UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY
KATEDRA APLIKOVANEJ MATEMATIKY A ŠTATISTIKY

Diplomová práca

Andrej Koršňák **Bratislava, 2007**

Odmocninová metóda vnútorného bodu v lineárnom programovaní

DIPLOMOVÁ PRÁCA

Andrej Koršňák

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY
KATEDRA APLIKOVANEJ MATEMATIKY A MATEMATICKEJ ŠTATISTIKY

Ekonomická a finančná matematika

Vedúci diplomovej práce
Doc. RNDr. Milan Hamala, CSc.

BRATISLAVA 2007

Čestne vyhlasujem , že som túto diplomovú prácu
vypracoval samostatne, len s použitím uvedenej literatúry.

.....

V Bratislave, 29.4.2007

Andrej Koršňák

Ďakujem vedúcemu mojej diplomovej práce
p. Doc. RNDr. Milanovi Hamalovi CSc. za cenné rady,
praktické pripomienky, trpezlivosť a poskytnutú literatúru.

Abstrakt.

Názov práce: Odmocninová metóda vnútorného bodu v lineárnom programovaní
Autor: Andrej Koršňák
Škola: Fakulta matematiky, fyziky a informatiky, Univerzita Komenského Bratislava
Školiteľ: doc. RNDr. Milan Hamala, CSc.

V mojej diplomovej práci som sa upriamil na skúmanie vlastností a správania sa odmocninovej bariérovej funkcie aplikovanej na primárnu a primárno – duálnu metódu riešenia úlohy lineárneho programovania.

Kľúčové slová: lineárne programovanie, metódy vnútorného bodu, odmocninová transformačná funkcia

Označenia v tabuľkách numerických experimentov.

Význam symbolov, ktoré budeme používať v tabuľkách pri numerických experimentoch je nasledujúci:

||x(l)-xopt||…….. vzdialenosť nami nájdenej aproximácie x^l optimálneho riešenia od skutočného optimálneho riešenia $\hat{x}$

||y(l)-yopt|……... vzdialenosť nami nájdenej aproximácie y^l optimálnej hodnoty duálnej premennej od skutočnej optimálnej hodnoty duálnej premennej $\hat{y}$

||z(l)-zopt||……...vzdialenosť nami nájdenej aproximácie z^l hodnoty doplnkovej premennej od skutočnej optimálnej hodnoty doplnkovej premennej $\hat{z}$

c*x(l)…….. hodnota účelovej funkcie $c^T x$ v $l-$tej iterácii

c*x(l)-c*xopt…….. odchýlka hodnoty účelovej funkcie $c^T x$ v $l-$tej iterácii od jej hodnoty v optimálnom riešení $\hat{x}$

mi..... hodnota bariérového parametra μ

T(x(l),mi)……... hodnota primárnej transformačnej funkcie $T(x,\mu)$ v $l-$tej iterácii

T(x(l),z(l),mi)……. hodnota primárno – duálnej transformačnej funkcie $T(x,z,\mu)$ v $l-$tej iterácii

||gradient||……... euklidovská norma gradientu $g = \nabla T(x,\mu)$

||hessian||……´frobéniova norma Hessovej matice $H = \nabla^2 T(x,\mu)$:

$$\|H\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{m}(H_{ij})^2}$$

||smer||……. euklidovská norma Newtonovského smeru $s = -\left[\nabla^2 T(x,\mu)\right]^{-1} \nabla T(x,\mu)$

lambda..... optimálna dĺžka kroku λ v smere s

σ ..... parameter vyjadrujúci veľkosť redukcie bariérového parametra μ :

$$\mu_{k+1} = \frac{\mu_k}{\sigma}$$

||rc||..... euklidovská norma rezidua $r_c = A^T y - c$

||rmi||……... euklidovská norma rezidua $r_\mu = \left[diag\left(2y_i\sqrt{z_i} - \mu\right) * e\right]$

dualita..... hodnota vyjadrujúca platnosť vzťahu $b^T y \le c^T x$, kde Y znamená, že vzťah platil vo všetkých testovaných prípadoch, N ani v jednom a Y / N pomer prípadov, v ktorých vzťah platil resp. neplatil

Spádovosť…….. hodnota vyjadrujúca platnosť vzťahu $(\Delta x, \Delta z, \Delta y) * \nabla T(x,z,\mu) < 0$, kde Y znamená, že vzťah platil vo všetkých testovaných prípadoch, N ani v jednom a Y / N pomer prípadov, v ktorých vzťah platil resp. neplatil

CPU time..... čas trvania riešenia danej úlohy v sekundách

Obsah

Úvod.

Lineárne programovania aj napriek tomu, že je špeciálnym prípadom konvexného programovania, sa od svojho vzniku v roku 1947 vyvíjalo samostatne. Za univerzálnu a spoľahlivú metódu na riešenie úlohy lineárneho programovania sa dlhé roky považovala simplexová metóda navrhnutá Gergeom B. Dantzigom. Táto metóda generovala konečnú postupnosť bodov z hranice a jej algoritmus „preskakoval“ z jedného krajného bodu k susednému tak, aby sa zmenšovala hodnota účelovej funkcie. Iné prístupy riešenia ostávali prakticky nepovšimnuté.

Situácia sa zmenila keď v roku 1972 Klee a Minty ukázali, že simplexová metóda nie je polynomiálna. Toto zistenie naštartovalo pokusy o alternatívne prístupy k riešeniu úloh lineárneho programovania. Prielom prišiel v roku 1984 keď Narendra Karmakar publikoval tzv. Projektívny algoritmus pre LP, ktorý bol nielen polynomiálny, ale aj veľmi rýchly. Tento algoritmus úzko súvisel s logaritmickou bariérovou metódou vnútorného bodu a spolu s rokom 1984 je považovaný za „revolučný“ v matematickom programovaní.

Dnes sa v moderných metódach vnútorného bodu využíva najmä logaritmická bariérová funkcia. Medzi jej najvýznamnejšie vlastnosti patrí fakt, že ju možno súčasne aplikovať na primárnu aj duálnu úlohu pri zachovaní symetrických primárno-duálnych vzťahov.

V mojej diplomovej práci som sa upriamil na skúmanie vlastností a správania sa odmocninovej bariérovej funkcie aplikovanej na primárnu a primárno – duálnu metódu riešenia úlohy lineárneho programovania.

Diplomová práca je rozdelená do štyroch kapitol: V prvej kapitole sú zhrnuté základné pojmy teórie lineárneho programovania, teoretické základy metód vnútorného bodu a vlastnosti generátora úloh. Druhá kapitola je venovaná primárnej odmocninovej metóde riešenia úlohy lineárneho programovania, jej popisu a numerickým experimentom. Kapitola tretia sa týka popisu a numerickým experimentom primárno – duálnej odmocninovej metódy riešenia úlohy lineárneho programovania. V poslednej kapitole porovnávame výsledky týchto dvoch skúmaných metód.

Práca je doplnená dvoma prílohami: tabuľkami s podrobnými záznamami z jednotlivých numerických experimentov pre sériu úloh rozmerov $50x100$ a zdrojovým kódom použitého programu.

1. Teoretické základy metód vnútorného bodu.

1.1. Základné pojmy a vzťahy v lineárnom programovaní.

- Formulácia úlohy lineárneho programovania.

Primárnou úlohou lineárneho programovania nazývame optimalizačnú úlohu tvaru

$$Min\left\{ c^T x \mid Ax \geq b \right\} \qquad (P)$$

, kde A je matica typu $m\,x\,n$, $c,\ x \in R^n,\ b \in R^m$. Účelovou funkciou nazveme vzťah $c^T x$. Množinu $K = \left\{ x \in R^n \mid Ax \geq b \right\}$ nazývame množinou prípustných riešení. Optimálnym riešením zase nazveme minimálnu hodnotu účelovej funkcie $c^T x$. Množinu $K^0 = \left\{ x \in R^n \mid Ax > b \right\}$ nazývame relatívnym vnútrom množiny prípustných riešení K . Každú úlohu lineárneho programovania možno preformulovať na primárnu úlohu tohto tvaru.

- Duálna úloha lineárneho programovania.

Duálnou úlohou k primárnej úlohe (P) nazývame úlohu tvaru

$$Max\left\{ b^T y \mid A^T y = c,\ y \geq 0 \right\} \quad . \qquad (D)$$

Podobne ako v prípade primárnej úlohy môžme aj pre duálnu úlohu zadefinovať pojmy ako množina prípustných riešení, či relatívne vnútro úlohy (D). Teda množinou prípustných riešení je množina

$$D = \left\{ y \in R_+^m \mid A^T y = c,\ y \geq 0 \right\}$$

a relatívne vnútro množiny D je množina

$$D^o = \left\{ y \in R_+^m \mid A^T y = c,\ y > 0 \right\}.$$

- Vety o dualite.

Pre hore uvedenú dvojicu duálnych úloh (P) a (D) platia nasledujúce vzťahy:

- Slabá veta o dualite:

  Nech x je prípustné riešenie úlohy (P)a y je prípustné riešenie úlohy (D). Potom

  $$b^T y \leq c^T x .$$

- Silná veta o dualite:

  Ak primárna úloha (P) má optimálne riešenie $\hat{x}$, potom aj duálna úloha (D) má optimálne riešenie $\hat{y}$ také, že

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

- Dôsledky slabej vety o dualite:

  a) Ak dvojica prípustných riešení x^o, y^o úloh $(P),(D)$ spĺňa vzťah

  $$b^T y^o = c^T x^o$$

  tak x^o je optimálnym riešením úlohy (P) a
  y^o je optimálnym riešením úlohy (D).

  b) Ak účelová funkcia úlohy (P) nie je zdola ohraničená, tak duálna úloha (D) nemá prípustné riešenie. Podobne ak účelová funkcia duálnej úlohy(D) nie je zhora ohraničená, tak primárna úloha (P) nemá prípustné riešenie.

- Veta o dualite:

  Ak majú obe úlohy $(P),(D)$ prípustné riešenie, tak obe majú aj optimálne riešenie $\hat{x}$, $\hat{y}$ a optimálne hodnoty ich účelových funkcií sa rovnajú, t.j.

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

- Nutné a postačujúce podmienky optimality.

Body $x \in R^n$, $y \in R^m$ sú optimálnymi riešeniami úloh $(P),(D)$ práve vtedy ak

$$\begin{aligned} Ax - b \geq 0 \quad & A^T y = c \\ y^T (Ax - b) = 0 \quad & \\ y \geq 0 \quad & \end{aligned} \tag{1.1}$$

Primárna úloha (P) býva často upravená na ekvivalentný tvar

$$Min\left\{c^T x \mid Ax - z = b,\ z \geq 0\right\}, \tag{$\widetilde{P}$}$$

z čoho vyplýva možnosť pretransformovania podmienky (1.1) na tvar

$$\begin{aligned} Ax - z = b \quad & z \geq 0 \\ y^T z = 0 \quad & y \geq 0 \\ A^T y = c \quad & \end{aligned} \tag{1.2}$$

Podmienky (1.1) a (1.2) sa v literatúre nazývajú tiež Kuhn-Tuckerove podmienky.

1.2. Newtonova metóda pre nelineárne úlohy.

- Newtonova metóda na riešenie sústav nelineárnych rovníc.

Majme sústavu N – nelineárnych rovníc o N – neznámych tvaru:

$$\begin{aligned} F_1(w_1, \ w_2, \ \cdots, \ w_N) &= 0 \\ F_2(w_1, \ w_2, \ \cdots, \ w_N) &= 0 \\ \vdots \qquad\qquad\quad & \vdots \ \ \vdots \\ F_N(w_1, \ w_2, \ \cdots, \ w_N) &= 0 \end{aligned},$$

ktorú môžeme zapísať vo vektorovom tvare takto

$$F(w) = 0_N, \tag{1.3}$$

kde $F : R^N \to R^N$.

Našim cieľom je odvodiť tzv. Newtonovu metódu na nájdenie koreňa $w^* \in R^N$ danej sústavy (resp. jeho dobrej aproximácie).

Predpokladajme, že poznáme nejaký odhad $w^0 \in R^N$ hľadaného koreňa w^*. V okolí bodu w^0 aproximujeme nelineárne zobrazenie $F : R^N \to R^N$ lineárnym zobrazením odvodeným z Taylorovho rozvoja:

$$F(w^0 + \Delta w) \approx F(w^0) + J(w^0)\Delta w, \tag{1.4}$$

kde $J(w^0)$ je Jacobiho matica zobrazenia F, t.j.

$$J\left(w^0\right)=\begin{bmatrix} \frac{\partial F_1\left(w^0\right)}{\partial w_1} & \frac{\partial F_1\left(w^0\right)}{\partial w_2} & \cdots & \frac{\partial F_1\left(w^0\right)}{\partial w_N} \\ \frac{\partial F_2\left(w^0\right)}{\partial w_1} & \frac{\partial F_2\left(w^0\right)}{\partial w_2} & \cdots & \frac{\partial F_2\left(w^0\right)}{\partial w_N} \\ \vdots & \vdots & & \vdots \\ \frac{\partial F_N\left(w^0\right)}{\partial w_1} & \frac{\partial F_N\left(w^0\right)}{\partial w_2} & \cdots & \frac{\partial F_N\left(w^0\right)}{\partial w_N} \end{bmatrix}.$$

Teda namiesto sústavy nelineárnych rovníc (1.3) budeme riešiť lineárnu sústavu:

$$F\left(w^0\right)+J\left(w^0\right)\Delta w=0_N \quad . \tag{1.5}$$

Za predpokladu, že štvorcová matica $J\left(w^0\right)$ je regulárna, riešením sústavy (1.5) je vektor

$$\Delta w=-J\left(w^0\right)^{-1}F\left(w^0\right).$$

Novú aproximáciu koreňa w^* dostaneme zo vzťahu:

$$w^1 \;=\; w^0+\Delta w \;=\; w^0-J\left(w^0\right)^{-1}F\left(w^0\right).$$

Uvedený postup možno opakovať, t.j. generujeme postupnosť $\left\{w^k\right\}$ vzťahom

$$w^{k+1}=w^k-J\left(w^k\right)^{-1}F\left(w^k\right).$$

- Newtonova metóda na minimalizáciu funkcie n premenných.

Na rozdiel od predchádzajúceho prípadu majme funkciu n premenných

$$T(x), \qquad x=\left[x_1,x_2,\cdots,x_n\right]^T,$$

ktorá má spojité druhé parciálne derivácie.

Našim cieľom je odvodiť metódu na nájdenie minima $\hat{x}\in R^N$ funkcie $T(x)$ (resp. jeho dobrej aproximácie).

Predpokladajme, že poznáme nejaký odhad $x^0\in R^N$ hľadaného minima $\hat{x}$. V okolí bodu x^0 aproximujeme funkciu $T(x)$ Talorovým polynómom druhého stupňa:

$$T\left(x^0+\Delta x\right)\approx T\left(x^0\right)+\nabla T\left(x^0\right)^T\Delta x+\frac{1}{2}(\Delta x)^T\nabla^2T\left(x^0\right)\Delta x=Q(x)\,, \tag{1.6}$$

kde $\nabla T\left(x^0\right)$ je gradient funkcie $T(x)$ a $\nabla^2T\left(x^0\right)$ je Hessova matica druhých parciálnych derivácií tejto funkcie.

Teda derivovaním funkcie (1.6) spolu so skutočnosťou, že funkcia nadobúda extrém, keď jej derivácia je nulová dostávame

$$\nabla Q(x)=\nabla T\left(x^0\right)+\nabla^2T\left(x^0\right)\Delta x=0\,,$$

a následne

$$\Delta x=-\nabla^2T\left(x^0\right)^{-1}\nabla T\left(x^0\right).$$

Nový odhad hľadaného minima $\hat{x}$ dostaneme zo vzťahu

$$x^1=x^0-\nabla^2T\left(x^0\right)^{-1}\nabla T\left(x^0\right).$$

Uvedený postup možno opakovať, t.j. generujeme postupnosť $\left\{x^k\right\}$ vzťahom

$$x^{k+1}=x^k-\nabla^2T\left(x^k\right)^{-1}\nabla T\left(x^k\right).$$

Newtonova metóda ale nezaručuje monotónnosť iteračného procesu, t.j. vlastnosť

$$T\left(x^{k+1}\right)<T\left(x^k\right) \quad . \tag{1.7}$$

Pri porušení vlastnosti (1.7) sa doporučuje skrátenie kroku, to znamená, že postupnosť $\{x^k\}$ budeme generovať vzťahom

$$x^{k+1} = x^k - \lambda^k \nabla^2 T(x^k)^{-1} T(x^k),$$

kde

$$0 < \lambda^k < 1.$$

Metóda, ktorá používa smer a regulovanú dĺžku kroku (optimálnu dĺžku kroku) sa nazýva Modifikovaná Newtonova metóda.

1.3. Logaritmická transformačná metóda riešenia úlohy lineárneho programovania.

Pod transformáciou úlohy lineárneho programovania rozumieme prechod od pôvodne zadanej úlohy k postupnosti optimalizačných úloh, ktoré sú v určitom zmysle ľahšie riešiteľné. Myšlienkou tejto transformácie je zjednodušenie pôvodnej úlohy (teda aj zjednodušenie procesu riešenia).

