{aligned}x_0 &= 6100 : \quad 6100^2 = 37\overline{2100}00 \\x_1 &= 2100 : \quad 2100^2 = 04\overline{4100}00 \\x_2 &= 4100 : \quad 4100^2 = 16\overline{8100}00 \\x_3 &= 8100 : \quad 8100^2 = 65\overline{6100}00\end{aligned}$$

a získame postupnosť s periódou dĺžky 4. V súčasnosti sa tento generátor už ne-používa, pretože nespĺňa podmienku nevypočítateľnosti.

Lineárny kongruenčný generátor

Vylepšené verzie tohto generátora sa používajú dodnes. Na začiatku vyberieme štyri čísla

$$\begin{aligned}k &: \text{činitel}, \\c &: \text{zdvih}, \\m &: \text{modul}, \\x_0 &: \text{štartovacie číslo}.\end{aligned}$$

Teraz pre i -te generované číslo, kde $i \geq 1$, platí

$$x_i = k \cdot x_{i-1} + c \pmod{m}.$$

Tento generátor generuje pseudonáhodné čísla s periódou $p \leq m$. Vhodné voľby sú $k = 5^{10}$, $c = 0$ a $m = 2^{40}$.

Blum Blum Shub generátor

Vyberú sa tri čísla

$$\begin{aligned}p, q &: \text{veľké prvočísla}, \\\bar{x}_0 &: \text{štartovacie číslo}.\end{aligned}$$

Teraz

$$\bar{x}_i = \bar{x}_{i-1}^2 \pmod{p \cdot q},$$

pričom za x_i sa vyberie iba malá informácia z \bar{x}_i , povedzme parita počtu jednotiek v binárnom zápise. Tento generátor je pomalý, na počítačové simulácie je nevhodný, využíva sa však v kryptografii, pretože je teoreticky dokázaná jeho vysoká bezpečnosť (nevypočítateľnosť).

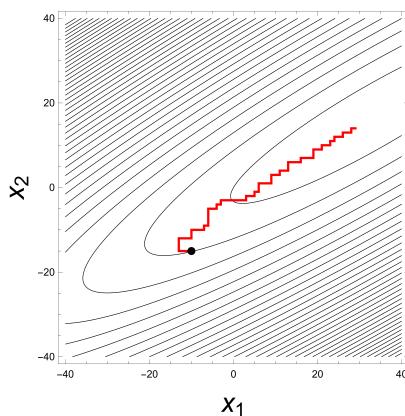
Medzi ďalšie generátory patria napríklad Multiply with carry, či Mersenne twister. V súčasnosti patrí práve Mersenne twister (a jeho varianty) medzi najpoužívanejšie, ide však o pomerne zložitý generátor.

Horolezecký algoritmus

Názov algoritmu vznikol pre maximalizačnú úlohu, avšak my riešime (ako bolo uvedené v úvode kapitoly) úlohu minimalizačnú. Horolezecký algoritmus je založený na prehľadávaní blízkych prípustných riešení. Zvolíme si úvodný bod $\mathbf{x}^0 \in D_f$. Teraz začneme prezerať blízke prípustné riešenia, a keď natrafíme na taký bod \mathbf{x} , v ktorom $f(\mathbf{x}) < f(\mathbf{x}^{i-1})$, tak položíme $\mathbf{x}^i = \mathbf{x}$.

Algoritmus môžeme modifikovať tak, že najprv prezrieme všetky prípustné riešenia blízke bodu \mathbf{x}^{i-1} a to, v ktorom nadobúda $f(\mathbf{x})$ minimum, prehlásime za \mathbf{x}^i .

Problémom Horolezeckého algoritmu sú „planiny“, čiže funkcie, v ktorých je len veľmi malý (až žiadnen) rozdiel medzi $f(\mathbf{x}^{i-1})$ a $f(\mathbf{x})$, kde \mathbf{x} je bod blízky \mathbf{x}^{i-1} . Druhý problém predstavujú „šikmé doliny“.



Obrázok 26

Na Obrázku 26 máme znázornený postup Horolezeckého algoritmu v „šikmej dolinke“. K \mathbf{x}^{i-1} sú blízke prípustné riešenia $\mathbf{x}^{i-1} \pm \mathbf{e}_j$, $1 \leq j \leq n$, kde \mathbf{e}_j je jednotkový vektor s jednotkou na j -tej pozícii, preto ide algoritmus cik-cakovito. Ak však bude dolinka „veľmi úzka“ a ak budú jej svahy „veľmi strmé“, tak sa môže stať, že vo všetkých bodoch blízkych k \mathbf{x}^{i-1} bude hodnota účelovej funkcie vyššia ako v samotnom bode \mathbf{x}^{i-1} , hoci v istom smere bude dolinka klesať.