Účelovú funkciu transformovanej úlohy nazývame transformačnou funkciou a môže nadobúdať rôzne tvary. Najčastejšie používanou transformačnou funkciou je logaritmická transformácia.

- Logaritmická bariérová transformácia.

Logaritmickou transformačnou funkciou pre primárnu úlohu lineárneho programovania

$$Min\{ c^T x \mid Ax \geq b\} \qquad (P)$$

je funkcia

$$T(x,\mu) = c^T x - \mu \sum_{i=1}^{m} \ln(Ax - b)_i \quad , \mu > 0 . \qquad (1.8)$$

Ide o parametrickú transformáciu, pričom parametrom je tu kladné číslo μ. Jednotlivé čiastkové transformované úlohy budú mať tvar:

$$Min\{ T(x,\mu) \mid x \in R^n\} \quad , \qquad (T_\mu)$$

Bod minima každej čiastkovej úlohy označíme ako $x(\mu)$. Z rýdzej konvexnosti funkcie $T(x,\mu)$ pritom vyplýva jednoznačnosť tohto bodu, ktorý spĺňa nutnú a postačujúcu podmienku

$$\nabla T(x(\mu),\mu) = 0_n . \qquad (1.9)$$

- Centrálna trajektória.

Minimá $x(\mu)$ jednotlivých čiastkových úloh (T_μ) tvoria krivku, ktorá prechádza relatívnym vnútrom K^0 množiny prípustných riešení K a končí v optimálnom riešení $\hat{x}$ pôvodnej úlohy (P). Takáto krivka sa nazýva centrálna trajektória.

- Bariérová vlastnosť funkcie:

Keďže na riešenie úloh (T_μ) použijeme metódy, ktorými sa riešia úlohy na voľný extrém, bude postupnosť bodov $x(\mu)$ vďaka logaritmickému členu $\sum_{i=1}^{m} \ln(Ax - b)_i$ odtláčaná od hranice množiny prípustných riešení K. Táto hranica bude preto pôsobiť ako neprekročiteľná bariéra. Pre všetky $\mu > 0$ potom bude bod $x(\mu)$ aj jeho aproximácia, ktorú získame, ležať

v relatívnom vnútre množiny prípustných riešení K, v množine K^0, tzn. bude splnená podmienka $Ax(\mu) > b$. Ide teda o metódu vnútorného bodu.
Bariérovú vlastnosť funkcie $T(x,\mu)$ možno formálnejšie popísať vzťahom:

$$\lim_{(Ax-b)_i \to 0^+} T(x,\mu) = +\infty$$

pre všetky $\mu > 0$ a všetky $i = 1, 2, \cdots, m$.

Navyše, transformačná funkcia je parametrizovaná parametrom $\mu > 0$, ktorý umožňuje meniť váhu kladenú na bariérový člen. Preto v prípade keď μ pôjde k nule, príslušné minimá transformačných funkcií (T_μ) sa budú blížiť k minimu pôvodnej úlohy (P), teda platí

$$x(\mu) \underset{\mu \downarrow 0}{\to} \hat{x}.$$

- Algoritmus transformačnej metódy :
  - Pre dané μ nájdeme približné riešenie úlohy (T_μ) pomocou nejakej metódy voľnej optimalizácie zo zadaného štartovacieho bodu,
  - zmenšíme hodnotu parametra μ a postup opakujeme, štartovacím bodom pre túto minimalizáciu je vypočítané približné riešenie predchádzajúcej úlohy.

- „Perturbované“ Kuhn-Tuckerove podmienky.

Aby sme mohli transformačnú metódu prakticky použiť, analyzujeme nutné a postačujúce podmienky optimality úlohy (T_μ) v súvislosti s (1.1). Z (1.9) vyplýva

$$\nabla T(x(\mu),\mu) = c - \mu \sum_{i=1}^{m} \frac{1}{(Ax(\mu)-b)_i} A_i = 0_n, \qquad i = 1,..,m \ .$$

Použitím substitúcie

$$y_i(\mu) = \mu \frac{1}{(Ax(\mu)-b)_i}, \qquad i = 1,...,m$$

dostávame

$$c - \sum_{i=1}^{m} y_i(\mu) A_i = 0_n, \qquad i = 1,...,m,$$

tzn. platí podmienka $A^T y = c$. Zo substitúcie ďalej vyplýva

$$y_i(\mu)(Ax(\mu)-b)_i = \mu, \qquad i = 1,...,m.$$

Navyše z vlastnosti logaritmu vylýva podmienka $Ax - b > 0$.
Dostávame sústavu podmienok, ktorá hovorí, že body $x \in R^n$, $y \in R^m$ sú optimálnymi riešeniami úlohy (T_μ) práve vtedy ak

$$\begin{aligned} Ax - b &\geq 0 \quad A^T y = c \\ y_i(Ax-b)_i &= \mu \\ y &\geq 0 \end{aligned} \qquad (1.10)$$

V limitnom prípade $\mu \downarrow 0^+$ prechádza sústava (1.10) do sústavy Kuhn-Tuckerových podmienok (1.1) pre pôvodnú úlohu lineárneho programovania. Sústava (1.10) sa preto nazýva aj sústavou „perturbovaných“ Kuhn-Tuckerových podmienok pre úlohu (T_μ).

1.4. Odmocninová transformačná metóda riešenia úlohy lineárneho programovania.

- Odmocninová transformačná funkcia.

Opäť uvažujme úlohu

$$Min\left\{ c^T x \mid Ax \geq b \right\}. \qquad (P)$$

Odmocninovou transformačnou funkciou pre túto úlohu je funkcia

$$T(x,\mu) = c^T x - \mu \sum_{i=1}^{m} (Ax-b)_i^{\frac{1}{2}} \qquad \mu > 0\,. \qquad (1.11)$$

Bod minima $x(\mu)$ každej čiastkovej úlohy (T_μ) je ako v predchádzajúcom prípade jednoznačný a spĺňa nutnú a postačujúcu podmienku (1.9).

- Kvázibariérová vlastnosť funkcie:

Platí, že $\lim_{(Ax-b)_i \to 0^+} T(x,\mu)$ je konečná, avšak

$$\lim_{(Ax-b)_i \to 0^+} \frac{\partial}{\partial x_i} T(x,\mu) = -\infty$$

pre všetky $\mu > 0$ a všetky $i = 1,...,m$ .V takom prípade hovoríme, že funkcia $T(x,\mu)$ má kvázibariérovú vlastnosť. Je totiž teraz definovaná aj na hranici množiny prípustných riešení K, avšak na tejto hranici má nekonečnú zápornú deriváciu. To znamená, že minimum funkcie $T(x,\mu)$ sa nemôže dosiahnuť na hranici množiny K a aj keď v tomto prípade neúčinkuje transformačný člen ako bariéra, jednako zabezpečuje, že ak minimum úlohy $T(x,\mu)$ budeme hľadať na množine K metódami, ktorými sa riešia úlohy na voľný extrém, nájdeme ho v relatívnom vnútri K^0 množiny K.

Rovnako ako v prípade logaritmickej transformácie, keď μ pôjde k nule, príslušné minimá transformačných funkcií sa budú blížiť k minimu pôvodnej úlohy, teda platí

$$x(\mu) \underset{\mu \downarrow 0}{\to} \hat{x}\,.$$

- „Perturbované“ Kuhn-Tuckerove podmienky.

Aj v prípade odmocninovej transformačnej funkcie analyzujeme nutné a postačujúce podmienky optimality úlohy (T_μ) v súvislosti s (1.1) a (1.10). Z (1.9) vyplýva

$$\nabla T(x(\mu),\mu) = c - \frac{\mu}{2} \sum_{i=1}^{m} \frac{1}{\sqrt{(Ax(\mu)-b)_i}} A_i = 0_n, \qquad i = 1,..,m\ .$$

Použitím substitúcie

$$y_i(\mu) = \frac{\mu}{2} \frac{1}{\sqrt{(Ax(\mu)-b)_i}}, \qquad i = 1,...,m$$

dostávame

$$c - \sum_{i=1}^{m} y_i(\mu) A_i = 0_n, \qquad i = 1,...,m\,,$$

tzn. platí podmienka $A^T y = c$. Zo substitúcie ďalej vyplýva

$$y_i(\mu)\sqrt{(Ax(\mu)-b)_i} = \frac{\mu}{2}, \qquad i = 1,...,m\,,$$

teda

$$[y_i(\mu)]^2 \left(Ax(\mu)-b\right)_i = \frac{\mu^2}{4}, \qquad i = 1,...,m .$$

Navyše z vlastnosti odmocniny vyplýva podmienka $Ax - b \geq 0$.

Dostávame sústavu podmienok, ktorá hovorí, že body $x \in R^n$, $y \in R^m$ sú optimálnymi riešeniami úlohy (T_μ) práve vtedy ak

$$\begin{aligned} Ax - b &\geq 0 \qquad A^T y = c \\ y_i^2 \left(Ax - b\right)_i &= \frac{\mu^2}{4} \\ y &\geq 0 \end{aligned} \tag{1.12}$$

V limitnom prípade $\mu \downarrow 0^+$ prechádza sústava (1.12) do sústavy Kuhn-Tuckerových podmienok (1.1) pre pôvodnú úlohu lineárneho programovania. Sústava (1.12) sa preto nazýva aj sústavou „perturbovaných" Kuhn-Tuckerových podmienok pre úlohu (T_μ).

1.5. Generátor úlohy lineárneho programovania $(\tilde{P})$ s vopred zadaným optimálnym riešením.

Chceme vytvoriť úlohu lineárneho programovania

$$Min\left\{c^T x \mid A x - z = b,\ z \geq 0\right\}, \tag{$\tilde{P}$}$$

s vopred zadaným optimálnym riešením $\hat{x} > 0_n$.

Pri konštrukcii musíme použiť aj jej duálnu úlohu

$$Max\left\{ b^T y \mid A^T y = c,\ y \geq 0\right\}. \tag{$\tilde{D}$}$$

Algoritmus.

1. Určíme rozmery $n\,x\,m$ úlohy lineárneho programovania, kde $n =$ počet premenných a $m =$ počet ohraničení.
2. Zvolíme kladnú maticu A rozmerov $m\,x\,n$, kde i – ty riadok matice A budeme označovať symbolom A_i^T.
3. Zvolíme optimálne riešenie $\hat{x} > 0_n$.
4. Zvolíme optimálnu hodnotu doplnkovej premennej $\hat{z} \geq 0_m$, kde pre polovicu zložiek bude $\hat{z}_i > 0$ a pre druhú polovicu zložiek $\hat{z}_i = 0$.
5. Dopočítame vektor $b \in R^m$ vzťahom $b = A\hat{x} - \hat{z}$.
6. Zvolíme optimálnu hodnotu duálnej premennej $\hat{y} > 0_m$, tak aby platilo
$$\hat{y}_i\,\hat{z}_i = 0 \quad , \quad \hat{y}_i + \hat{z}_i > 0 .$$
7. Dopočítame vektor $c \in R^n$ pomocou duálnej úlohy $(\tilde{D})$ vzťahom $c = A^T \hat{y}$.
8. Zvolíme $\rho > 0$ vyjadrujúce vzdialenosť štartovacieho bodu x^o od optimálneho riešenia $\hat{x}$, t.j. $\rho = \left\|\hat{x} - x^o\right\|$.
9. Zvolíme štartovací bod $x^o > \hat{x}$ tak, aby platilo
$$\left\|\hat{x} - x^o\right\| = \rho .$$

Štartovací bod x^o konštruujeme nasledovne:

Najskôr zvolíme pomocný bod $\xi^o \in R^n_{++}$ v každej zložke vačší ako optimálne riešenie $\hat{x}$, t.j. $\xi^o > \hat{x} > 0_n$. Z bodu $\hat{x}$ konštruujeme polpriamku pretínajúcu bod ξ^o:

$$x_\nu = \hat{x} + \nu\left(\xi^o - \hat{x}\right) \quad \nu > 0 .$$

Vypočítame $\nu > 0$ tak , aby platilo

$$\left\|x_\nu - \hat{x}\right\| = \rho .$$

Odtiaľ

$$\left\|\nu\left(\xi^o - \hat{x}\right)\right\| = \rho \quad \Leftrightarrow \quad \nu\left\|\xi^o - \hat{x}\right\| = \rho \Leftrightarrow \quad \nu_o = \rho \Big/ \left\|\xi^o - \hat{x}\right\| \quad \Rightarrow \quad x^o = \hat{x} + \nu_o\left(\xi^o - \hat{x}\right)$$

Nakoľko vektor ξ^o je vo všetkých zložkách väčší ako optimálne riešenie $\hat{x}$, výsledný štartovací bod bude tiež väčší vo všetkých zložkách, čím bude splnená nerovnosť $Ax^o > b$. Teda štartovací bod x^o bude ležať vo vnútri množiny prípustných riešení K.

10. Dopočítame štartovací bod $z^o > 0_m$ vzťahom

$$z^o = \left(Ax^o - b\right).$$

11. Zvolíme štartovací bod $y^o > 0_m$, ktorý nemusí byť prípustným riešením.

2. Primárna odmocninová metóda riešenia úlohy lineárneho programovania.

V diplomovej práci sa budeme venovať dvom metódam vnútorného bodu: primárnej a primárno-duálnej. V obidvoch metódach ako transformačnú funkciu využijeme odmocninovú funkciu

$$T(x,\mu)=c^T x-\mu\sum_{i=1}^{m}\sqrt{(Ax-b)_i}\qquad \mu>0,$$

resp. jej ekvivalentný tvar

$$T(x,z,\mu)=c^T x-\mu\sum_{i=1}^{m}\sqrt{z_i}\ ,\qquad Ax-z=b,\ \mu>0.$$

Primárna metóda bude využívať pri hľadaní minima len primárnu premennú, teda pôjde o klasickú modifikovanú Newtonovu metódu na minimalizáciu funkcie n - premenných . Duálnu premennú budeme odhadovať, aby sme ukázali jej správanie sa. Algoritmus Primárno-duálnej metódy bude závislý aj na duálnej premennej, teda bude riešiť sústavu nelineárnych rovníc . Následne porovnáme kvalitu týchto metód.

2.1. Popis metódy.

Máme úlohu lineárneho programovania tvaru:

$$Min\left\{c^T x \mid Ax\geq b\right\}. \qquad (P)$$

K nej máme duálnu úlohu:

$$Max\left\{b^T y \mid A^T y=c, y\geq 0\right\} \qquad (D)$$

Ako transformačnú funkciu použijeme odmocninovú transformačnú funkciu tvaru:

$$T(x,\mu)=c^T x-\mu\sum_{i=1}^{m}\sqrt{(Ax-b)_i}\quad \mu>0 \qquad (2.1.)$$

a namiesto riešenia pôvodnej úlohy lineárneho programovania budeme riešiť postupnosť úloh na voľné minimum:

$$\underset{x\in R^n}{Min}\left\{T(x,\mu)=c^T x-\mu\sum_{i=1}^{m}\sqrt{(Ax-b)_i}\mid x\in K^0\right\}, \qquad (T_\mu)$$

pričom optimálne riešenie tejto úlohy spĺňa podmienku

$$\nabla T(\hat{x}(\mu),\mu)=0_n.$$

Opakovanému riešeniu týchto úloh pre zmenšujúci sa parameter $\mu_k\downarrow 0^+$ hovoríme vonkajšie iterácie. Klesajúca postupnosť $\{\mu_k\}_{k=1}^{\infty}$ je tvorená vzťahom $\mu_{k+1}=\frac{\mu_k}{\sigma}$, kde σ predstavuje redukčný parameter.

V skutočnosti ani nepotrebujeme nájsť optimálny bod $\hat{x}(\mu_k)$ presne, postačí nám jeho „dostatočne presná“ aproximácia. Na hľadanie tejto aproximácie použijeme modifikovanú Newtonovu metódu na minimalizáciu funkcie n – premenných popísanú v kapitole 1.2.. Na reguláciu dĺžky kroku využijeme algoritmus metódy zlatého rezu, pomocou ktorého získame „optimálny“ krok. Body získané počas iteračného procesu modifikovanej Newtonovej metódy sú spresňujúcimi aproximáciami hľadaného optima $\hat{x}(\mu_k)$. Prechod od bodu $x_i(\mu_k)$ do bodu $x_{i+1}(\mu_k)$ nazveme vnútornou iteráciou metódy.

Ako štartovací bod $x^0(\mu_k)$ pre každú vnútornú iteráciu zvolíme poslednú aproximáciu získanú v predošlej vonkajšej iterácii s výnimkou prvej vonkajšej iterácie. Spôsob voľby štartovacieho bodu modifikovanej Newtonovej iterácii v prvej vonkajšej iterácii bližšie nešpecifikujeme, predpokladáme že prípustný štartovací bod je zadaný na vstupe. Otvorenou otázkou taktiež ostáva hodnota počiatočného bariérového parametra $\mu_1 > 0$. Ten sa v praxi volí nasledovne:

$$\mu_1 = \frac{1}{m}\sum_{i=1}^{m}\sqrt{4\left(y_i^0\right)^2\left(Ax^0 - b\right)_i}\ . \qquad (2.2.)$$

Z uvedeného popisu metódy vyplýva, že $\lim_{\mu_k \downarrow 0}\hat{x}(\mu_k) = \hat{x}$. Ale kedy iteračný proces zastaviť? Z teórie lineárneho programovania, konkrétne z Vety o dualite vieme, že v optimálnom riešení $(\hat{x}, \hat{y})$ platí $c^T\hat{x} - b^T\hat{y} = 0$. Algoritmus teda možno zastaviť, ak je rozdiel dostatočne malý.