V základnej forme sú Metóda Monte Carlo a Horolezecký algoritmus úplne

opačné metódy, pretože zatiaľ čo Metóda Monte Carlo vyberá „rovnomerne“ rozmiestnené body prípustnej množiny ale nehľadá lokálny extrém, Horolezecký algoritmus nájde lokálny extrém blízky úvodnému prípustnému riešeniu, avšak ignoruje zvyšok prípustnej množiny riešení. Preto sa zvyčajne tieto algoritmy kombinujú. Čiže náhodne vygenerujeme \mathbf{z}^i Metódou Monte Carlo a Horolezeckým algoritmom nájdeme zo \mathbf{z}^i lokálny extrém \mathbf{x}^i . Potom náhodne vygenerujeme \mathbf{z}^{i+1} a proces opakujeme.

Metóda Monte Carlo a Horolezecký algoritmus často predstavujú krajné medze pre ostatné algoritmy v tom zmysle, že pri nevhodne zvolených parametroch (napríklad aj v simulovanom žíhaní, pozri nižšie) môžeme dostať z týchto algoritmov práve Metódu Monte Carlo, či Horolezecký algoritmus.

Simulované žíhanie

Simulované žíhanie (Simulated annealing) sa podobá na Horolezecký algoritmus. Rozdiel je v tom, že s istou pravdepodobnosťou, ktorú postupne znižujeme, prijmemme aj riešenie horšie od súčasného. Algoritmus bol motivovaný postupným chladením taveniny, pri ktorom sa vďaka občasnemu miernemu zahriatiu dosahuje štruktúra bez defektov. Práve z toho dôvodu sa pri opise algoritmu používajú pojmy ako „teplota“, avšak my si pod teplotou jednoducho predstavíme parameter T a opíšeme, akým spôsobom budeme tento parameter meniť. Podobne to platí pre „plán chladenia“ a podobne. V najzákladnejšej forme má algoritmus nasledujúci tvar.

ALGORITMUS: Simulované žíhanie.

Vstup: počet opakovania k , plán chladenia, \mathbf{x}^0 , T_0 , T_{\min} .

Begin

$\mathbf{x} = \mathbf{x}^0; T = T_0;$ { inicializácia }

Repeat

For $i = 1$ To k Do Begin

Vygeneruj nové prípustné riešenie \mathbf{x}_n ;

Ak $P(f(\mathbf{x}), f(\mathbf{x}_n), T) \geq \text{Random}(0, 1)$ Tak $\mathbf{x} = \mathbf{x}_n$;

End;

Zniž T podľa plánu chladenia;

Until $T < T_{\min}$;

End.

Tu $P(f(\mathbf{x}), f(\mathbf{x}_n), T)$ je pravdepodobnosť prijatia nového prípustného riešenia \mathbf{x}_n (s hodnotou účelovej funkcie $f(\mathbf{x}_n)$) pri súčasnom riešení \mathbf{x} (s hodnotou účelovej funkcie $f(\mathbf{x})$) a teplote T a $\text{Random}(0, 1)$ dáva náhodnú hodnotu z intervalu $(0, 1)$.

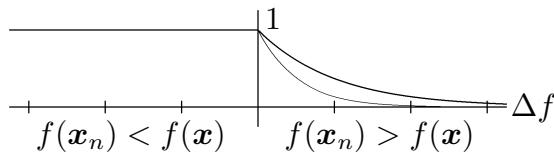
Algoritmus má viacero modifikácií. Opíšeme takzvaný **Metropolisov algoritmus**, ktorý je založený na **Boltzmannovskom rozdelení pravdepodobnosti**. Pri tomto rozdelení je pravdepodobnosť, že sa systém pri teplote T nachádza v prípustnom riešení \mathbf{x} , daná

$$w_T(\mathbf{x}) = \frac{1}{Q(T)} \cdot e^{-\frac{f(\mathbf{x})}{k_B \cdot T}}$$

kde k_B je konštantá (takzvaná Boltzmannova konštantá) a $Q(T)$ je normalizačný faktor, $Q(T) = \sum_{\mathbf{x}} e^{-\frac{f(\mathbf{x})}{k_B \cdot T}}$, pričom suma ide cez všetky možné prípustné riešenia. Určiť hodnotu $Q(T)$ je náročné, no v ďalšom ju nebudem potrebovať, pretože pravdepodobnosť prijatia nového riešenia bude podiel pravdepodobností, takže $Q(T)$ vypadne. Potom

$$P(f(\mathbf{x}), f(\mathbf{x}_n), T) = \min \left\{ 1, e^{-\frac{f(\mathbf{x}_n) - f(\mathbf{x})}{k_B \cdot T}} \right\}.$$

To znamená, že ak $f(\mathbf{x}_n) \leq f(\mathbf{x})$, tak pravdepodobnosť prijatia \mathbf{x}_n je 1, teda ak je nové prípustné riešenie lepšie ako staré, tak ho prijmeme vždy. Avšak s istou pravdepodobnosťou prijmeme aj horšie prípustné riešenie. Na Obrázku 27 sú dve krivky pre $e^{-\frac{f(\mathbf{x}_n) - f(\mathbf{x})}{k_B \cdot T}}$, pričom tenkou čiarou je zobrazená krivka pre nižšie T , teda pre neskoršie dosahovanú teplotu. (Hrubšou čiarou je v Obrázku 27 graf pre $k_B \cdot T = 1$, tenšou pre $k_B \cdot T = 0,5$.) Na vertikálnej osi sme vyniesli pravdepodobnosť a na horizontálnej osi máme rozdiel $\Delta f = f(\mathbf{x}_n) - f(\mathbf{x})$.



Obrázok 27

Kedže pre nás je T iba parameter, nie skutočná teplota, tak všade namiesto výrazu $k_B \cdot T$ budeme písat iba T . Potom úvodnú teplotu (čiže úvodný parameter) T_0 volíme tak, aby sa akceptovala približne polovica horších prípustných riešení.

Ešte potrebujeme navrhnúť plán chladenia, čiže spôsob, ako budeme meniť hodnotu parametra T . S Boltzmannovským simulovaným žíhaním je konzistentný logaritmický plán chladenia, čiže

$$T_\ell = \frac{T_0}{\ln(\ell)},$$

kde ℓ je časový index chladenia. Väčšinou však kvôli jednoduchosti volíme exponentiálny plán chladenia

$$T_\ell = \alpha^\ell \cdot T_0 \quad \text{čiže} \quad T_\ell = \alpha \cdot T_{\ell-1},$$

pričom $0 \ll \alpha < 1$. V takom prípade hovoríme o **Simulovanom kalení (Simulated quenching)**. Vhodné voľby pre α ležia v intervale $\langle 0,8; 0,99 \rangle$, čiže je vhodné pomalé chladenie. Pri veľmi rýchлом chladení by sa totiž algoritmus dostal do lokálneho minima a tam by skončil, čiže by sa zmenil na klasický Horolezecký algoritmus.

PRÍKLAD. Majme vo štvorci $\langle 0, 1 \rangle \times \langle 0, 1 \rangle$ zvolených n bodov. Nájdite najkratšiu otvorenú cestu (lomenú čiaru), ktorá prechádza cez všetky tieto body.

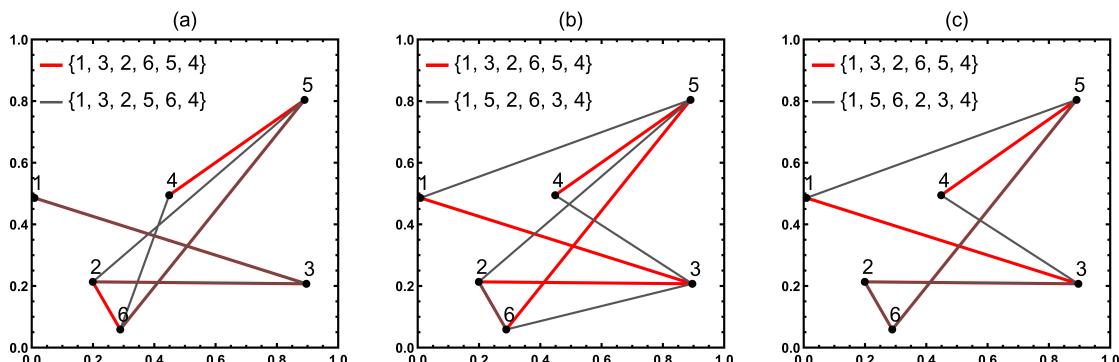
RIEŠENIE. Táto úloha patrí medzi náročné. Dopolnil nepoznáme polynomiálny algoritmus na nájdenie najkratšej otvorenej cesty, preto použijeme stochastickú metódu Simulované žíhanie.

Pomocou lomenej čiary dané body usporiadame. Bod, v ktorom čiara začína, bude prvý, nasledujúci bod na čiare bude druhý atď. Takže prípustné riešenia \mathbf{x} budú permutácie n prvkov a účelovou funkciou $f(\mathbf{x})$ bude dĺžka lomenej čiary určenej permutáciou \mathbf{x} . Máme teda $n!$ prípustných riešení. V skutočnosti je prípustných riešení iba $n!/2$, pretože jedna lomená čiara zodpovedá dvom permutáciám (pri druhej body „čítame“ od posledného k prvému). V každom prípade je pre väčšie n počet prípustných riešení privyleký, preto nemôžeme prejsť celý priestor prípustných riešení. Použijeme Simulované žíhanie.

Algoritmus sme už opísali, zostáva určiť, ako z \mathbf{x} vygenerujeme „blízke“ nové prípustné riešenie \mathbf{x}_n . Môžeme použiť jeden z nasledujúcich generátorov susedného prípustného riešenia.

- a) *Výmena ľubovoľných dvoch susedných bodov v permutácii.* Napríklad z danej permutácie $(1, 3, 2, 6, 5, 4)$, v ktorej za bodom 1 nasleduje 3, potom 2, atď., zamenením 6 za 5 dostaneme permutáciu $(1, 3, 2, 5, 6, 4)$.
- b) *Výmena ľubovoľných dvoch bodov v permutácii.* Napríklad z permutácie $(1, 3, 2, 6, 5, 4)$ zamenením 3 za 5 dostaneme permutáciu $(1, 5, 2, 6, 3, 4)$.
- c) *Preklopenie úseku permutácie medzi ľubovoľnými dvoma bodmi.* Napríklad z permutácie $(1, 3, 2, 6, 5, 4)$ preklopením úseku od bodu 3 po 5 dostaneme permutáciu $(1, 5, 6, 2, 3, 4)$.