Nakoľko generátor úlohy lineárneho programovania použitý v diplomovej práci vytvára úlohu s vopred zadaným optimálnym riešením, ďalším kritériom presnosti môže byť vzdialenosť aproximácie od optimálneho riešenia, teda

$$\left\|x_i(\mu_k) - \hat{x}\right\| < \varepsilon\ ,$$

kde $\varepsilon > 0$ je nami zvolené kritérium presnosti.

„STOP“ kritériom pre vnútorné iterácie existuje tiež viacero alternatív. Môže nás napríklad zaujímať veľkosť gradientu transformačnej funkcie. Teda aproximácia bodu centrálnej trajektórie $\hat{x}(\mu_k)$ môže byť pre nás dostatočne presná , ak

$$\left\|\nabla T(x,\mu)\right\| < \varepsilon\ .$$

Ja sa v diplomovej práci pokúsim experimentálne určiť pre testované prípady „optimálny“ počet vnútorných iterácií.

.

2.2. Algoritmus primárnej odmocninovej metódy.

Našim ďalším krokom bude detailne popísať algoritmus primárnej „odmocninovej“ metódy použitý v diplomovej práci. Pomocou vyššie uvedeného generátora vytvoríme primárnu úlohu (P), duálnu úlohu (D) , odmocninovú transformačnú funkciu (T_μ) a štartovací bod $x^o \in R^n$. Zvolíme si postupnosť $\{\mu_l\}_{l=1}^{\infty}$ a vyberieme jej prvý člen, ktorý zafixujeme pre prvú vonkajšiu iteráciu. Následne môžme začať s iteráciou vnútornou. Tú zabezpečuje modifikovaná Newtonova metóda:

- Výpočet gradientu:

$$g_k = \nabla T(x,\mu) = c - \frac{1}{2}\mu\sum_{i=1}^{m}\frac{1}{\sqrt{(Ax-b)_i}}A_i$$

Na to nám slúži funkcia gTxmi (A, b, c, x, g, mi), ktorá počíta gradient transformačnej funkcie.

Vstup: $A^T = [A_1 \mid A_2 \mid \ldots \mid A_m]$, b , c , μ a $x \in R^n$ taký, že $Ax > b$

Výstup: $g = \nabla T(x,\mu)$

Algoritmus: $g = 0_n$

$for \quad i = 1,...,m$

$\alpha = (A x - b)_i$

$\beta = 1/\sqrt{\alpha}$

$g = g * \beta A_i$

……

$g = g + \frac{\mu}{2} * g$

- Výpočet Hessovej matice:

$$H = \nabla^2 T(x, \mu) = \frac{1}{4}\mu \sum_{i=1}^{m} (Ax - b)_i^{-\frac{3}{2}} (A_i A_i^T)$$

Použijeme funkciu hTxmi (A, b, x, H, mi), ktorá počíta Hessovu maticu transformačnej funkcie.

Vstup: $A^T = [A_1 \mid A_2 \mid \ldots \mid A_m]$, b, x, μ

Výstup: $H = \nabla^2 T(x, \mu)$

Algoritmus: $H = 0_{n x n}$

$for \quad i = 1,\ldots,m$

$\alpha = (A x - b)_i$

$\beta = 1/\alpha^{\frac{3}{2}}$

$V = \beta A_i$ (pomocný vektor)

$H = H + V * A_i^T$

……

$H = \frac{\mu}{4} * H$

- Výpočet Newtonovského smeru:

Výpočet smeru $s = -H^{-1} g$ realizujeme riešením sústavy rovníc $H s = -g$ Q-R metódou. Použijeme nato funkciu HansLaws (g, H, s).

Vstup: H, g

Výstup: smer $s \in R^n$

Algoritmus: 1) utvorenie rozšíreného poľa G rozmerov $n\,x(n+1)$

$G = [H \mid -g]$

2) *for* $j = 1,\ldots,n$

$u = \max_i |G_{ij}|$

$if\ (u = 0)THEN\ STOP$

$u_{inv} = 1/u$

$sum = \sum_{i=j}^{n} (G_{ij} * u_{inv})^2$

$u = u * \sqrt{sum}$

$if\ (G_{jj} > 0)THEN\ u = -u$

$G_{jj} = G_{jj} - u$

$\alpha = u * G_{jj}$

$if\ (\alpha = 0)THEN\ GOTO\ koniec$

$\alpha_{inv} = 1/\alpha$

$for\quad k = j+1,\ldots,n+1$

$sum = \sum_{i=j}^{n} G_{ij} * G_{ik}$

……

$sum = sum * \alpha_{inv}$

$for\quad i = j,\ldots,n$

$G_{ik} = G_{ik} + sum * Gij$

……

$G_{jj} = u$

$koniec$

……

3) riešime trojuholníkovú sústavu

- Výpočet dĺžky kroku λ.

Najprv vypočítame maximálnu prípustnú dĺžku kroku $\overline{\lambda} > 0$ a potom metódou zlatého rezu približne riešime úlohu

$$Min\left\{\varphi(\lambda) = T\left(x^k + \lambda s^k, \mu\right) \mid 0 \le \lambda \le \overline{\lambda}\right\}.$$

I. **Výpočet $\overline{\lambda}$.**

Zo vzťahu

$$A_i\left(x^k + \lambda s^k\right) \ge b_i, \qquad i = 1,..,m$$

dostávame

$$A_i x^k + \lambda A_i s^k \ge b_i.$$

Nakoľko vieme, že $A_i x^k > b_i$ pre $\forall i = 1,...,m$, môžu nastať 2 prípady:

- $\exists i : A_i s^k < 0$, teda existuje „priesečník“ $\overline{\lambda}$ polpriamky $x(\lambda) = x^k + \lambda s^k \quad (\lambda > 0)$ s množinou prípustných riešení $K = \{x \in R^n \mid Ax \geq b\}$. Potom maximálnu prípustnú dĺžku kroku vypočítame zo vzťahu:

$$\overline{\lambda} = \min \overline{\lambda}_i, \quad i \in I \quad kde\ \overline{\lambda}_i = \frac{A_i x^k - b_i}{-A_i s^k}.$$

- $\forall i : A_i s^k \geq 0 \quad , \lambda \to \infty$.

Vieme, že

$$\varphi'(0) = \nabla T\left(x^k, \mu\right)^T s^k < 0,$$

Zvolíme postupnosť $\{\lambda_k\}_{k=0}^{\infty}$, kde

$$\lambda_0 = 0,\ \lambda_1 = 1 \text{ a pre } k > 1 \ \ \lambda_k = \tau^{k-1} + \lambda_{k-1}, \tau = \frac{1}{2}\left(\sqrt{5}+1\right).$$

Nastavíme $k = 1$ a následne určíme interval, na ktorom budeme pomocou metódy zlatého rezu hľadať optimálnu dĺžku kroku následovne:

If $\varphi(\lambda_{k+1}) \geq \varphi(\lambda_k)$ then $\underline{\lambda} = \lambda_{k-1}$ a $\overline{\lambda} = \lambda_{k+1}$

If $\varphi(\lambda_{k+1}) < \varphi(\lambda_k)$ then $k = k+1$, cyklus opakovať.

Tento algoritmus nazveme *I. fáza*.

II. Výpočet optimálnej dlžky kroku λ.

Po určení maximálnej prípustnej dĺžky kroku $\overline{\lambda} > 0$, minimalizujeme funkciu $\varphi(\lambda) = T\left(x^k + \lambda s^k, \mu\right)$ na intervale $\langle 0, \overline{\lambda} \rangle$ (v prípade neexistencie priesečníka na intervale $\langle \underline{\lambda}, \overline{\lambda} \rangle$) metódou *Zlatého rezu.*

Algoritmus metódy zlatého rezu:

Vstup: a/ Unimodálna funkcia $f(x)$ definovaná na intervale $\langle a, b \rangle$.

b/ Požadovaná presnoť $\varepsilon > 0$.

Výpočet:

0/ $\tau = \left(\sqrt{5}+1\right)/2 \quad , \quad z_1 = 2 - t \quad , \quad z_2 = \tau - 1'$

1/ $c_1 = a + z_1(b-a) \quad , \quad f_1 = f(c_1)$

$c_2 = a + z_2(b-a) \quad , \quad f_2 = f(c_2)$

2/ IF $f_1 < f_2$ THEN GO TO 4

3/ (Prípad $f_1 \geq f_2$):

Položme: $a = c_1 \quad , \quad c_1 = c_2 \quad , \quad f_1 = f_2$

IF $(b-a) < \varepsilon$ THEN GO TO 5

ELSE $c_2 = a + z_2(b-a) \quad , \quad f_2 = f(c_2)$

GO TO 2

4/ (Prípad $f_1 < f_2$):

Položme: $b = c_2 \quad , \quad c_2 = c_1 \quad , \quad f_2 = f_1$

IF $(b-a) < \varepsilon$ THEN GO TO 5

ELSE $c_1 = a + z_1(b-a) \quad , \quad f_1 = f(c_1)$

GO TO 2

Výstup:

5/ $x_0 = c_1 \quad , \quad f_0 = f_1$

6/ STOP

Na výpočet dĺžky kroku použijeme nasledovné naprogramované funkcie:

- Funkcia Txmi (A, b, c, x, mi), ktorá počíta funkčnú hodnotu transformačnej funkcie

$$T(x,\mu) = c^T x - \mu \sum_{i=1}^{m} \sqrt{(Ax-b)_i} \ .$$

Vstup: $A^T = [A_1 \mid A_2 \mid \ldots \mid A_m]$, b, c, μ, $x \in R^n$ taký, že $Ax > b$

Výstup: funkčná hodnota *Txmi* transformačnej funkcie

Algoritmus:

$Txmi = c^T x$
for $i = 1, \ldots, m$
⋮ $\alpha = (Ax - b)_i$
⋮ $\beta = \beta + \sqrt{\alpha}$
⋮……
$Txmi = Txmi - \mu * \beta$

- Funkcia Lambda (A, b, c, s, x, zlrez), ktorá slúži na výpočet priesečníka polpriamky $x(\lambda) = x^k + \lambda s^k \quad (\lambda > 0)$ s množinou prípustných riešení $K = \{x \in R^n \mid Ax \geq b\}$. Na základe jeho existencie, resp. neexistencie rozhoduje o tom, či použijeme metódu zlatého rezu na nájdenie optimálneho kroku λ, alebo či použijeme I.fázu za účelom nájdenia konečného intervalu, na ktorom budeme hľadať optimálnu dĺžku kroku λ.

Vstup: $A^T = [A_1 \mid A_2 \mid \ldots \mid A_m]$, b, c, s, x, μ, počet krokov zlatého rezu *zlrez*

Výstup: privolanie funkcie *FazaI* alebo *ZlatyRez* a následný výpočet λ

Algoritmus:

$u = 0, \ \lambda_0 = 0, \ \overline{\lambda} = 10^{10}$
for $i = 1, \ldots, m$
⋮ $\gamma = A_i^T * s$
⋮ *if* $(\gamma < 0)$
⋮ ⋮ $\alpha = (Ax - b)_i$
⋮ ⋮ $\lambda_0 = -\alpha / \gamma$
⋮ ⋮ $u = 1$
⋮ ⋮ *if* $(\lambda_0 < \overline{\lambda}) \overline{\lambda} = \lambda_0$
⋮ ⋮……
⋮……
if $(u = 0)$ *GO TO FazaI* // popisaný na str.18
else *GO TO Zlaty* Re z // popísaný na str. 19

- Funkcia FazaI (A, b, c, x, s, zlrez) slúži na výpočet konečného intervalu $\langle \lambda_0, \lambda_2 \rangle$ n na ktorý sa bude aplikovať metóda zlatého rezu. Použije sa vtedy, keď pre $\forall i = 1, \ldots, m$ platí $A_i^T s \geq 0$.

Vstup: $A^T = [A_1 \mid A_2 \mid \ldots \mid A_m]$, b , c , s , x , počet krokov zlatého rezu $zlrez$
Výstup: interval, na ktorom budeme aplikovať metódu zlatého rezu

Algoritmus: $\tau = (\sqrt{5}+1)/2$
$\lambda_2 = \tau^0, \quad \lambda_1 = 0, \quad \lambda_0 = 0, \quad i = 1$
$x^1 = x + \lambda_1 * s$
$x^2 = x + \lambda_2 * s$
$while(Txmi(A,b,c,x^2,\mu) < Txmi(A,b,c,x^1,\mu))$
$\lambda_0 = \lambda_1, \quad \lambda_1 = \lambda_2, \quad \lambda_2 = \lambda_2 + \tau^i, \quad i = i+1$
$x^1 = x + \lambda_1 * s$
$x^2 = x + \lambda_2 * s$
……
GO TO Zlaty Re *z* //popísaný na str. 19

- Funkcia ZlatyRez (A, b, c, x, s, mi, aa, bb, zlrez), ktorá vypočíta optimálnu dĺžku kroku λ. Počet krokov metódy Zlatého rezu môžme regulovať, čo bude aj predmetom našich experimentov.

Vstup: $A^T = [A_1 \mid A_2 \mid \ldots \mid A_m]$, b, c, s, x, μ, dolná hranica intervalu aa, horná hranica intervalu bb, počet krokov zlatého rezu $zlrez$

Výstup: optimálna dĺžka kroku λ

Algoritmus:

$\tau = (\sqrt{5}+1)/2, \quad z^1 = 2-\tau, \quad z^2 = \tau - 1, \quad k = 0$

$c^1 = aa + z_1 * (bb - aa)$

$x^1 = x + c^1 * s$

$f^1 = Txmi(A, b, c, x^1, \mu), \quad c^2 = aa + z^2 * (bb - aa)$

$x^2 = x + c^2 * s$

$f^2 = Txmi(A, b, c, x^2, \mu)$

$while(k \neq zlrez)$

$\vdots \quad if\ (f^1 < f^2)$

$\vdots \quad \vdots \quad bb = c^2, \quad c^2 = c^1, \quad f^2 = f^1, \quad c^1 = aa + z^1 * (bb - aa)$

$\vdots \quad \vdots \quad x^1 = x + c^1 * s$

$\vdots \quad \vdots \quad f^1 = Txmi(A, b, c, x^1, \mu)$

$\vdots \quad \vdots \ldots\ldots\ldots\ldots$

$\vdots \quad else$

$\vdots \quad \vdots \quad aa = c^1, \quad c^1 = c^2, \quad f^1 = f^2, \quad c^2 = aa + z^2 * (bb - aa)$

$\vdots \quad \vdots \quad x^2 = x + c^2 * s$

$\vdots \quad \vdots \quad f^2 = Txmi(A, b, c, x^2, \mu)$

$\vdots \quad \vdots \ldots\ldots\ldots\ldots$

$\vdots \quad k = k + 1$

$\vdots \ldots\ldots\ldots\ldots$

$return \quad \lambda = c^1$

.

- <u>Výpočet novej iterácie</u> $x^{k+1} = x^k + \lambda s^k$

- <u>Opakovanie cyklu.</u>
  Tento budeme opakovať, kým nebude splnené nami stanovené „STOP“ kritérium.

- Vypočítali sme teda „dostatočnú“ aproximáciu bodu centrálnej trajektórie $x(\mu_k)$ prvej transformačnej úlohy, čím sme ukončili aj prvú vonkajšiu iteráciu. Z postupnosti $\{\mu_l\}_{l=1}^{\infty}$ teraz vyberieme jej ďalší člen a prejdeme na druhú vonkajšiu iteráciu a jej sériu iterácii vnútorných. Toto aplikujeme, kým nedostaneme dostatočne presné optimálne riešenie $\hat{x}$ úlohy lineárneho programovania (P).

- Nakoľko máme definovanú aj duálnu úlohu, môžme sledovať aj správanie sa odhadu duálnej premennej. Ona samotná nemá vplyv pri výpočte optimálneho riešenia $\hat{x} \in R^n$, teda jej hodnoty vyplývajú len zo správania sa primárnej premennej. Keďže v iteráciách Newtonovej metódy sa prakticky rieši sústava

$$\nabla T(x,\mu) = 0_n \quad \Rightarrow \quad \frac{1}{2}\mu \sum_{i=1}^{m} \frac{1}{\sqrt{(Ax-b)_i}} A_i = c$$

a súčasne vieme, že $A^T y = c$.

Potom

$$y_i^k \approx \frac{1}{2}\mu \frac{1}{\sqrt{(Ax^k - b)_i}} > 0 \quad i = 1,2,...,m$$

je približné riešenie duálnej úlohy.