Na Obrázku 28 máme tieto generátory zobrazené. Úvodná permutácia je znázorená červenou čiarou, výsledná sivou.



Obrázok 28

Všimnime si, že pri operáciach (a) a (c) zameníme dve úsečky v lomenej čiare za dve iné, zatiaľ čo pri (b) zameníme vo všeobecnosti až štyri úsečky za štyri iné. Pri týchto operáciach je dôležité, aby sme sa z ľubovoľnej permutácie do ľubovoľnej inej vedeli dostať na „malý počet krokov“, čiže aby mal graf zodpovedajúci operácií malý priemer. Ak označíme príslušné grafy G_a , G_b a G_c , tak pre ich počet vrcholov platí $|V(G_a)| = |V(G_b)| = |V(G_c)| = n!/2$ a

$$\begin{aligned} \frac{n^2}{4} &\sim \frac{n-2}{2} \cdot \frac{n}{2} \leq diam(G_a) \leq \binom{n}{2} \sim \frac{n^2}{2} \\ diam(G_b) &\leq n-1 \sim n \\ diam(G_c) &\leq n-1 \sim n \end{aligned}$$

pričom dolnú hranicu pre $diam(G_a)$ sme určili pomocou vzdialenosť permutácie

$1, 3, 5, \dots, 2, 4, 6, \dots$ od $1, 2, \dots, n$. Vidíme, že operácie (b) a (c) by mali byť vhodnejšie. Odporuča sa pre vyššie teploty (spočiatku) voliť radšej (b), pre nižšie teploty radšej (c).

Cvičenia

CVIČENIE 10.1. Nájdenie najkratšej otvorenej cesty spájajúcej n bodov vo štvorcí $\langle 0, 1 \rangle \times \langle 0, 1 \rangle$.

1. Vykreslite štvorec $\langle 0, 1 \rangle \times \langle 0, 1 \rangle$.
2. Vygenerujte a vykreslite n náhodne zvolených bodov vo štvorci z časti 1.
3. Vygenerujte náhodnú permutáciu \mathbf{x}^0 čísel $1, 2, \dots, n$ (teda náhodne vyberte prvé číslo z $1, 2, \dots, n$, potom zo zvyšných $n-1$ náhodne vyberte druhé atď.) a vykreslite cestu zodpovedajúcu tejto permutácii do obrázka.
4. Naprogramujte Stochastický horolezecký algoritmus (v štandardnom Horolezeckom algoritme prehľadáme všetkých susedov \mathbf{x} , tu skúsime k_{max} náhodných krokov a posunieme sa vtedy, keď sa zlepší hodnota účelovej funkcie.)

Vstup: \mathbf{x}^0 , k_{max} .

Begin

```

 $\mathbf{x} = \mathbf{x}^0;$ 
For  $k = 1$  To  $k_{max}$  Do Begin
    Generuj nový stav  $\mathbf{x}_n = neighbour[\mathbf{x}]$ ;
    If  $f(\mathbf{x}_n) < f(\mathbf{x})$  Then  $\mathbf{x} = \mathbf{x}_n$ ;
End;
End.
```

Pre stochastický operátor $neighbour[\mathbf{x}]$ použite výmenu dvoch susedných bodov (čiže operátor (a)).

5. Pre rýchly výpočet je dobré na začiatku vypočítať vzdialenosť medzi všetkými dvojicami bodov a uložiť ich do matice so zložkami $L_{i,j}$ = „vzdialenosť medzi bodmi i a j “. Možno na to použiť funkciu `DistanceMatrix`. Dĺžka cesty $f(\mathbf{x})$ sa potom určí spočítaním správnych prvkov matice L .

6. Pre malé n (pre $n < 10$) nájdite hrubou silou globálne minimum a zistite, či ho Horolezecký algoritmus našiel.

7. Naprogramujte simulované žihanie.

- a) Definujte funkciu $P(f(\mathbf{x}), f(\mathbf{x}_n), T) = P(\Delta f, T) = \min\{1, e^{-\frac{\Delta f}{T}}\}$ (za Δf sa bude v algoritme dosádzovať $f(\mathbf{x}_n) - f(\mathbf{x})$). Vykreslite túto funkciu ako funkciu premennej Δf a parametra T pomocou procedúry `Manipulate` cez T .
 - b) V Horolezeckom algoritme zmeňte kritérium pre prijatie nového stavu a doplňte chladenie.
 - c) Doplňte zvyšné dva stochastické operátory (čiže operátory (b) a (c)).
8. Vizualizujte kroky algoritmu pomocou `Manipulate` cez všetky iterácie. V každej iterácii bude
- a) Štvorec $\langle 0, 1 \rangle \times \langle 0, 1 \rangle$ s vykreslenými bodmi.