Ďalej zo slabej vety o dualite vyplýva

$$b^T y^k \le b^T \hat{y} = c^T \hat{x} \le c^T x^k .$$

2.3. Numerické experimenty.

Cieľom tejto kapitoly je otestovanie numerického správania sa primárnej odmocninovej metódy riešenia úlohy lineárneho programovania. Na tento účel sme použili programovú realizáciu primárneho algoritmu, ktorá je obsahom prílohy B. Konkrétne sme si ako ciele vytýčili:

- sledovať správanie sa metódy v závislosti od presnosti výpočtu optimálnej dĺžky kroku λ meranej počtom iterácií metódy zlatého rezu a následne výber takého počtu, aby príslušná Newtonova metóda bola najefektívnejšia,

- sledovať správanie sa metódy pre rôzne počty Newtonovských iterácií a následne empiricky vybrať taký „optimálny počet“ týchto iterácií, aby celkový výpočet bol najefektívnejší,

- porovnať výpočtovú zložitosť Newtonovského smeru s a optimálneho kroku λ,

- sledovať správanie sa metódy pre rôzne stratégie výberu bariérového parametra μ a následne vybrať stratégiu, s ktorou metóda dosahuje najlepšie výsledky.

Experimenty sme robili na sériách úloh nasledovných rozmerov $n \times m$: 5 x 10, 10 x 20 a 50 x 100. Uvažované série boli vygenerované pomocou generátora z kapitoly 1.5., takže pre úlohu bolo známe hneď aj teoreticky optimálne riešenie. Generátor pritom súčasne s úlohou generoval aj prípustný počiatočný vnútorný bod, tzv. štartovací bod algoritmu.

V prvom rade sme v jednotlivých experimentoch sledovali počet iterácií, ktoré sa urobia, kým nájdu aproximáciu riešenia zadanej úlohy lineárneho programovania s nami požadovanou presnosťou, resp. sme sledovali dosiahnutú presnosť riešenia po určitom počte iterácií.

Ako sme už uviedli, pri riešení úlohy lineárneho programovania sme sa hľadaním vhodného počiatočného vnútorného bodu nezaoberali. Predpokladali sme, že ho máme k dispozícii. Zostala nám však otázka voľby vhodného štartovacieho μ, ktoré pri použití vzťahu $(2.2.)$ môže viesť k priveľkému dôrazu na transformačný člen a k následnému predĺženiu výpočtu optimálneho riešenia. Naopak pri voľbe veľmi malého parametra, môže nastať situácia, že sa rýchlo dostaneme na hranicu množiny prípustných riešení a tým tiež spomalíme rýchlosť konvergencie algoritmu.

Pretože detailné záznamy z experimentov sú pomerne obsiahle, uvádzame ich v prílohe A.

Analýza počtu iterácií metódy zlatého rezu.

V rámci tohto experimentu sme sledovali presnosť s akou metóda nachádza aproximáciu bodu centrálnej trajektórie pre $\mu = 1$ pri použití rôzneho počtu iterácií metódy zlatého rezu v jednotlivých Newtonovských iteráciách. Výsledky experimentu sú zhrnuté v tabuľkách 2.1. – 2.3.

Spoločné vstupné parametre tabuliek 2.1. – 2.3.:
$\mu = 1$, $\rho = 100$ je vzdialenosť štartovacieho bodu x^0 od optimálneho riešenia $\hat{x}$.

Tab 2.1. *rozmer úlohy* $(n\,x\,m) = (5\,x\,10)$

	počet iterácii Newtonovej metódy			
	2	3	4	5
počet iterácií zlatého rezu	**priemerné vzdialenosti bodu optima *xopt* od vypočítanej aproximácie *x(1)* centrálnej trajektórie *x(mi)***			
5	157.8059	104.3930	56.9796	26.7817
10	150.7574	86.9999	38.3510	14.1033
15	150.4824	83.2916	33.6596	10.8061
20	150.4783	83.2232	33.6057	6.3528
25	150.4784	83.2172	33.6121	4.3503
30	150.4784	83.2167	33.6125	4.3518
35	150.4784	83.2167	33.6125	4.3519
40	150.4784	83.2167	33.6125	4.3519
45	150.4784	83.2167	33.6125	4.3519
50	150.4784	83.2167	33.6125	4.3519

Tab 2.2. *rozmer úlohy* $(n\,x\,m) = (10\,x\,20)$

	počet iterácii Newtonovej metódy			
	2	3	4	5
počet iterácií zlatého rezu	**priemerné vzdialenosti bodu optima *xopt* od vypočítanej aproximácie *x(1)* centrálnej trajektórie *x(mi)***			
5	236.5953	182.4127	109.3841	51.1162
10	236.4109	168.4046	88.4958	35.6447
15	236.5086	167.4232	87.0289	34.5048
20	236.5227	167.3457	86.9291	34.4523
25	236.5240	167.3393	86.9223	34.4495
30	236.5242	167.3388	86.9218	34.4495
35	236.5242	167.3387	86.9217	34.4495
40	236.5242	167.3387	86.9217	34.4495
45	236.5242	167.3387	86.9217	34.4495
50	236.5242	167.3387	86.9217	34.4495

Tab 2.3. *rozmer úlohy* $(n\,x\,m)=(50\,x\,100)$

	počet iterácii Newtonovej metódy			
	2	3	4	5
počet iterácií zlatého rezu	**priemerné vzdialenosti bodu optima *xopt* od vypočítanej aproximácie *x(1)* centrálnej trajektórie *x(mi)***			
5	490.4381	473.9489	363.6580	201.8341
10	499.7109	463.5108	325.5171	147.1674
15	500.5677	462.4909	322.0520	142.7927
20	500.6459	462.4015	321.7448	142.4166
25	500.6530	462.3936	321.7174	142.3846
30	500.6537	462.3929	321.7149	142.3818
35	500.6537	462.3928	321.7147	142.3815
40	500.6537	462.3928	321.7146	142.3819
45	500.6538	462.3926	321.7145	142.3796
50	500.6537	462.3926	321.7147	142.3780

Z tabuliek 2.1. – 2.3. vyplýva, že vplyv počtu iterácii metódy zlatého rezu je pri použití rôzneho počtu iterácií Newtonovej metódy odlišný. Rozlišujeme 2 prípady:

- Počet iterácií Newtonovej metódy sa rovná 2.
  V tomto prípade experimenty ukazujú, že použitie metódy zlatého rezu je zbytočné, dokonca v prípade úloh veľkých rozmerov (n = 50, m = 100) až kontraproduktívne. Otázna však ostáva samotná konvergencia metódy pri použití takéhoto počtu iterácií Newtonovej metódy. Toto bude predmetom nášho ďalšieho testovania.

- Počet iterácií Newtonovej metódy je viac ako 2.
  Výsledky testov naznačujú, že vhodný počet iterácií zlatého rezu je v intervale 25 – 30. Vyšší počet iterácií sa ukazuje byť zbytočný.

Ďalší fakt, ktorý tieto experimenty odkryli je, že pre prvú vonkajšiu iteráciu sa vzdialenosť aproximácie od optimálneho riešenia zväčší a až následne dochádza k približovaniu. Je to spôsobené faktom, že algoritmus sa snaží dostať do „blízkosti“ centrálnej trajektórie. Zdá sa,že čím má úloha lineárneho programovania väčší rozmer, tým je toto počiatočné „oddialenie“ väčšie.

Voľba vhodného počtu iterácií Newtonovej metódy.

Keďže sa nám podarilo určiť vhodný počet iterácií metódy zlatého rezu môžme sa zamerať na pozorovanie dosiahnutej presnosti v jednotlivých iteráciách Newtonovej metódy ešte stále pri použití zafixovaného parametra $\mu = 1$.

Vstupné parametre tabuľky 2.4.:
$\mu = 1$, rozmery úloh $(n\,x\,m)=(10\,x\,20)$, $\rho = 100$, počet iterácií zlatého rezu $= 25$

Tab 2.4. *Dosiahnutá presnosť v závislosti od poč tu Newtonovských iterácií*

iteracia	c*x(l)-c*xopt	T(x(l),mi)	||gradient||	||hessian||	||smer||	lambda	dualita	||y(l)-yopt||	||x(l)-xopt||
l = 1	317 969.44	432 195.57	3 521.62	1.17E-03	166 596.5476	0.0003	Y	37.2684	231.6325
l = 2	114 363.51	228 872.80	3 358.61	9.49E+10	38 624.1362	0.0007	Y	35.5905	237.9784
l = 3	44 273.69	158 999.31	3 048.27	1.75E+13	14 969.9652	0.0017	Y	33.2733	170.6315
l = 4	15 860.51	130 782.22	2 527.50	1.69E+15	6 111.9829	0.0043	Y	32.6200	88.5215
l = 5	4 482.21	119 576.85	1 791.54	3.65E+17	1 610.6272	0.0097	96 / 4	39.6903	35.5940
l = 6	1 140.37	116 339.94	1 960.12	8.71E+19	382.2942	0.0192	45 / 55	67.3903	11.9783
l = 7	286.47	115 540.09	4 088.58	4.21E+21	79.4738	0.0415	13 / 87	110.8999	3.4216
l = 8	80.78	115 357.64	6 903.39	1.70E+23	5.2896	0.1084	3 / 97	165.0125	2.3277
l = 9	52.37	115 334.44	9 133.04	9.14E+23	1.8264	0.3228	N	201.4283	1.8822
l = 10	42.15	115 326.63	11 622.15	3.68E+26	0.4687	1.4563	N	253.8301	1.4508
l = 11	35.72	115 322.11	13 168.82	1.89E+27	0.2126	9.9428	4 / 96	309.2781	1.0885
l = 12	23.40	115 311.17	11 956.36	1.03E+28	0.1943	5.2071	16 / 84	311.5463	0.8205
l = 13	16.92	115 305.87	15 801.50	2.40E+28	0.1262	5.0619	10 / 90	418.9564	0.6688
l = 14	13.26	115 302.71	15 422.46	1.26E+28	0.0776	1.1630	24 / 76	383.8109	0.5323
l = 15	9.68	115 299.72	12 818.37	2.66E+27	0.0526	2.0122	27 / 73	317.2264	0.4569
l = 16	8.40	115 298.71	11 846.62	1.32E+28	0.0266	7.8329	46 / 54	298.6689	0.3889
l = 17	2.75	115 293.50	6 153.37	2.82E+26	0.0079	38.3503	53 / 47	162.9574	0.3819
l = 18	2.42	115 293.20	3 505.15	8.86E+25	0.0112	271.2015	61 / 39	96.5893	0.3519
l = 19	1.85	115 292.65	1 104.61	1.33E+22	0.0042	50.7962	68 / 32	33.2232	0.3495
l = 20	1.78	115 292.59	858.23	5.92E+21	0.0039	152.4763	71 / 29	25.0438	0.3421

Z výsledkov v tabuľke 2.4. vidíme monotónnosť funkčnej hodnoty účelovej funkcie $c^T x$ úlohy lineárneho programovania ako aj transformačnej funkcie $T(x,\mu)$. Túto vlastnosť metódy nám zabezpečuje algoritmus metódy zlatého rezu. Ďalej z výsledku experimentov vidno, že pri rýchlom približovaní aproximácie k bodu centrálnej trajektórie dochádza k nárastu normy Hessovej matice. V situácii keď začne rýchlosť konvergencie $x^l(\mu) \to \hat{x}(\mu)$ klesať nastáva naopak mierny pokles normy Hessovej matice.

Nakoľko sme v teoretickej časti hovorili o odporúčaní redukcie dĺžky kroku, experimenty ukázali, že táto redukcia nastáva v priemere do 9 iterácie. Pri vyššom počte vnútorných iterácií je výpočet optimálnej dĺžky kroku zbytočný.

Keďže pri tejto metóde používame len odhad prípustného riešenia duálnej úlohy, ktorý nemusí byť skutočne prípustným riešením, stĺpec *dualita* nám hovorí o početnosti prípadov zachovania platnosti slabej vety o dualite, teda o tom, či odhad y je resp. nie je prípustným riešením. Experimenty ukázali, že ešte pri použití 4 vnútorných iterácií je odhad duálnej premennej y prípustným riešením úlohy (D) vo všetkých testovaných prípadoch.

Experimenty boli robené pri fixovanom μ, teda je nemožné dosiahnuť optimálne riešenie $\hat{x}$. Ale aj výsledky pre fixované μ nám ukazujú konvergenciu k bodu centrálnej trajektórie. Takisto ako v predchádzajúcom experimente si môžeme všimnúť počiatočné „oddialenie“ od optimálneho riešenia, nakoľko vzdialenosť štartovacieho bodu od optimálneho riešenia bola vo všetkých prípadoch 100 jednotiek.

Keďže výsledok experimentu nám ukázal, že správanie algoritmu je v poriadku po 4 vnútornú iteráciu, v nasledujúcom experimente otestujeme správanie sa algoritmu pri použití klesajúcej postupnosti $\{\mu_k\}_{k=1}^{\infty}$ pre rôzne počty vnútorných iterácii, maximálne však 4. Ako štartovací parameter sme použili $\mu = 1$ a nový bariérový parameter počítali vzťahom $\mu_{k+1} = \frac{\mu_k}{\sigma}$.

Spoločné vstupné parametre tabuliek 2.4. – 2.8.:
$\mu_1 = 1$, rozmery úloh $(n\,x\,m) = (10\,x\,20)$, $\rho = 100$, počet iterácií zlatého rezu $= 25$, $\sigma = 10$ je redukčný parameter bariérového parametra μ

Tab 2.5. *Dosiahnutá presnosť pri zmene* μ *po* 2 *Newtonovských iteráciách*

mi	iteracia	c*x(l)-c*xopt	T(x(l),mi)	||gradient||	||hessian||	||smer||	lambda	dualita	||y(l)-yopt||	||x(l)-xopt||
1	l = 1	320 671	437 175	3 553	1.17E-03	171 659	0.000338	Y	37.4665	232.8653
	l = 2	115 034	231 823	3 390	9.59E+10	39 003	0.000744	Y	35.7721	236.5240
1.E-01	l = 3	44 730	162 273	3 511	2.04E+11	155 769	0.000175	Y	36.9725	166.8621
	l = 4	15 704	133 267	3 458	1.62E+13	61 154	0.000429	Y	36.5652	86.1671
1.E-02	l = 5	4 383	121 984	3 539	4.12E+13	161 978	0.000094	Y	37.3074	33.7679
	l = 6	1 097	118 698	3 511	2.44E+16	36 912	0.000178	Y	37.0510	11.0909
1.E-03	l = 7	290	117 892	3 549	7.12E+15	75 172	0.000036	Y	37.3959	3.5854
	l = 8	104	117 706	3 538	1.57E+18	5 929	0.000083	Y	37.2996	2.5377
1.E-04	l = 9	72	117 675	3 554	2.26E+19	21 977	0.000020	Y	37.4407	2.1290
	l = 10	61	117 664	3 545	1.30E+22	4 808	0.000048	Y	37.3343	1.9161
1.E-05	l = 11	57	117 659	3 556	2.09E+22	39 662	0.000012	Y	37.4506	1.5361
	l = 12	50	117 652	3 550	7.39E+25	40 067	0.000030	Y	37.3875	1.3599
1.E-06	l = 13	39	117 641	3 557	1.30E+25	197 504	0.000007	Y	37.4696	1.0558
	l = 14	30	117 633	3 555	4.77E+26	283 201	0.000016	Y	37.4433	0.7876
1.E-07	l = 15	25	117 628	3 558	2.35E+27	612 465	0.000004	Y	37.4813	0.6252
	l = 16	22	117 625	3 557	1.24E+30	1 189 060	0.000010	Y	37.4672	0.5030

Tab 2.6. *Dosiahnutá presnosť pri zmene* μ *po* 3 *Newtonovských iteráciách*

mi	iteracia	c*x(l)-c*xopt	T(x(l),mi)	||gradient||	||hessian||	||smer||	lambda	dualita	||y(l)-yopt||	||x(l)-xopt||
1	l = 1	320 671	437 175	3 553	1.17E-03	171 659	0.000338	Y	37.4665	232.8653
	l = 2	115 034	231 823	3 390	9.59E+10	39 003	0.000744	Y	35.7721	236.5240
	l = 3	44 730	161 736	3 074	2.04E+13	15 500	0.001753	Y	33.5283	167.3393
1.E-01	l = 4	15 749	133 312	3 458	1.62E+13	61 406	0.000428	Y	36.5660	86.3960
	l = 5	4 395	121 975	3 356	4.07E+15	16 221	0.000940	Y	35.8233	33.8832
	l = 6	1 101	118 691	3 084	2.19E+18	3 693	0.001786	Y	34.3305	11.1408
1.E-02	l = 7	291	117 892	3 459	6.91E+17	7 566	0.000355	Y	36.5823	3.5899
	l = 8	104	117 706	3 349	1.55E+20	594	0.000830	Y	35.8265	2.5398
	l = 9	72	117 674	3 045	5.79E+23	220	0.002002	Y	35.1533	2.1308
1.E-03	l = 10	61	117 664	3 422	1.26E+24	490	0.000483	Y	36.0680	1.9104
	l = 11	57	117 659	3 204	1.97E+26	393	0.001217	Y	34.3899	1.5355
	l = 12	50	117 652	2 753	7.40E+29	371	0.002988	97 / 3	35.0395	1.3621
1.E-04	l = 13	39	117 641	3 367	3.49E+28	1 984	0.000703	Y	35.6323	1.0549
	l = 14	30	117 633	3 107	4.67E+30	2 737	0.001619	Y	34.4889	0.7870
	l = 15	25	117 627	2 687	8.01E+33	521	0.003785	96 / 4	37.2790	0.6209
1.E-05	l = 16	22	117 625	3 356	2.13E+33	11 362	0.000930	Y	35.8227	0.4767

Tab 2.7. *Dosiahnutá presnosť pri zmene* μ *po* 4 *Newtonovských iteráciách*

mi	iteracia	c*x(l)-c*xopt	T(x(l),mi)	||gradient||	||hessian||	||smer||	lambda	dualita	||y(l)-yopt||	||x(l)-xopt||
1	l = 1	320 671	437 175	3 553	1.17E-03	171 659	0.000338	Y	37.4665	232.8653
	l = 2	115 034	231 823	3 390	9.59E+10	39 003	0.000744	Y	35.7721	236.5240
	l = 3	44 730	161 736	3 074	2.04E+13	15 500	0.001753	Y	33.5283	167.3393
	l = 4	15 749	132 955	2 562	1.62E+15	6 079	0.004319	Y	32.9566	86.9223
1.E-01	l = 5	4 422	122 002	3 358	3.90E+15	16 362	0.000936	Y	35.8297	34.0519
	l = 6	1 106	118 697	3 088	1.68E+18	3 719	0.001782	Y	34.3232	11.1919
	l = 7	292	117 887	2 598	6.64E+19	756	0.003583	99 / 1	34.3384	3.5934
	l = 8	104	117 702	1 945	4.05E+21	59	0.008632	93 / 7	39.6931	2.5451
1.E-02	l = 9	72	117 675	3 201	1.53E+23	221	0.002025	Y	35.2153	2.1347
	l = 10	61	117 664	2 565	1.06E+26	51	0.004861	98 / 2	39.6377	1.9099
	l = 11	57	117 659	2 193	1.71E+28	40	0.012538	76 / 24	60.3808	1.5310
	l = 12	49	117 652	4 326	6.19E+31	33	0.031817	36 / 24	120.3386	1.3556
1.E-03	l = 13	39	117 641	2 706	2.61E+30	251	0.006717	93 / 7	39.5546	1.0474
	l = 14	30	117 633	2 398	2.60E+32	277	0.015279	71 / 29	66.5199	0.7727
	l = 15	25	117 627	3 988	6.27E+34	55	0.038158	41 / 59	121.1221	0.6110
	l = 16	22	117 625	7 576	4.82E+35	156	0.102936	17 / 83	193.4664	0.4661

Tabuľky 2.5. – 2.7. nám ukazujú výsledky týchto experimentov pre jednotlivé počty použitých vnútorných iterácií. Z experimentov vyplýva, že zvýšením počtu vnútorných iterácií z 2 na 3 dostávame po 16 iteráciách v priemere o 5% presnejšiu aproximáciu optimálneho riešenia $\hat{x}$. Pri zvýšení na 4 je toto zlepšenie 7%. Ďalším záverom je, že pri použití 4 vnútorných iterácií dochádza k situácii, keď odhad prípustného riešenia duálnej úlohy nemusí byť skutočne prípustným riešením. S tým je úzko spojené aj „oddialenie“ od optimálneho riešenia duálnej úlohy.

Tab 2.8. *Početnosť prípadov v jednotlivých iteráciách*

počet Newton. iterácii = 2			počet Newton. iterácii = 3			počet Newton. iterácii = 4		
poradie iterácie	mi	počet prípadov	poradie iterácie	mi	počet prípadov	poradie iterácie	mi	počet prípadov
1	1.E+00	100	**1**	1.E+00	100	**1**	1.E+00	100
2		100	**2**		100	**2**		100
3	1.E-01	100	**3**		100	**3**		100
4		100	**4**	1.E-01	100	**4**		100
5	1.E-02	100	**5**		100	**5**	1.E-01	100
6		100	**6**		100	**6**		100
7	1.E-03	100	**7**	1.E-02	100	**7**		100
8		100	**8**		100	**8**		100
9	1.E-04	98	**9**		98	**9**	1.E-02	99
10		86	**10**	1.E-03	86	**10**		86
11	1.E-05	75	**11**		74	**11**		76
12		66	**12**		64	**12**		65
13	1.E-06	61	**13**	1.E-04	58	**13**	1.E-03	58
14		56	**14**		53	**14**		55
15	1.E-07	50	**15**		45	**15**		47
16		38	**16**	1.E-05	34	**16**		36
17	1.E-08	32	**17**		23	**17**	1.E-04	28
18		26	**18**		19	**18**		23
19	1.E-09	24	**19**	1.E-06	14	**19**		16
20		18	**20**		10	**20**		10
21	1.E-10	14	**21**		6	**21**	1.E-05	8
22		9	**22**	1.E-07	5	**22**		6
23	1.E-11	6	**23**		4	**23**		6
24		3	**24**		2	**24**		5
25	1.E-12	3	**25**	1.E-08	1	**25**	1.E-06	5
26		2	**26**		1	**26**		1
27	1.E-13	2						
28		1						

Pri experimente zobrazenom v tabuľke 2.8. sme použili ako kritérium presnosti riešenia vzťah $\|x_i(\mu_k) - \hat{x}\| < \varepsilon$, pričom $\varepsilon = 0{,}001$. Následne sme zaznamenali početnosť prípadov, v ktorých sa algoritmus dostal do i - tej iterácie. Z tohto pohľadu ako aj z predchádzajúcich zistení sa javí použitie 3 vnútorných iterácií ako najvhodnejšie.

Porovnanie výpočtovej zložitosti Newtonovského smeru a optimálneho kroku.

Na základe hore uvedených experimentov môžeme v algoritme primárnej odmocninovej metódy použiť 25 – 30 krokov zlatého zlatého rezu a 3 iterácie Newtonovej metódy. Teraz sa na tieto parametre pozrieme z pohľadu ich výpočtovej zložitosti.

- Výpočet smeru.
  Na určení smeru sa podieľa výpočet funkčnej hodnoty transformačnej funkcie, jej gradientu, Hessovej matice a Q-R rozklad metódou Hansona a Lawsona. Pri stanovení výpočtovej zložitosti jednotlivých funkcií sme sa zaoberali operáciami násobenia a odmocňovania, z jednak z dôvodu ich náročnosti a na druhej strane z dôvodu početnosti ich použitia. Výpočtová zložitosť jednotlivých funkcií je nasledujúca:

  1. $T(x,\mu)$ - operácie násobenia: $mn + n + 1$
     - operácie odmocňovania: m
  2. $\nabla T(x,\mu)$ - operácie násobenia: $2mn + m$
     - operácie odmocňovania: m
  3. $\nabla^2 T(x,\mu)$ - operácie násobenia: $mn^2 + 2mn + m$
     - operácie odmocňovania: m
  4. Q-R rozklad - operácie násobenia: $n^3 + 3.5n^2 + 2n + 1.5$
     - operácie odmocňovania: n

Teda výsledná výpočtová zložitosť Newtonovského smeru je pre operáciu násobenia $mn^2 + n^3 + 5mn + 3.5n^2 + 3m + 3n + 2.5$ a pre operáciu odmocňovania $3m + n$.

- Výpočet kroku.
  Na určenie kroku použijeme algoritmus zlatého rezu, ktorý má pre operáciu násobenie výpočtovú zložitosť $(k+2)mn + (k+1)m + (k+3)n + k + 2$, kde k znamená počet iterácií zlatého rezu. Pre operáciu odmocňovania je zložitosť rovná $(k+1)m + 1$.

Môžme povedať, že hlavný člen výpočtovej zložitosti operácie násobenie pre Newtonovský smer je $n^2(n+m)$ a pre krok $k(mn+m+n)$. To znamená, že pre úlohy lineárneho programovania malých rozmerov počet iterácii zlatého rezu výrazne ovplyvňuje výpočtovú zložitosť algoritmu, kým pre úlohu veľkých rozmerov jeho vplyv geometricky slabne. Podobný záver vyplýva aj z porovnania operácie odmocňovania, pričom použitie operácie násobenia je omnoho početnejšie.

Citlivosť metódy na voľbu postupnosti bariérového parametra μ.

V tejto časti sa budeme venovať spôsobu nastavenia počiatočného parametra μ_1, ako aj experimentálnemu testovaniu rôznych stratégii klesajúcej postupnosti $\{\mu_k\}_{k=1}^{\infty}$. Ako sme už uviedli, v praxi sa počiatočný parameter volí vzťahom $\mu_1 = \frac{1}{m}\sum_{i=1}^{m}\sqrt{4\left(y_i^0\right)^2\left(Ax^0 - b\right)_i}$. Pre postupnosť bariérového parametra môžme otestovať 2 druhy stratégie. Prvou je výpočet tohto parametra v každej iterácii pomocou metódy najmenších štvorcov a druhov určenie kladného parametra σ, pomocou ktorého by sme bariérový parameter vzťahom $\mu_{k+1} = \frac{\mu_k}{\sigma}$ v každej vonkajšej iterácii menili. Pričom počet iterácii zlatého rezu sme nastavili 25 a počet vnútorných iterácii 3.

1. stratégia.

Vstupné parametre tabuľky 2.9.:
$\mu_k = \frac{1}{m}\sum_{i=1}^{m}\sqrt{4\left(y_i^k\right)^2\left(Ax^k - b\right)_i}$,rozmery úloh $(n\,x\,m) = (10\,x\,20)$, $\rho = 100$, počet iterácií zlatého rezu $= 25$, počet Newtonovských iterácií $= 3$

Tab 2.9. *Dosiahnutá presnosť v závislosti od parametra* μ

mi	iteracia	c*x(l)-c*xopt	T(x(l),mi)	||gradient||	||hessian||	||smer||	lambda	dualita	||y(l)-yopt||	||x(l)-xopt||
2702.9156	l = 1	320671	-2532153	12938.1480	8559.4040	70.8246	21.9160	14 / 86	93.2145	6790.9888
	l = 2	8685765	-8461272	719.0673	0.4875	10607.2678	0.2912	Y	27.8908	16902.5768
	l = 3	5460989	-10996852	1191.9620	5137.4615	8516.2176	0.8678	Y	36.4118	40190.8800
3041.1005	l = 4	6055814	-16043882	3536.2259	1694711.5516	13139.2389	3.5772	Y	69.5611	289237.1090
	l = 5	20719838	-24744672	2397.2922	4210478557.3940	62380.5835	1.4595	Y	70.8786	486862.8800
	l = 6	24303031	-28044048	1962.9862	791330995.6702	44204.2152	1.3379	Y	57.8805	731466.5730
3287.1961	l = 7	28432623	-39606770	1400.5680	8229435.5504	78780.9118	1.2279	Y	44.0418	1800842.1780
	l = 8	45543705	-45384008	419.9053	29581.8445	5406.9223	248.8576	Y	28.3901	1814395.6910
	l = 9	45152671	-45533445	159.0373	571.0818	9422.8841	13.4167	Y	23.2474	1856726.3130
3323.7340	l = 10	45738014	-49575519	85.5063	40.9640	68968.6526	1.4666	Y	21.8875	2172713.1890
	l = 11	50055960	-49821241	22.5832	28.2093	3719.7689	951.7064	Y	21.2252	2150516.5990
	l = 12	49846758	-49832123	1.7012	27.3965	2188.7676	297.3657	Y	21.0120	2158335.7310
3324.3762	l = 13	49948933	-49889996	0.5653	27.4033	659.8244	20.2360	Y	21.0210	2161308.2230
	l = 14	50007175	-49890086	0.0014	27.3969	0.0888	813.8627	Y	21.0188	2161306.9310
	l = 15	50007173	-49890086	0.0041	27.3969	0.3136	770.4550	Y	21.0188	2161308.2130
3324.3755	l = 16	50007176	-49889981	0.0026	27.3968	1.2487	249.5943	Y	21.0188	2161303.3220
	l = 17	50007101	-49889981	0.0013	27.3969	0.4204	231711.7232	Y	21.0188	2161301.0600
	l = 18	50007052	-49889981	0.0009	27.3969	0.2802	298326.1299	Y	21.0188	2161302.1540
3324.3757	l = 19	50007061	-49890013	0.0015	27.3969	0.3322	458.6639	Y	21.0188	2161302.6310
	l = 20	50007088	-49890013	0.0005	27.3969	0.0396	695.3678	Y	21.0188	2161300.0950

Z tabuľky 2.9. vyplýva, že pri použitý prvej stratégie nedochádza ku konvergencii k optimálnemu riešeniu $\hat{x}$ a bariérový parameter je natoľko veľký, že aproximácia optimálneho riešenia je „veľmi ďaleko“ od optimálneho riešenia. Pri testovaní tejto stratégie sme ani v jednom prípade nedosiahli dostatočne presnú aproximáciu minima ani po 25 vonkajších iteráciách. Všimnúť si môžme aj fakt. že v prvej iterácii dochádza v 86 % prípadov k porušeniu slabej vety o dualite. Na základe týchto dôvodov v ďalších experimentoch odporúčame počiatočný bariérový parameter zmenšiť niekoľko násobne (napr. 100 krát, 1000 krát, ...)

2.stratégia.

Spoločné vstupné parametre pre tabuľky 2.10. – 2.12.:

$\mu_1 = \frac{1}{m}\sum_{i=1}^{m}\sqrt{4\left(y_i^0\right)^2\left(Ax^0 - b\right)_i}$, rozmery úloh $(n\,x\,m) = (10\,x\,20)$, $\rho = 100$, počet iterácií zlatého rezu $= 25$, počet Newtonovských iterácií $= 3$

Pri použití 2 stratégie sme vypočítaný počiatočný bariérový parameter zmenšili vždy 1000 krát. Redukčný parameter σ sme postupne menili. Výsledky experimentu sú zhrnuté v tabuľkách 2.10. a 2.11..

Tab 2.10. *Dosiahnutá presnosť v závislosti od redukčného parametra* σ

	redukčný parameter σ								
	2			3			4		
iteracia	mi	dualita	‖x(l)-xopt‖	mi	dualita	‖x(l)-xopt‖	mi	dualita	‖x(l)-xopt‖
l = 1		Y	233.1311		Y	233.1311		Y	233.1311
l = 2	2.70E+00	Y	237.3694	2.70E+00	Y	237.3694	2.70E+00	Y	237.3694
l = 3		Y	168.8038		Y	168.8038		Y	168.8038
l = 4		Y	87.8278		Y	87.5660		Y	87.4357
l = 5	1.35E+00	81 / 19	34.9111	9.01E-01	96 / 4	34.6245	6.76E-01	99 / 1	34.4827
l = 6		26 / 74	11.6852		52 / 48	11.5112		71 / 29	11.4267
l = 7		50 / 50	3.6792		88 / 12	3.6407		97 / 3	3.6248
l = 8	6.76E-01	27 / 73	2.5835	3.00E-01	58 / 42	2.5630	1.69E-01	81 / 19	2.5550
l = 9		10 / 90	2.1429		33 / 67	2.1435		49 / 51	2.1423
l = 10		47 / 53	1.8635		71 / 29	1.8866		83 / 17	1.8944
l = 11	3.38E-01	13 / 87	1.5186	1.00E-01	36 / 64	1.5299	4.22E-02	51 / 49	1.5314
l = 12		N	1.1532		14 / 86	1.3234		21 / 79	1.3509
l = 13		39 / 61	0.7022		42 / 58	0.9389		53 / 47	0.9957
l = 14	1.69E-01	22 / 78	0.5018	3.34E-02	18 / 82	0.6547	1.06E-02	28 / 72	0.7385
l = 15		6 / 94	0.3444		2 / 98	0.4017		14 / 86	0.5204
l = 16		45 / 55	0.2710		40 / 60	0.3726		45 / 55	0.3997
l = 17	8.45E-02	27 / 73	0.2258	1.11E-02	21 / 79	0.3150	2.64E-03	28 / 72	0.3410
l = 18		5 / 95	0.1486		2 / 98	0.2206		6 / 94	0.2719
l = 19	4.22E-02	63 / 37	0.1175	3.71E-03	58 / 42	0.1681	6.60E-04	57 / 43	0.2099
l = 20		30 / 70	0.0970		27 / 73	0.1506		40 / 60	0.1902

Tab 2.11. *Dosiahnutá presnosť v závislosti od redukčného parametra* σ

	redukčný parameter σ								
	10			15			30		
iteracia	mi	dualita	||x(l)-xopt||	mi	dualita	||x(l)-xopt||	mi	dualita	||x(l)-xopt||
l = 1		Y	232.0382		Y	233.6952		Y	238.7478
l = 2	2.71E+00	Y	237.3089	2.71E+00	Y	242.7858	2.70E+00	Y	250.2574
l = 3		Y	167.2679		Y	173.9524		Y	181.0950
l = 4		Y	84.3376		Y	89.6792		Y	93.2572
l = 5	2.71E-01	Y	32.6663	1.81E-01	Y	35.4997	9.00E-02	Y	36.3132
l = 6		99 / 1	10.0350		Y	11.6987		Y	13.1835
l = 7		Y	3.6649		Y	4.4499		Y	4.5969
l = 8	2.71E-02	Y	2.6182	1.20E-02	Y	3.2645	3.00E-03	Y	3.3338
l = 9		97 / 3	2.2108		Y	2.7072		Y	2.7817
l = 10		Y	1.9831		Y	2.4162		Y	2.4834
l = 11	2.71E-03	99 / 1	1.6081	8.02E-04	Y	1.9799	1.00E-04	Y	2.0375
l = 12		82 / 18	1.4038		98 / 2	1.7751		Y	1.8303
l = 13		Y	1.0852		Y	1.3802		Y	1.4346
l = 14	2.71E-04	94 / 6	0.7975	5.35E-05	Y	1.0256	3.33E-06	Y	1.0677
l = 15		88 / 12	0.6451		Y	0.8640		Y	0.8819
l = 16		Y	0.5396		Y	0.7035		Y	0.7175
l = 17	2.71E-05	97 / 3	0.4982	3.57E-06	Y	0.6520	1.11E-07	Y	0.6475
l = 18		84 / 16	0.3396		Y	0.5600		Y	0.5936
l = 19	2.71E-06	Y	0.2227	2.38E-07	Y	0.3895	3.70E-09	Y	0.4631
l = 20		94 / 6	0.1971		Y	0.2759		Y	0.4223

Z experimentu vyplynulo, že čím nižší je redukčný parameter σ, tým lepšia je po 20 iteráciách výsledná presnosť. Na druhej strane čím je tento parameter väčší, tým menej dochádza k porušeniam slabej vety o dualite. Ako ukazuje tabuľka 2.12. pri použití redukčného parametra viac ako10 a menej ako 30 metóda konverguje najrýchlejšie.