- b) Zakreslená aktuálna cesta \boldsymbol{x} .
 - c) Výpis hodnoty $f(\boldsymbol{x})$ a aktuálnej teploty T .
9. Vymyslite algoritmus na hľadanie vhodného T_0 .
10. Pre ktorý stochastický operátor funguje algoritmus najlepšie?

CVIČENIE 10.2. Pomocou simulovaného žíhania nájdite najkratšiu otvorenú cestu spájajúcu 10 najväčších slovenských miest.

1. Vyberte si jeden z dvojice (predpripravených) súborov. V jednom sa vzdialosti medzi mestami uvádzajú ako geodetické vzdialnosti, v druhom ide o vzdialosti po cestách.

2. Doplňte chýbajúce časti (stačí ich skopírovať z Cvičenia 10.1).

- a) Stochastické generátory poruchy.
- b) Definíciu funkcie $P(f(\boldsymbol{x}), f(\boldsymbol{x}_n), T) = P(\Delta f, T)$.
- c) Algoritmus na hľadanie vhodného T_0 .
- d) Algoritmus simulovaného žíhania.

11 Evolučné algoritmy

Veľká skupina stochastických metód (ako napríklad genetické programovanie, evolučná stratégia, neuroevolúcia a iné) je založená na evolučných procesoch. Tie-to metódy (s veľmi podobnými názvami) nenájdú optimálne riešenie, iba jeho aproximáciu v reálnom čase. Do tejto skupiny patria **genetické algoritmy**, ktorými sa budeme zaoberať v tejto kapitole.

Schéma genetického algoritmu

Opíšeme jednu iteráciu genetického algoritmu.

1. Po i -tej iterácii budeme mať množinu prípustných riešení $\mathbf{x}_1, \dots, \mathbf{x}_g$. Túto množinu nazývame **i -ta generácia**.
2. Vyberieme z i -tej generácie tie riešenia, ktoré majú lepšiu hodnotu účelovej funkcie.
3. Teraz budeme riešenia vybranej skupiny krížiť (kombinovať) za účelom získania novej množiny riešení.
4. S istou nie veľkou pravdepodobnosťou budú nové riešenia mutovať (čiže niektoré mierne pozmeníme).
5. Dostávame novú skupinu riešení, $(i+1)$ -vú generáciu $\mathbf{x}'_1, \dots, \mathbf{x}'_g$.

Niekedy riešenie s najlepšou hodnotou účelovej funkcie automaticky presunieme do novej generácie. Tým dosiahneme, že v poslednej generácii bude najlepšie nájdené riešenie.

Kedže optimálne riešenie nepoznáme, nemáme rozumnú podmienku na ukončenie algoritmu. Preto ho obyčajne ukončíme po zostrojení predpísaného počtu generácií.

Najnáročnejšie býva navrhnutie takého kríženia, pri ktorom sa na potomkov presunú dobré vlastnosti rodičov. Z toho dôvodu sa často mutácia prijíma iba vtedy, keď zlepší hodnotu účelovej funkcie.

Na zostavenie genetického algoritmu potrebujeme navrhnúť:

- a) Zakódovanie riešenia.
- b) Vyhodnotenie riešenia a výber riešení na kríženie.
- c) Kríženie.
- d) Mutácie.
- e) Dekódovanie riešenia.

Zostavenie genetického algoritmu si opíšeme na príklade. Objasníme si, ako navrhnúť vyššie vymenované kroky, avšak krok e) vynecháme, pretože je inverzný k a).

Problém obchodného cestujúceho

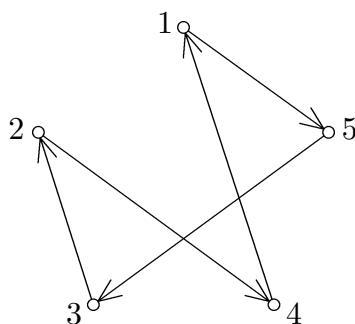
DEFINÍCIA. Majme n miest, pričom niektoré dvojice sú pospájané cestami daných dĺžok. **Problém obchodného cestujúceho** spočíva v zistrojení najkratšej okružnej prechádzky, počas ktorej sa navštívi každé mesto práve raz.

Ináč povedané, problém obchodného cestujúceho spočíva v nájdení najkratšej Hamiltonovskej kružnice v ohodnotenom grafe.

Tu sa budeme zaoberať modifikáciou problému obchodného cestujúceho, v ktorej budú mestá reprezentovať náhodne vybrané body štvorca $\langle 0, 1 \rangle \times \langle 0, 1 \rangle$. Budeme predpokladať, že každá dvojica je spojená cestou, ktorej dĺžka sa rovná klasickej Euklidovskej vzdialenosťi týchto bodov. Rozdiel oproti príkladu z minulej kapitoly je v tom, že zatiaľ čo tam sme hľadali najkratšiu otvorenú prechádzku (Hamiltonovskú cestu), teraz budeme hľadať najkratšiu uzavretú prechádzku (Hamiltonovskú kružnicu, pozri [11]).

Kódovanie riešenia

Ukážeme si štyri spôsoby kódovania Hamiltonovskej kružnice. Majme v rovine päť bodov, 1, 2, 3, 4 a 5, pričom kružnica bude obsahovať hrany $(2, 4)$, $(4, 1)$, $(1, 5)$, $(5, 3)$ a $(3, 2)$. Okrem toho budeme mať zvolený smer prechádzania tejto kružnice tak, že z 2 pôjdeme do 4, ďalej do 1, atď. Na Obrázku 29 máme týchto päť bodov rovnomerne rozmiestnených na kružnici a hrany so zvoleným smerom prechádzania máme znázornené šípkami. Vrcholy sú zakreslené malými krúžkami.



Obrázok 29

Kódovanie pomocou permutačnej matice

Táto matica obsahuje iba nuly a jedničky. Pritom v i -tom riadku a j -tom stĺpci je jednička práve vtedy, keď sa v Hamiltonovskej kružnici vyskytuje hrana (i, j) , ktorú prechádzame v smere od i k j . Znamená to, že táto matica typu $(n \times n)$ bude mať práve n jedničiek, pričom v každom riadku (stĺpcu) bude práve jedna. Kružnici

z nášho príkladu, čiže permutácií prvkov 1,2,3,4,5, zodpovedá permutačná matica

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}.$$

Kódovanie pomocou permutácie

Tu budeme uvažovať matice typu $(2 \times n)$. V hornom riadku budú zoradené všetky body štandardne, pričom v dolnom budú zapísaní susedia na orientovanej Hamiltonovskej kružnici. Znamená to, že každý stĺpec $\binom{i}{j}$ zodpovedá hrane (i, j) , ktorú prechádzame v smere od i k j . V našom príklade dostaneme maticu

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 4 & 2 & 1 & 3 \end{pmatrix}$$

čo skrátene zapíšeme

$$[5, 4, 2, 1, 3],$$

protože prvý riadok matice je vždy rovnaký.

Kódovanie pomocou cyklu

Tu zapíšeme za sebou body v tom poradí, v akom ich prechádzame. V našom príklade dostávame

$$(2, 4, 1, 5, 3).$$

Keď sa chceme vyhnúť nejednoznačnosti, môžeme si prvý bod zvoliť. Obyčajne si volíme najmenší prvak, čiže 1. Potom v našom príklade dostávame zápis

$$(1, 5, 3, 2, 4).$$

Ordinálna reprezentácia

Pri tejto reprezentácii si prvý bod zvolíme fixne. Nech je ním 1. Teraz zapíšeme napravo všetky body v štandardnom poradí a naľavo bude prázdny reťazec $\langle \rangle$. Postupne budeme vyberať z pravej strany body podľa toho, ako ich prechádzame na zorientovanej Hamiltonovskej kružnici, a naľavo zapíšeme ich momentálne (relatívne) poradie. V našom príklade dostávame

$\langle \rangle$	1, 2, 3, 4, 5
$\langle 1 \rangle$	2, 3, 4, 5
$\langle 1, 4 \rangle$	2, 3, 4
$\langle 1, 4, 2 \rangle$	2, 4
$\langle 1, 4, 2, 1 \rangle$	4
$\langle 1, 4, 2, 1, 1 \rangle$	\emptyset

Teda našej Hamiltonovskej kružnici zodpovedá v ordinálnej reprezentácii reťazec

$$\langle 1, 4, 2, 1, 1 \rangle.$$

Vyhodnotenie riešenia

Hamiltonovská kružnica má n hrán, takže spočítame vzdialenosť n dvojíc susedných bodov. Aby sme nepočítali rovnaké vzdialenosť veľa krát, je vhodné spočítať na začiatku algoritmu vzdialenosť pre všetky dvojice bodov a uložiť ich do matice typu $(n \times n)$. Za lepšie riešenie považujeme to, ktoré má menší súčet vzdialostí.

Kríženie

Používa sa veľmi veľa krížení, tu spomenieme iba niektoré, a aj tie iba v základnej forme.

Klasické kríženie

Majme dvoch rodičov r_1 a r_2 , zapísaných pomocou reťazcov. Napríklad

$$\begin{aligned}r_1 &: \langle 1, 3, 2, 1, 2, 1 \rangle \\r_2 &: \langle 1, 4, 2, 2, 1, 1 \rangle.\end{aligned}$$

Teraz si vyberieme pozíciu na kríženie, povedzme medzi tretím a štvrtým prvkom, a na tejto pozícii konce reťazcov zameníme. Získame potomkov

$$\begin{aligned}p_1 &: \langle 1, 3, 2, 2, 1, 1 \rangle \\p_2 &: \langle 1, 4, 2, 1, 2, 1 \rangle.\end{aligned}$$

Keď budeme toto kríženie robiť pri ordinálnej reprezentácii, tak obaja potomkovia budú reprezentovať Hamiltonovské kružnice. V ostatných prípadoch bude treba robiť opravy.