Tab 2.12. *Početnosť prípadov v závislosti od parametra* σ

	redukčný parameter σ					
	2	3	4	10	15	30
iteracia	Počet prípadov v %					
1	100	100	100	100	100	100
2	100	100	100	100	100	100
3	100	100	100	100	100	100
4	100	100	100	100	100	100
5	100	100	100	100	100	100
6	100	100	100	100	100	100
7	100	100	100	100	100	100
8	100	100	99	100	100	100
9	99	100	99	98	99	99
10	97	90	86	84	83	86
11	88	78	76	73	72	75
12	83	69	66	64	63	65
13	80	65	59	57	57	59
14	69	56	52	52	53	55
15	61	48	47	43	45	48
16	49	31	37	32	34	36
17	33	26	27	24	27	29
18	26	20	19	18	21	23
19	24	15	14	15	17	16
20	15	13	13	11	11	13
21	13	12	12	7	7	10
22	10	10	6	5	5	6
23	5	7	4	5	1	5
24	5	6	3	3	1	4
25	3	5	3	1	1	3

Zhrnutie – závery numerických experimentov

- Aproximácia optimálneho riešenia sa v prvej vonkajšej iterácii vzdiali od optimálneho riešenia $\hat{x}$.
- Ako vhodný počet iterácii metódy zlatého rezu sa javí počet v intervale 25 – 30. Použitie väčšieho počtu je zbytočné.
- Monotónnosť funkčnej hodnoty účelovej funkcie $c^T x$ ako aj transformačnej funkcie $T(x, \mu)$.
- Časté zlyhanie odhadu duálnej premennej y v podmienke prípustnosti pri použití viac ako 3 vnútorných iterácií.
- Ako vhodný počet Newtonovských iterácií sa javí počet $= 3$.
- S rastom rozmerov úlohy lineárneho programovania (P) klesá vplyv zvoleného počtu iterácií metódy zlatého rezu na celkovej výpočtovej zložitosti algoritmu.
- Potreba redukcie štartovacieho bariérového parametra μ_1 za účelom zrýchlenia konvergencii algoritmu.
- S rastom redukčného parametra σ je odhad duálnej premennej y spoľahlivejší.

  Najrýchlejšiu konvergenciu k optimálnemu riešeniu $\hat{x}$ ukázali experimenty pri voľbe σ v rozmezí 10 – 30.

3. Primárno-duálna odmocninová metóda riešenia úlohy lineárneho programovania.

3.1. Popis metódy.

Pri primárno-duálnej metóde taktiež budeme hľadať minimum úlohy

$$Min\left\{c^T x \mid Ax \geq b\right\} \qquad (P)$$

Primárnu úlohu lineárneho programovania upravíme o pomocnú premennú $z \in R_+^m$:

$$Min\left\{c^T x \mid Ax - z = b,\ z \geq 0\right\} \qquad (P^*)$$

Tento tvar je ekvivalentný z úlohou (P). K primárnej úlohe (P^*) máme úlohu duálnu:

$$Max\left\{b^T y \mid A^T y = c,\ y \geq 0\right\} \qquad (D^*)$$

Ako transformačnú funkciu použijeme odmocninovú transformačnú funkciu tvaru:

$$T(x,z,\mu) = c^T x - \mu \sum_{i=1}^{m} \sqrt{z_i}\ , \qquad Ax - z = b,\ \mu > 0 \qquad (3.1)$$

Teda namiesto riešenia úlohy (P^*) budeme riešiť postupnosť transformačných úloh:

$$\underset{x \in R^n, z \in R^m}{Min} \left\{ T(x,z,\mu) = c^T x - \mu \sum_{i=1}^{m} \sqrt{z_i} \mid x \in K^0, z > 0_m \right\} \qquad (T_\mu^*)$$

Vonkajšími iteráciami nazveme opakované riešenie týchto úloh pre zmenšujúci sa parameter $\mu_k \downarrow 0^+$. Spôsob voľby klesajúcej postupnosti popíšeme nižšie. Tak ako bolo uvedené v predchádzajúcej kapitole, v skutočnosti ani nepotrebujeme nájsť optimálny bod $(\hat{x}(\mu_k), \hat{y}(\mu_k), \hat{z}(\mu_k))$ presne, postačí nám jeho „dostatočne presná“ aproximácia. Na hľadanie tejto aproximácie použijeme modifikovanú Newtonovu metódu na riešenie sústav nelineárnych rovníc vysvetlenú v kapitole 1.2.. Na reguláciu dĺžky kroku využijeme algoritmus metódy zlatého rezu, pomocou ktorého získame optimálnu dĺžku kroku. Body získané počas iteračného procesu modifikovanej Newtonovej metódy sú spresňujúcimi aproximáciami hľadaného optima $(\hat{x}(\mu_k), \hat{y}(\mu_k), \hat{z}(\mu_k))$. Teda prechod od bodu $(x_i(\mu_k), y_i(\mu_k), z_i(\mu_k))$ do bodu $(x_{i+1}(\mu_k), y_{i+1}(\mu_k), z_{i+1}(\mu_k))$ nazveme vnútornou iteráciou metódy.

Ako štartovací bod $(x^0(\mu_k), y^0(\mu_k), z^0(\mu_k))$ pre každú vnútornú iteráciu zvolíme poslednú aproximáciu získanú v predošlej vonkajšej iterácii s výnimkou prvej vonkajšej iterácie. Spôsob voľby štartovacieho bodu modifikovanej Newtonovej iterácii v prvej vonkajšej iterácii bližšie nešpecifikujeme, predpokladáme že prípustný štartovací bod je zadaný na vstupe.

3.2. Algoritmus primárno - duálnej odmocninovej metódy.

V tejto podkapitole detailne popíšeme algoritmus Primárno – duálnej „odmocninovej“ metódy použitý v diplomovej práci. Pomocou generátora úlohy lineárneho programovania s vopred zadaným optimálnym riešením vytvoríme primárnu úlohu (P^*), duálnu úlohu (D^*), odmocninovú transformačnú funkciu (T_μ^*) a štartovací bod $(x^0, y^0, z^0) \in R^{n+2m}$.

Otvorenou otázkou ostáva voľba štartovacieho bariérového parametra $\mu_1 > 0$ ako aj klesajúcej postupnosti bariérového parametra $\{\mu_k\}_{k=1}^{\infty}$. Z „perturbovaných“ Kuhn - Tuckerových podmienok $(1.12.)$ a z definície doplnkovej premennej vieme, že $2y_i\sqrt{z_i} = \mu$,

teda $2y_i^0\sqrt{z_i^0} \approx \mu \quad i = 1,2,...,m$. Štartovaciu hodnotu určíme metódou najmenších štvorcov pre m rovníc a jednu neznámu.

$$\begin{array}{cccc} 1 & \mu & = & 2y_1^0\sqrt{z_1^0} \\ 1 & \mu & = & 2y_2^0\sqrt{z_2^0} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & \mu & = & 2y_m^0\sqrt{z_m^0} \end{array} = h \in R^m$$

Z čoho vyplýva:

$$e\mu \approx h \quad / e^T$$
$$e^T e\mu = e^T h$$
$$\overline{\mu}_1 = \frac{e^T h}{e^T e} = \frac{e^T h}{m}$$

Riešením $\overline{\mu}_1$ je „aritmetický priemer“ pravých strán $h_i = 2y_i^0\sqrt{z_i^0}$. Nakoniec označíme

$$\mu_1 = \overline{\mu}_1 = \frac{1}{m}\sum_{i=1}^{m} 2y_i^0\sqrt{z_i^0}\ . \qquad (3.2.)$$

Pri voľbe postupnosti tohto bariérového parametra môžme uvažovať dvoma spôsobmi. Prvým spôsobom je zmena na základe nami určeného parametra, t.j. $\mu_{k+1} = \frac{\mu_k}{\sigma} \quad \sigma > 1$. Druhým spôsobom je opätovný výpočet aritmetického priemeru $\overline{\mu}_{k+1}$.

Po určení bariérového parametra μ môžeme pristúpiť k vnútorným iteráciám. Tú zabezpečuje modifikovaná Newtonova metóda na riešenie sústav nelineárnych rovníc. Úlohu lineárneho programovania transformujeme do tvaru popísanom v kapitole 1.2., konkrétne na tvar (1.5). Teda namiesto sústavy nelineárnych rovníc budeme riešiť lineárnu sústavu $F(w^0) + J(w^0)\Delta w = 0_N$.

- Vytvorenie sústavy lineárnych rovníc.

K úlohe (T_μ^*) môžeme napísať Lagrangeove podmienky optimality. Z konvexnosti funkcií tvoriacich úlohu (T_μ^*) vyplýva, že ide o nutné a zároveň postačujúce podmienky. Lagrangeova funkcia pre úlohu (T_μ^*) má tvar

$$L(x,z,\mu,y) = c^T x - \mu\sum_{i=1}^{m}\sqrt{z_i} - y^T(Ax - z - b), \qquad z \geq 0, y \geq 0$$

a jej derivovaním podľa jednotlivých premenných dostávame Lagrangeove podmienky

$$\nabla_x L = c - A^T y = 0$$
$$\nabla_y L = Ax - z - b = 0, \quad y \geq 0$$
$$\nabla_{z_i} L = -\mu\frac{1}{2}\frac{1}{\sqrt{z_i}} + y_i = 0, \quad z \geq 0$$

Úpravou dostávame sústavu nelineárnych rovníc

$$A^T y = c \qquad y \geq 0$$
$$Ax - z = b \quad z \geq 0\,.$$
$$2y_i\sqrt{z_i} = \mu$$

Teraz si môžeme zadefinovať vektor $F\left(w^0\right)$:

$$F\left(w^0\right) = \begin{pmatrix} \left(Ax^0 - z^0 - b\right) \\ \left(A^T y^0 - c\right) \\ \left(diag\left(2y_i^0\sqrt{z_i^0} - \mu\right) * e\right) \end{pmatrix} = \begin{pmatrix} r_b \\ r_c \\ r_\mu \end{pmatrix},$$

kde $e^T = (1,1,\ldots,1)$. Parciálnym derivovaním tohto vektora dostávame maticu $J\left(w^0\right)$, ktorej blokový zápis vyzerá následovne:

$$J\left(w^0\right) = \begin{pmatrix} A & -I & 0 \\ 0 & 0 & A^T \\ 0 & diag\left(\dfrac{y_i^0}{\sqrt{z_i^0}}\right) & diag\left(2\sqrt{z_i^0}\right) \end{pmatrix}.$$

Môžeme pristúpiť k riešeniu sústavy $J\left(w^0\right)\Delta w = -F\left(w^0\right)$.

- Riešenie sústavy $(2m+n)x(2m+n)$:

$$\begin{aligned} A\Delta x - I\Delta z &= -r_b && (1) \\ A^T\Delta y &= -r_c && (2), \\ D_1\Delta z + D_2\Delta y &= -r_\mu && (3) \end{aligned}$$

kde $D_1 = diag\left(\dfrac{y_i^0}{\sqrt{z_i^0}}\right)$ a $D_2 = diag\left(2\sqrt{z_i^0}\right)$ pre $i = 1,\ldots,m$.

- Eliminujeme Δz :

Rovnicu (1) vynásobíme zľava $D_1 \Rightarrow (1a)$:

$$\begin{aligned} D_1 A\Delta x - D_1\Delta z &= -D_1 r_b && (1a) \\ D_1\Delta z + D_2\Delta y &= -r_\mu && (3) \end{aligned}$$ sčítame.

Ďalej:

$$\begin{aligned} D_1 A\Delta x + D_2\Delta y &= -r_\mu + D_1 r_b && (4) / A^T D_2^{-1}\ zľava \\ A^T\Delta y &= -r_c && (3) \end{aligned}.$$

Dostávame rovnicu:

$$\left(A^T D_2^{-1} D_1 A\right)\Delta x = A^T D_2^{-1}\left(-r\mu + D_1 r_b\right) + r_c \qquad \left(= r \in R^n\right) \qquad (3.3.)$$

- Riešením sústavy $(3.3.)$ rozmerov $(n\,x\,n)$ metódou Q – R rozkladu dostávame Δx .

- Z rovnice (1) vypočítame $\Delta z = A\Delta + r_b$.
- Z rovnice (3) vypočítame $\Delta y = D_2^{-1}\left(-r_\mu - D_1\Delta z\right)$.

- Výpočet optimálnej dĺžky kroku λ.

Najprv vypočítame maximálnu prípustnú dĺžku kroku $\bar{\lambda} > 0$ a potom metódou zlatého rezu približne riešime úlohu

$$Min\left\{\varphi(\lambda) \overset{def}{=} T\left(x^k + \lambda\Delta x, z^k + \lambda\Delta z, \mu\right) = c^T x + \lambda c^T \Delta x - \mu\sum_{i=1}^{m}\sqrt{z_i^k + \lambda\Delta z_i}\right\}$$

takým istým spôsobom ako v prípade primárnej odmocninovej metódy.
Na výpočet optimálnej dĺžky kroku použijeme nasledovné naprogramované funkcie:

- Funkcia Txmi2 (z, c, x, mi), ktorá počíta funkčnú hodnotu transformačnej funkcie $T\left(x, z, \mu\right) = c^T x - \mu\sum_{i=1}^{m}\sqrt{z_i}$.

Vstup: c, x, μ, z taký, že $Ax - z = b$
Výstup: funkčná hodnota *Txmi* transformačnej funkcie

Algoritmus:
$Txmi = c^T x$
$\alpha = 0$
$for \quad i = 1,...,m$
$\quad \alpha = \alpha + \sqrt{z_i}$
$\quad \ldots\ldots\ldots\ldots$
$Txmi = Txmi - \mu * \alpha$

- Funkcia Lambda2 (x, Δ x, y, Δ y, z, Δ z, c, b, mi, zlrez). ktorá slúži na výpočet priesečníka polpriamok $y(\lambda) = y^k + \lambda\Delta y^k$, $z(\lambda) = z^k + \lambda\Delta z^k$ $(\lambda > 0)$ s množinou prípustných riešení $K = \left\{x \in R^n \mid Ax \geq b\right\}$. Na základe jeho existencie, resp. neexistencie rozhoduje o tom, či použijeme metódu Zlatého rezu na nájdenie optimálneho kroku λ, alebo či použijeme I. fázu za účelom nájdenia konečného intervalu na ktorom budeme hľadať optimálnu dĺžku kroku λ.

Vstup: x, Δx, $y > 0_m$, Δy, $z \geq 0_m$, r Δz, c, b, μ, počet krokov zlatého rezu *zlrez*
Výstup: privolanie funkcie FazaI2 alebo ZlatyRez2 a následný výpočet λ

Algoritmus: $u = 0, \quad \lambda_0 = 0, \quad \overline{\lambda} = 10^{10}$

for $i = 1,...,m$

⋮ *if* $(\Delta y_i < 0)$

⋮ ⋮ $\lambda_0 = -y_i / \Delta y_i$

⋮ ⋮ $u = 1$

⋮ ⋮ *if* $(\lambda_0 < \overline{\lambda})\overline{\lambda} = \lambda_0$

⋮ ⋮……

⋮ *if* $(\Delta z_i < 0)$

⋮ ⋮ $\lambda_0 = -z_i / \Delta z_i$

⋮ ⋮ $u = 1$

⋮ ⋮ *if* $(\lambda_0 < \overline{\lambda})\overline{\lambda} = \lambda_0$

⋮ ⋮……

if $(u = 0)$ *GO TO FazaI2* // popísaný na str. 34

else *GO TO Zlaty* Re $z2$ // popísaný na str. 35

- Funkcia FazaI2 (x, Δ x, z, Δ z, c, zlrez) slúži na výpočet konečného intervalu $\langle \lambda_0, \lambda_2 \rangle$, na ktorý sa bude aplikovať metóda zlatého rezu. Použije sa vtedy, keď smer $\Delta y \geq 0_m$, resp. smer $\Delta z \geq 0_m$.

Vstup: x, Δx, z, Δz, c, počet krokov zlatého rezu *zlrez*

Výstup: interval, na ktorom budeme aplikovať metódu zlatého rezu a následný výpočet λ

Algoritmus: $\tau = (\sqrt{5} + 1)/2$

$\lambda_2 = \tau^0, \quad \lambda_1 = 0, \quad \lambda_0 = 0, \quad k = 1$

$x^1 = x + \lambda_1 * \Delta x$

$x^2 = x + \lambda_2 * \Delta x$

$z^1 = z + \lambda_1 * \Delta z$

$z^2 = z + \lambda_2 * \Delta z$

while $(Txmi2(z^2, c, x^2, \mu) < Txmi2(z^1, c, x^1, \mu))$

⋮ $\lambda_0 = \lambda_1, \quad \lambda_1 = \lambda_2, \quad \lambda_2 = \lambda_2 + \tau^k, \quad k = k + 1$

⋮ $x^1 = x + \lambda_1 * \Delta x$

⋮ $x^2 = x + \lambda_2 * \Delta x$

⋮ $z^1 = z + \lambda_1 * \Delta z$

⋮ $z^2 = z + \lambda_2 * \Delta z$

⋮……

GO TO Zlaty Re $z2$ // popísaný na str. 35

- Funkcia ZlatyRez2 (x, Δ x, z, Δ z, c, mi, aa, bb, zlrez), ktorá vypočíta optimálnu dĺžku kroku λ. Počet krokov metódy zlatého rezu môžme regulovať, čo bude aj predmetom našich experimentov.

Vstup: x, Δx, z, Δz, c, μ, dolná hranica intervalu aa, horná hranica intervalu bb, počet krokov zlatého rezu $zlrez$

Výstup: optimálna dĺžka kroku λ

Algoritmus:

$\tau = \left(\sqrt{5}+1\right)/2, \quad z^1 = 2-\tau, \quad z^2 = \tau - 1, \quad k = 0$

$c^1 = aa + z_1 * (bb - aa)$

$x^1 = x + c^1 * \Delta x$

$z^1 = z + c^1 * \Delta z$

$f^1 = Txmi2\left(z^1, c, x^1, \mu\right), \quad c^2 = aa + z^2 * (bb - aa)$

$x^2 = x + c^2 * \Delta x$

$z^2 = z + c^2 * \Delta z$

$f^2 = Txmi2\left(z^2, c, x^2, \mu\right)$

$while\left(k \neq zlrez\right)$

⋮ $if\left(f^1 < f^2\right)$

⋮ ⋮ $bb = c^2, \quad c^2 = c^1, \quad f^2 = f^1, \quad c^1 = aa + z^1 * (bb - aa)$

⋮ ⋮ $x^1 = x + c^1 * \Delta x$

⋮ ⋮ $z^1 = z + c^1 * \Delta z$

⋮ ⋮ $f^1 = Txmi2\left(z^1, c, x^1, \mu\right)$

⋮ ⋮……

⋮ $else$

⋮ ⋮ $aa = c^1, \quad c^1 = c^2, \quad f^1 = f^2, \quad c^2 = aa + z^2 * (bb - aa)$

⋮ ⋮ $x^2 = x + c^2 * \Delta x$

⋮ ⋮ $z^2 = z + c^2 * \Delta z$

⋮ ⋮ $f^2 = Txmi2\left(z^2, c, x^2, \mu\right)$

⋮ ⋮……

⋮ $k = k + 1$

⋮……

$return \quad \lambda = c^1$

- <u>Výpočet novej aproximácie:</u>

$$x^{k+1} = x^k + \lambda \Delta x$$
$$y^{k+1} = y^k + \lambda \Delta y$$
$$z^{k+1} = z^k + \lambda \Delta z$$

- <u>Opakovanie cyklu.</u>

Tento budeme opakovať, kým nebude splnené nami stanovené „STOP“ kritérium.

- Vypočítali sme teda „dostatočnú“ aproximáciu optimálneho bodu $(\hat{x}(\mu_k), \hat{y}(\mu_k), \hat{z}(\mu_k))$ prvej transformačnej úlohy, čím sme ukončili aj prvú vonkajšiu iteráciu. Z postupnosti $\{\mu_l\}_{l=1}^{\infty}$ teraz vyberieme jej ďalší člen a prejdeme na druhú vonkajšiu iteráciu a jej sériu iterácii vnútorných. Toto aplikujeme, kým nedostaneme dostatočne presné optimálne riešenie $(\hat{x}, \hat{z})$ úlohy lineárneho programovania (P^*), resp. dostatočne presné duálne riešenie $\hat{y}$ úlohy (D^*).

3.3. Numerické experimenty.

Našou ďalšou úlohou je experimentálne otestovať numerické správanie sa primárno – duálnej metódy riešenia úlohy lineárneho programovania. Na tento účel sme použili programovú realizáciu primárno – duálneho algoritmu, ktorá je obsahom prílohy B. Rovnako ako v prípade primárnej metódy sme si ako ciele vytýčili:

- sledovať správanie sa metódy v závislosti od presnosti výpočtu optimálnej dĺžky kroku λ meranej počtom iterácií metódy zlatého rezu a následne výber takého počtu, aby príslušná Newtonova metóda bola najefektívnejšia,

- sledovať správanie sa metódy pre rôzne počty Newtonovských iterácií a následne empiricky vybrať taký „optimálny počet“ týchto iterácií, aby celkový výpočet bol najefektívnejší,

- porovnať výpočtovú zložitosť Newtonovského smeru s a optimálneho kroku λ,

- sledovať správanie sa metódy pre rôzne stratégie výberu bariérového parametra μ a následne vybrať stratégiu, s ktorou metóda dosahuje najlepšie výsledky.

Nakoľko všetko, čo sme uviedli pri úvode do experimentálnej časti v kapitole 2.3.ostáva v platnosti i v tomto prípade, prejdeme rovno k samotným experimentom. Výnimku tvorí len spôsob voľby vhodného štartovacieho μ, kde sa v prípade primárno – duálnej metódy obvykle volí vzťahom $(3.2.)$.

Analýza počtu iterácii metódy zlatého rezu.

V rámci tohto experimentu sme sledovali presnosť s akou metóda nachádza aproximáciu bodu centrálnej trajektórie pre $\mu = 1$ pri použití rôzneho počtu iterácií metódy zlatého rezu v jednotlivých Newtonových iteráciách. Výsledky experimentu sú zhrnuté v tabuľkách 3.1. – 3.3.

Spoločné vstupné parametre tabuliek 3.1. – 3.3.:
$\mu = 1$, $\rho = 100$ - vzdialenosť štartovacieho bodu x^0 od optimálneho riešenia $\hat{x}$.

Tab 3.1. *rozmer úlohy* $(n\,x\,m)=(5\,x\,10)$

	počet iterácii Newtonovej metódy			
	2	3	4	5
počet iterácií zlatého rezu	**priemerné vzdialenosti bodu optima *xopt* od vypočítanej aproximácie *x(1)* centrálnej trajektórie *x(mi)***			
5	159.0559	137.7785	109.5198	79.8440
10	158.0720	130.2043	93.1779	57.3169
15	159.3684	129.4068	83.1026	45.9553
20	158.9156	124.3629	76.1423	37.3735
25	158.6126	114.9065	62.5998	28.5405
30	158.2560	108.9206	57.1835	27.8795
35	158.2561	108.9222	57.1838	27.8831
40	158.2561	108.9223	57.1839	27.8831
45	158.2561	108.9223	57.1839	27.8832
50	158.2561	108.9223	57.1839	27.8832

Tab 3.2. *rozmer úlohy* $(n\,x\,m)=(10\,x\,20)$

	počet iterácii Newtonovej metódy			
	2	3	4	5
počet iterácií zlatého rezu	**priemerné vzdialenosti bodu optima *xopt* od vypočítanej aproximácie *x(1)* centrálnej trajektórie *x(mi)***			
5	226.3529	199.6774622	159.7731	114.1682
10	221.4557	184.5754874	134.0629	83.5060
15	220.5989	181.1916336	125.3745	73.6065
20	220.5615	175.6400898	116.0733	65.3161
25	221.5643	169.599275	107.7532	56.8294
30	222.0786	167.3792138	105.4356	55.0040
35	222.0786	167.3791847	105.4360	55.0052
40	222.0786	167.3791877	105.4358	55.0052
45	222.0786	167.3791867	105.4358	55.0052
50	222.0786	167.3791857	105.4358	55.0052

Tab 3.3. *rozmer úlohy* $(n\,x\,m)=(50\,x\,100)$

	počet iterácii Newtonovej metódy			
	2	3	4	5
počet iterácií zlatého rezu	**priemerné vzdialenosti bodu optima *xopt* od vypočítanej aproximácie *x(1)* centrálnej trajektórie *x(mi)***			
5	432.2311	425.5558	397.8307	354.7865
10	437.6916	425.7785	391.1856	328.6724
15	438.5417	429.4421	383.9719	309.3825
20	440.1792	428.5237	373.9018	288.1775
25	445.9449	427.0559	361.1943	268.2559
30	451.0773	425.2022	353.9620	256.9112
35	451.0773	425.2022	353.9620	256.9110
40	451.0773	425.2022	353.9621	256.9110
45	451.0773	425.2022	353.9621	256.9109
50	451.0773	425.2022	353.9621	256.9109

Z tabuliek 3.1. – 3.3. vyplýva, že vplyv počtu iterácii zlatého rezu je pri použití rôzneho počtu iterácií Newtonovej metódy odlišný. Rozlišujeme 2 prípady:

- Počet iterácií Newtonovej metódy sa rovná 2.
  V tomto prípade experimenty ukazujú, že nízky počet iterácií zlatého rezu zabezpečuje lepšiu výslednú vzdialenosť aproximácie od optimálneho riešenia. Dokonca v prípade úloh veľkých rozmerov (n = 50, m = 100) je použitie metódy zlatého rezu kontraproduktívne. Otázna však ostáva samotná konvergencia metódy pri použití takéhoto počtu iterácií Newtonovej metódy.
- Počet iterácií Newtonovej metódy je viac ako 2.
  Výsledky testov naznačujú, že vhodný počet iterácií zlatého rezu je 30. Vyšší počet iterácií sa ukazuje byť zbytočný.

Ďalší výsledkom experimentov je fakt, že pre prvú vonkajšiu iteráciu sa vzdialenosť aproximácie od optimálneho riešenia zväčší a až následne dochádza k približovaniu. Je to spôsobené faktom, že algoritmus sa snaží dostať do „blízkosti" centrálnej trajektórie. Zdá sa, že čím má úloha lineárneho programovania väčší rozmer, tým je toto počiatočné „oddialenie" väčšie.

Voľba vhodného počtu iterácií Newtonovej metódy.

Našim ďalším experimentom bude pozorovanie dosiahnutej presnosti v jednotlivých iteráciách Newtonovej metódy pri použití zafixovaného parametra $\mu = 1$ a počte iterácií zlatého rezu 30.

Vstupné parametre tabuľky 3.4.:
$\mu = 1$, rozmery úloh $(n\,x\,m) = (10\,x\,20)$, $\rho = 100$, počet iterácií zlatého rezu $= 30$

Tab 3.4. *Dosiahnutá presnosť v závislosti od počtu Newtonovských iterácií*

iteracia	c*x(l)	T(x(l),z(l),mi)	||rc||	||rmi||	lambda	Spádovosť	||x(l)-xopt||	||y(l)-yopt||	||z(l)-zopt||
l = 1	438 273.1200	437 174.7200	13 074.4470	13 728.5930	0.8132	Y	211.5852	26.6368	9 857.8199
l = 2	266 674.9900	265 812.6400	2 465.8428	3 783.8046	0.4617	Y	222.0786	19.8251	6 933.7036
l = 3	179 302.0020	178 678.1270	1 273.5670	1 923.4092	0.4664	Y	167.3792	17.3515	4 577.4225
l = 4	142 001.2580	141 565.9330	663.1984	942.3109	0.4753	Y	105.4356	16.6661	2 699.8210
l = 5	126 588.4950	126 302.7590	339.0446	439.1901	0.4578	Y	55.0040	16.2781	1 372.6126
l = 6	120 665.9240	120 488.6510	176.6847	199.9805	0.4170	Y	24.5983	15.2617	610.3903
l = 7	118 560.8360	118 453.0360	98.0949	95.6280	0.3591	Y	10.2932	13.8565	254.6727
l = 8	117 934.5880	117 863.6600	62.2939	54.1844	0.3334	Y	3.9217	12.3857	93.7737
l = 9	117 738.5490	117 686.6260	43.1674	34.2553	0.3895	Y	2.1811	10.8408	48.8442
l = 10	117 664.2400	117 620.9270	29.0561	21.4937	0.4733	Y	1.5073	9.6251	34.4998
l = 11	117 639.9630	117 600.1520	20.0750	14.2759	0.5463	Y	0.9568	8.1487	22.7429
l = 12	117 620.5120	117 583.7180	12.3969	8.4736	0.6264	Y	0.6475	6.6979	15.0966
l = 13	117 610.4890	117 575.4650	6.8612	4.6670	0.8263	Y	0.4786	5.8536	10.9612
l = 14	117 606.3980	117 572.2790	3.2097	2.2054	1.0166	Y	0.4135	5.2376	9.5141
l = 15	117 605.6580	117 571.7890	1.5488	1.1027	0.9801	Y	0.4015	5.1005	9.3155
l = 16	117 605.5170	117 571.6700	1.0394	0.7258	7.7259	99 / 1	0.3280	4.8994	7.5160
l = 17	117 604.1490	117 570.6530	0.5989	0.4000	3.8997	93 / 7	0.3182	4.8031	7.2996
l = 18	117 604.0550	117 570.6160	0.3168	0.2032	5.2690	91 / 9	0.3166	4.7792	7.2565
l = 19	117 604.0450	117 570.6060	0.1518	0.1065	5.1472	86 / 14	0.3133	4.7150	7.1964
l = 20	117 604.0070	117 570.5860	0.0990	0.0705	136.7652	81 / 19	0.3120	4.6812	7.1732

Z výsledkov v tabuľke 3.4. vidíme monotónnosť funkčnej hodnoty účelovej funkcie úlohy lineárneho programovania $c^T x$ ako aj transformačnej funkcie $T(x,z,\mu)$. Túto vlastnosť metódy nám zabezpečuje algoritmus metódy zlatého rezu.

Hodnoty v stĺpcoch *rc* a *rmi* hovoria o veľkosti reziduí $r_c = A^T y - c$ a $r_\mu = diag\left(2y_i\sqrt{z_i} - \mu\right) * e$ v jednotlivých iteráciach. Experimenty potvrdili, že $r_c, r_\mu \to 0$. Čo sa týka rezidua $r_b = Ax - b - z$, je stále rovné nule. To znamená, že sa stále pohybujeme v množine prípustných riešení K.

Nakoľko sme v teoretickej časti hovorili o odporúčaní redukcie dĺžky kroku, experimenty ukázali, že táto redukcia nastáva v priemere do 14 iterácie, jednotková dĺžka kroku je výrazne

prekoročená od iterácie číslo 16. V tejto iterácii nastáva aj porušenie spádovosti Newtonovského smeru $(\Delta x, \Delta y, \Delta z)$.

Ako sme vyššie spomenuli, experimenty boli robené pri fixovanom μ, takže nie je možné dosiahnuť optimálne riešenie $\hat{x}$, resp. duálne optimálne riešenie $\hat{y}$. Ale aj výsledky pre fixované μ nám ukazujú konvergenciu k bodu centrálnej trajktórie $\hat{x}(\mu)$, resp. $\hat{y}(\mu)$ úlohy lineárneho programovania.

Takisto ako v predchádzajúcom experimente si môžeme všimnúť počiatočné „oddialenie“ od optimálneho riešenia, nakoľko vzdialenosť štartovacieho bodu od optimálneho riešenia bola vo všetkých prípadoch 100 jednotiek.

V nasledujúcom experimente otestujeme správanie sa algoritmu pri použití klesajúcej postupnosti $\{\mu_k\}_{k=1}^{\infty}$ pre rôzne počty vnútorných iterácii. Ako štartovací parameter sme použili $\mu = 1$ a každý nový parameter počítali vzťahom $\mu_{k+1} = \frac{\mu_k}{\sigma^k}$.