Čiastočne párované kríženie

Majme dvoch rodičov zadaných pomocou permutácie

$$\begin{aligned}r_1 &: [2, 3, 4, 5, 6, 7, 8, 1] \\r_2 &: [3, 1, 6, 7, 2, 4, 8, 5].\end{aligned}$$

Teraz si zvolíme pozície, pomocou ktorých prebehne kríženie. Povedzme pozície 3 a 4. Volíme ich tak, aby sme v stĺpcoch dostali čiastočné párovanie. V našom príklade získame 4 – 6 a 5 – 7. Keďže sa tu žiadne číslo neopakuje, dvojice reprezentujú čiastočné párovanie. Teraz pomocou párovania zameníme body v permutácii. Získame potomkov

$$\begin{aligned}p_1 &: [2, 3, 6, 7, 4, 5, 8, 1] \\p_2 &: [3, 1, 4, 5, 2, 6, 8, 7].\end{aligned}$$

Problém je v tom, že potomok nemusí reprezentovať Hamiltonovskú kružnicu. V našom príklade \mathbf{p}_1 reprezentuje Hamiltonovskú kružnicu, avšak \mathbf{p}_2 nie. V takom prípade potomka buď opravíme, alebo ho zahodíme.

Maticové kríženie

Majme rodičov \mathbf{r}_1 a \mathbf{r}_2 zadaných permutačnými maticami

$$\mathbf{r}_1 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix} \quad \text{a} \quad \mathbf{r}_2 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}.$$

Zvolíme si kríženie medzi prvým a druhým plus tretím a štvrtým stĺpcom. Zámenou na týchto pozíciach dostávame

$$\mathbf{p}'_1 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix} \quad \text{a} \quad \mathbf{p}'_2 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}.$$

Kedže takto dostaneme riadky s násobným výskytom jednotiek, niektoré z nich presunieme do nových riadkov. Povedzme

$$\mathbf{p}_1 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix} \quad \text{a} \quad \mathbf{p}_2 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}.$$

Opäť sa môže stať, že potomok nereprezentuje Hamiltonovskú kružnicu, ako \mathbf{p}_1 v našom príklade. V takom prípade ho buď opravíme, alebo zahodíme.

Kríženie pomocou susedov

Majme rodičov

$$\begin{aligned} \mathbf{r}_1 : & (1, 2, 3, 4, 5, 6) \\ \mathbf{r}_2 : & (3, 6, 4, 2, 1, 5). \end{aligned}$$

Zostrojíme zoznam susedov bodov. Napríklad v \mathbf{r}_1 susedí bod 1 s bodmi 2 a 6 a v \mathbf{r}_2 susedí bod 1 s bodmi 2 a 5. Preto sú v zozname bodu 1 susedia 2, 5, 6. Dostávame

$$\begin{aligned} 1 : & 2, 5, 6 \\ 2 : & 1, 3, 4 \\ 3 : & 2, 4, 5, 6 \\ 4 : & 2, 3, 5, 6 \\ 5 : & 1, 3, 4, 6 \\ 6 : & 1, 3, 4, 5 \end{aligned}$$

Teraz začneme v nejakom bode (odporúča sa bod s najmenším počtom susedov) a vyberieme nasledovníka zo zoznamu susedov náhodne. Potom sa presunieme do nasledovníka a procedúru opakujeme. Dostaneme

$$\mathbf{p}'_1 : (1, 5, 4, 3, 6,$$

a ďalej daným spôsobom nemôžeme pokračovať. Vtedy volíme ďalší prvok náhodne tak, aby sme cyklus neuzavreli príliš skoro. Získame

$$\mathbf{p}_1 : (1, 5, 4, 3, 6, 2).$$

Heuristické kríženie

Opäť budeme mať dvoch rodičov \mathbf{r}_1 a \mathbf{r}_2 . Zvolíme si ľubovoľný bod ako počiatocný. Nato prezrieme nasledovníkov tohto bodu u oboch rodičov a vyberieme ten, ku ktorému je od zvoleného bodu menšia vzdialenosť. Takto pokračujeme, pokiaľ to je možné. Keď to možné nie je, zvolíme nasledujúci bod náhodne, pričom dávame pozor, aby sme cyklus neuzavreli príliš skoro.

Pri tomto krížení na 20 bodoch sa zistilo, že keď budú rodičia \mathbf{r}_1 a \mathbf{r}_2 volení náhodne, tak asi 30 % hrán zdedí potomok z \mathbf{r}_1 , 30 % z \mathbf{r}_2 a zvyšných 40 % predstavujú nové hrany. Pritom priemerne sú potomkovia o 10 % lepší ako lepší z rodičov.

Mutácie

Mutácie môžeme uvažovať také isté ako pri simulovanom žíhaní. Avšak z dôvodu nedokonalého kríženia ich prijíname iba vtedy, keď znížia hodnotu účelovej funkcie.

Na záver uvedme, že genetické algoritmy sa používajú vtedy, keď je bodov veľa. V našom príklade stovky, prípadne tisíce. Z toho dôvodu sú aj generácie veľmi veľké. Na kríženie sa vyberá nie viac ako 50 % najlepších jedincov z generácie a pravdepodobnosť mutácie býva okolo 10 %.

Pre záujemcov o ďalšie štúdium stochastických metód s dôrazom na simulované žíhanie a genetické algoritmy odporúčame špecializovanú literatúru [3,5,13].

Cvičenia

CVIČENIE 11.1. Pomocou genetického algoritmu nájdite najkratšiu uzavretú cestu medzi náhodne vygenerovanými n bodmi vo štvorci $\langle 0, 1 \rangle \times \langle 0, 1 \rangle$.

1. Vyberte si jedno kríženie a jednu mutáciu. Potom zvoľte vhodné kódovanie a naprogramujte genetický algoritmus.
2. Vizualizujte najlepšie riešenie v rámci jednej generácie.
3. Porovnajte výsledky a časovú náročnosť pre rôzne veľké populácie a pre rôzne nastavené percentá na kríženie a mutácie.

CVIČENIE 11.2. Navrhnite nové kríženie. Pokúste sa ho navrhnuť tak, aby zachovalo dobré vlastnosti rodičov.

Literatúra

- [1] ANDERSON, D.R., SWEENEY, D.J., WILLIAMS, T.A., CAMM, J.D., MARTIN, R.K.: *An Introduction to Management Science: Quantitative Approaches to Decision Making*. South-Western / Cengage Learning 2012.
- [2] BOYD S., VANDENBERGHE L.: *Convex Optimization*. Cambridge University Press, Cambridge 2004.
- [3] BRYANT, K: *Genetic Algorithms and the Travelling Salesman Problem*. Thesis, Harvey Mudd College, Department of Mathematics 2000.
- [4] CIARLET P.G.: *Introduction to numerical linear algebra and optimisation*. Cambridge University Press, Cambridge 1989.
- [5] ČERNÝ, V.: *Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm*. Journal of Optimization Theory and Applications **45** (1985), 41-51.
- [6] GREFENSTETTE, J., GOPAL, R., ROSMAITA, B., GUCHT, D.: *Genetic algorithms for the travelling salesman problem*. Proceedings of the 1st International Conference on Genetic Algorithms, Hillsdale 1985.
- [7] HAMALA M.: *Nelineárne programovanie*. Alfa, Bratislava 1972.
- [8] HARSHBARGER R.J., REYNOLDS J.J.: *Calculus with applications*. D. C. Heath and Company, Lexington, Massachusetts 1990.
- [9] JUNG, J.H., O'LEARY, D.P.: *Implementing an interior point method for linear programs on a cpu-gpu system*. Electronic Transactions on Numerical Analysis **28** (2008), 174-189.
- [10] KNOR, M.: *Lineárna a nelineárna optimalizácia*. Slovenská Technická Univerzita v Bratislave, Bratislava 2009.
- [11] KNOR, M.: *Teória grafov*. Slovenská Technická Univerzita v Bratislave, Bratislava 2008.
- [12] KOLMAN, P.: *Polynomiální algoritmy pro lineární programování. ITI Series, 2013-311* Karlova Univerzita, Praha 2013.
- [13] KRIVÁ, Z.: *Základný algoritmus simulovaného žíhania. Simulated annealing – basic algorithm*. Forum Statisticum Slovacum **5** (2013), 62-73.
- [14] MATOUŠEK, J.: *Lineární programování a lineární algebra pro informatiky. ITI Series, 2006-311* Karlova Univerzita, Praha 2006.
- [15] TOMA V.: *Teória a algoritmy lineárneho programovania*. Univerzita Komenského, Bratislava 1983.

- [16] VANDERBEI R.J.: *Linear Programming*. Springer, New York 2008.
- [17] VOGT J.: *Primal-Dual Path-Following Methods for Linear Optimization*. Diploma Thesis, University of Konstanz, Konstanz 2008.
- [18] WINSTON W.L.: *Operations research*. 2nd ed., PWS-KENT Publishing Company, Belmont, California 1991.

Obsah

Predhovor	3
1 Podmienky komplementarity	5
2 Metóda vnútorného bodu	14
3 Algoritmy založené na centrálnej ceste	24
4 Karushove-Kuhnove-Tuckerove podmienky	34
5 Dualita v matematickom programovaní	45
6 Úlohy s ohraničeniami tvaru rovníc	54
7 Úlohy konvexného programovania	67
8 Primárno-duálne metódy	77
9 Metóda projekcie gradientu	86
10 Stochastické optimalizačné metódy	97
11 Evolučné algoritmy	105
Literatúra	111

prof. RNDr. Martin Knor, PhD., Ing. Mgr. Lukáš Tomek, PhD.

OPTIMALIZÁCIA 2

Vydala Slovenská technická univerzita v Bratislave vo Vydavateľstve SPEKTRUM STU,
Bratislava, Vazovova 5, v roku 2019.

Edícia vysokoškolských učebníc

Rozsah 113 strán, 29 obrázkov, 7 tabuliek, 4,620 AH, 4,795 VH, 1. vydanie,
edičné číslo 6015, vydané v elektronickej forme.

85 – 211 – 2019

ISBN 978-80-227-4918-3