Spoločné vstupné parametre tabuliek 3.4. – 3.8.:
$\mu_1 = 1$, rozmery úloh $(n\,x\,m) = (10\,x\,20)$, $\rho = 100$, počet iterácií zlatého rezu $= 30$, $\sigma = 10$ - redukčný parameter bariérového parametra

Tab 3.5. *Dosiahnutá presnosť pri zmene* μ *po* 2 *Newtonovských iteráciách*

mi	iteracia	c*x(l)	T(x(l),z(l),mi)	||rc||	||rmi||	Spádovosť	lambda	||x(l)-xopt||	||y(l)-yopt||	||z(l)-zopt||
1	l = 1	438273.1200	437174.7200	13074.4470	13728.5930	Y	0.8132	211.5852	26.6368	9857.8199
	l = 2	266674.9900	265812.6400	2465.8428	3783.8046	Y	0.4617	222.0786	19.8251	6933.7036
1.E-01	l = 3	179302.0020	179239.5600	1273.5670	1926.6832	Y	0.4296	169.4506	17.5164	4729.7642
	l = 4	145195.3720	145150.0800	731.2994	1044.4322	Y	0.4602	110.5853	16.7083	2876.3818
1.E-02	l = 5	128348.6050	128345.6020	385.4638	507.0956	Y	0.4183	63.7713	16.3146	1614.5491
	l = 6	121940.9410	121938.8940	217.5429	263.8778	Y	0.4008	30.8577	15.6629	780.8083
1.E-03	l = 7	119034.8910	119034.7290	122.3032	130.0054	Y	0.3068	15.1269	14.6724	384.1845
	l = 8	118146.1060	118146.0300	80.6797	77.1877	Y	0.2556	7.6344	13.2627	192.8525
1.E-04	l = 9	117854.4850	117854.4820	59.4918	52.8399	Y	0.2107	4.2732	12.0567	104.0316
	l = 10	117756.5280	117756.5240	48.2409	41.0972	Y	0.2211	2.7600	11.0596	64.0212
1.E-05	l = 11	117694.9760	117694.9760	38.6558	31.5278	Y	0.2136	2.0236	10.0686	46.0408
	l = 12	117659.1580	117659.1580	30.8143	24.1837	97 / 3	0.2426	1.5539	9.0991	35.5128
1.E-06	l = 13	117639.5250	117639.5250	24.5694	18.3345	99 / 1	0.2233	1.0824	8.3546	24.7003
	l = 14	117626.8290	117626.8290	20.6672	14.6515	94 / 6	0.2485	0.8761	7.6494	19.6398
1.E-07	l = 15	117620.0350	117620.0350	18.0005	11.5426	92 / 8	0.2601	0.7010	6.9639	15.3246
	l = 16	117611.7180	117611.7180	13.9941	8.7704	93 / 7	0.2596	0.6341	6.3275	13.6823
1.E-08	l = 17	117610.3690	117610.3690	13.9951	7.1385	89 / 11	0.2626	0.5891	5.9197	12.6229
	l = 18	117609.3520	117609.3520	14.5984	5.8476	83 / 17	0.2791	0.5248	5.5288	11.2494
1.E-09	l = 19	117608.4590	117608.4590	15.0159	4.8865	77 / 23	0.2599	0.5061	5.3648	10.8653
	l = 20	117608.3420	117608.3420	17.0263	4.3528	76 / 24	0.2719	0.4578	5.6134	9.7253

Tab 3.6. *Dosiahnutá presnosť pri zmene* μ *po* 3 *Newtonovských iteráciách*

mi	iteracia	c*x(l)	T(x(l),z(l),mi)	||rc||	||rmi||	Spádovosť	lambda	||x(l)-xopt||	||y(l)-yopt||	||z(l)-zopt||
1	l = 1	438273.1200	437174.7200	13074.4470	13728.5930	Y	0.8132	211.5852	26.6368	9857.8199
	l = 2	266674.9900	265812.6400	2465.8428	3783.8046	Y	0.4617	222.0786	19.8251	6933.7036
	l = 3	179302.0020	178678.1270	1273.5670	1923.4092	Y	0.4664	167.3792	17.3515	4577.4225
1.E-01	l = 4	142001.2580	141957.7000	663.1984	945.5025	Y	0.4480	107.4027	16.6982	2773.6200
	l = 5	127451.8540	127422.3770	360.0272	475.2004	Y	0.4423	58.3807	16.2912	1467.9145
	l = 6	121139.5180	121120.8520	193.1151	226.1156	Y	0.3985	26.8264	15.4853	672.1370
1.E-02	l = 7	118730.2680	118729.0920	109.0006	111.4387	Y	0.3075	12.1557	14.3432	304.7436
	l = 8	118013.6730	118012.9070	72.5782	66.8841	Y	0.2534	5.4035	12.8521	132.7350
	l = 9	117797.4540	117796.9580	54.8546	47.0510	Y	0.2591	3.0371	11.6074	71.2080
1.E-03	l = 10	117701.3600	117701.2930	41.5145	33.4608	Y	0.2255	2.3211	10.5906	53.6719
	l = 11	117669.6570	117669.6250	33.0417	25.6401	Y	0.2648	1.6363	9.4149	36.9073
	l = 12	117642.0210	117641.9840	25.2315	18.8271	Y	0.2830	1.0316	8.3563	24.3166
1.E-04	l = 13	117625.7310	117625.7300	19.3011	13.8325	Y	0.2777	0.7470	7.3933	18.0178
	l = 14	117616.3270	117616.3260	14.6753	10.1906	Y	0.3320	0.5930	6.4335	13.7591
	l = 15	117611.2400	117611.2400	10.8043	7.3734	Y	0.3527	0.4729	5.6461	10.7772
1.E-05	l = 16	117606.7160	117606.7160	7.7110	5.1543	Y	0.3334	0.3767	4.8404	8.5959
	l = 17	117605.4530	117605.4530	5.8105	3.8430	Y	0.4053	0.2830	4.2569	6.4861
	l = 18	117604.5700	117604.5700	4.2173	2.7426	Y	0.4415	0.2332	3.6587	5.4083
1.E-06	l = 19	117604.2880	117604.2880	3.0777	2.0092	Y	0.3968	0.1132	3.1848	2.8134
	l = 20	117603.4330	117603.4330	2.3516	1.5142	97 / 3	0.4789	0.0630	2.6919	1.3344

Tab 3.7. *Dosiahnutá presnosť pri zmene* μ *po* 4 *Newtonovských iteráciách*

mi	iteracia	c*x(l)	T(x(l),z(l),mi)	||rc||	||rmi||	Spádovosť	lambda	||x(l)-xopt||	||y(l)-yopt||	||z(l)-zopt||
1	l = 1	438273.1200	437174.7200	13074.4470	13728.5930	Y	0.8132	211.5852	26.6368	9857.8199
	l = 2	266674.9900	265812.6400	2465.8428	3783.8046	Y	0.4617	222.0786	19.8251	6933.7036
	l = 3	179302.0020	178678.1270	1273.5670	1923.4092	Y	0.4664	167.3792	17.3515	4577.4225
	l = 4	142001.2580	141565.9330	663.1984	942.3109	Y	0.4753	105.4356	16.6661	2699.8210
1.E-01	l = 5	126588.4950	126559.9100	339.0446	442.2706	Y	0.4306	57.1037	16.2583	1429.4826
	l = 6	120936.1460	120917.8960	185.9969	218.3598	Y	0.3870	26.4019	15.4180	658.2261
	l = 7	118689.0010	118677.7110	107.1499	109.4708	Y	0.3386	11.2172	14.0087	279.7601
	l = 8	117969.9910	117962.6310	69.0548	62.4586	Y	0.2881	4.8533	12.4943	117.8765
1.E-02	l = 9	117763.8160	117763.2660	50.1570	42.0991	Y	0.2229	2.7416	11.5219	63.3726
	l = 10	117685.4390	117684.9810	39.3348	31.3826	Y	0.2747	1.9169	10.2505	42.9150
	l = 11	117650.5940	117650.1240	29.4692	22.3556	Y	0.3124	1.3706	8.8473	31.4165
	l = 12	117630.3120	117629.9710	21.4468	15.5269	Y	0.3468	0.7649	7.5567	17.9229
1.E-03	l = 13	117614.6740	117614.6470	15.1160	10.4842	Y	0.3084	0.5402	6.6167	12.2722
	l = 14	117607.7680	117607.7400	11.0647	7.4341	Y	0.3528	0.4649	5.6587	10.5120
	l = 15	117606.5450	117606.5230	8.0501	5.3250	Y	0.4366	0.3152	4.7358	7.1370
	l = 16	117604.7070	117604.6940	5.3064	3.4571	Y	0.5039	0.2571	3.9311	5.8382
1.E-04	l = 17	117604.3880	117604.3880	3.5431	2.3129	Y	0.3924	0.2034	3.4086	4.8889
	l = 18	117604.1220	117604.1210	2.5926	1.6986	Y	0.5151	0.1032	2.8582	2.5246
	l = 19	117602.9820	117602.9820	1.7307	1.0995	Y	0.6086	0.0431	2.3013	0.8883
	l = 20	117602.6930	117602.6930	1.1762	0.7431	Y	0.9070	0.0304	1.7665	0.6170

Tabuľky 3.5. – 3.7. nám ukazujú výsledky týchto experimentov pre jednotlivé počty použitých vnútorných iterácií. Z experimentov vyplýva, že zvyšovaním počtu vnútorných iterácií zvyšujeme presnosť dosiahnutej aproximácie optimálneho riešenia $\hat{x}$, $\hat{y}$ a $\hat{z}$. Pri použití 2 vnútorných iterácií dochádza k prípadom nespádovosti smeru po viac ako 11 iteráciách. Táto skutočnosť je už použitím 4 vnútorných iterácií úplne eliminovaná.

Tab 3.8. *Početnosť prípadov v jednotlivých iteráciách*

počet Newton. iterácii = 2			počet Newton. iterácii = 3			počet Newton. iterácii = 4		
poradie iterácie	mi	počet prípadov	poradie iterácie	mi	počet prípadov	poradie iterácie	mi	počet prípadov
1	1	100	**1**	1	100	**1**	1	100
2		100	**2**		100	**2**		100
3	1.E-01	100	**3**		100	**3**		100
4		100	**4**	1.E-01	100	**4**		100
5	1.E-02	100	**5**		100	**5**	1.E-01	100
6		100	**6**		100	**6**		100
7	1.E-03	100	**7**	1.E-02	100	**7**		100
8		100	**8**		100	**8**		100
9	1.E-04	100	**9**		100	**9**	1.E-02	98
10		100	**10**	1.E-03	97	**10**		97
11	1.E-05	95	**11**		93	**11**		91
12		93	**12**		89	**12**		83
13	1.E-06	87	**13**	1.E-04	80	**13**	1.E-03	73
14		84	**14**		72	**14**		63
15	1.E-07	78	**15**		62	**15**		48
16		70	**16**	1.E-05	52	**16**		35
17	1.E-08	61	**17**		39	**17**	1.E-04	22
18		57	**18**		30	**18**		17
19	1.E-09	52	**19**	1.E-06	21	**19**		11
20		50	**20**		13	**20**		6
21	1.E-10	47	**21**		10	**21**	1.E-05	6
22		47	**22**	1.E-07	10	**22**		3
23	1.E-11	46	**23**		9	**23**		2
24		46	**24**		8	**24**		2
25	1.E-12	46	**25**	1.E-08	7	**25**	1.E-06	1
26		46	**26**		7	**26**		1
27	1.E-13	46	**27**		7	**27**		1
28		46	**28**	1.E-09	7	**28**		1
29	1.E-14	46	**29**		7	**29**	1.E-07	1
30		46	**30**		7	**30**		1

Pri experimente zobrazenom v tabuľke 3.8. sme použili ako kritérium presnosti riešenia vzťah $\left\| x_i\left(\mu_k\right) - \hat{x} \right\| < \varepsilon$, pričom $\varepsilon = 0{,}001$. Následne sme zaznamenali početnosť prípadov, v ktorých sa algoritmus dostal do k - tej iterácie. Experiment ukázal, že pri použitý 2 vnútorných iterácii v 46% prípadov ku konvergencii k optimálnemu riešeniu nedochádza. Aj z toho pohľadu sa javí použitie 4 vnútorných iterácií ako najvhodnejšie.

Porovnanie výpočtovej zložitosti Newtonovského smeru a optimálneho kroku.

Na základe hore uvedených experimentov môžeme v algoritme primárnej odmocninovej metódy použiť 30 krokov zlatého zlatého rezu a 4 iterácie Newtonovej metódy. Teraz sa na tieto parametre pozrieme z pohľadu ich výpočtovej zložitosti.

- Výpočet Newtonovského smeru.
  Na určení smeru sa podieľa Newtonova metóda na riešenie sústav nelineárnych rovníc a Q-R rozklad metódou Hansona a Lawsona. Pri stanovení výpočtovej zložitosti jednotlivých funkcií sme sa zaoberali operáciami násobenia a odmocňovania, z jednak z dôvodu ich náročnosti a na druhej strane z dôvodu početnosti ich použitia. Výpočtová zložitosť jednotlivých funkcií je nasledujúca:
  1. Newtonova metóda - operácie násobenia: $2m^2n + mn^2 + 4mn + m^2 + 5m$
     - operácie odmocňovania: $2m$
  2. Q-R rozklad - operácie násobenia: $n^3 + 3.5n^2 + 2n + 1.5$
     - operácie odmocňovania: n

Teda výsledná výpočtová zložitosť smeru je pre operáciu násobenia $2m^2n + mn^2 + n^3 + 4mn + 3.5n^2 + m^2 + 5m + 2n + 1.5$ a pre operáciu odmocňovania $2m + n$.

- Výpočet kroku.
  Na určenie kroku použijeme algoritmus metódy zlatého rezu, ktorý má pre operáciu násobenie výpočtovú zložitosť $(2k+4)m + (2k+4)n + k + 2$, kde k znamená počet iterácií metódy zlatého rezu. Pre operáciu odmocňovania je zložitosť rovná $(k+2)m + 1$.

Teda môžme povedať, že hlavný člen výpočtovej zložitosti operácie násobenie pre Newtonovský smer je $mn(n+2m)+n^3$ a pre krok $k(2m+2n)$. To znamená, že pre úlohy lineárneho programovania malých rozmerov počet iterácii metódy zlatého rezu ovplyvňuje výpočtovú zložitosť algoritmu výraznejšie ako pre úlohy veľkých rozmerov. Podobný záver vyplýva aj z porovnania operácie odmocňovania, pričom početnosť tejto operácie je mnohonásobne nižšia ako v prípade operácie násobenia.

Citlivosť metódy na voľbu postupnosti bariérového parametra μ.

V tejto časti sa budeme venovať spôsobu nastavenia počiatočného bariérového parametra μ_1, ako aj experimentálnemu testovaniu rôznych stratégii klesajúcej postupnosti $\{\mu_k\}_{k=1}^{\infty}$. Ako sme už uviedli, v praxi sa počiatočný parameter volí vzťahom $\mu_1 = \frac{1}{m}\sum_{i=1}^{m} 2y_i^0\sqrt{z_i^0}$. Pre postupnosť bariérového parametra môžme otestovať 2 druhy stratégie. Prvou je výpočet tohto parametra v každej iterácii pomocou metódy najmenších štvorcov a druhou určenie kladného parametra σ, pomocou ktorého by sme bariérový parameter vzťahom $\mu_{k+1} = \frac{\mu_k}{\sigma}$ v každej vonkajšej iterácii menili. Počet iterácii metódy zlatého rezu sme nastavili 30 a počet vnútorných iterácii 4.

1. stratégia.

Vstupné parametre tabuľky 3.9.:
$\mu_k = \frac{1}{m}\sum_{i=1}^{m} 2y_i^k\sqrt{z_i^k}$, rozmery úloh $(n\,x\,m) = (10\,x\,20)$, $\rho = 100$, počet iterácií zlatého rezu $= 30$, počet Newtonovských iterácií $= 4$

Tab 3.9. *Dosiahnutá presnosť v závislosti od parametra* μ

iteracia	mi	c*x(l)	T(x(l),z(l),mi)	||rc||	||rmi||	Spádovosť	lambda	||x(l)-xopt||	||y(l)-yopt||	||z(l)-zopt||
l = 1	2702.9156	438273	-2532153	13074.4470	6483.3467	Y	0.3821	378.1912	61.3258	24625.3690
l = 2		596830	-3288769	8075.3662	3833.1885	Y	0.5049	1262.8774	36.5523	53239.2270
l = 3		934911	-4532460	4022.9277	3795.7018	Y	0.6016	4212.3167	24.7098	138002.4780
l = 4		1654601	-6563645	1654.9203	4338.9968	Y	0.7589	15083.1445	21.4029	423974.1600
l = 5	1931.7499	3473916	-5943872	610.4439	2267.1951	Y	0.7464	30122.2210	20.6181	772401.3800
l = 6		4196228	-7237622	218.1590	1438.3664	Y	0.9393	62663.9890	20.7724	1546191.8300
l = 7		5941698	-8641155	74.2079	1047.4387	Y	0.9843	124380.3150	20.8034	2963709.2800
l = 8		8044132	-9964225	22.0013	627.6409	Y	0.9686	198791.3820	21.0066	4813471.4500
l = 9	1890.3147	9803680	-9990723	6.0422	195.4801	Y	1.0788	232885.3580	21.0226	5684653.1600
l = 10		10151350	-10128025	0.8808	57.6874	Y	1.2059	246070.2850	21.0172	6033979.5000
l = 11		10353163	-10140385	0.0621	5.5636	99 / 1	11.1183	247484.3610	21.0186	6071314.2700
l = 12		10372594	-10140497	0.0068	0.6101	99 / 1	4.0859	247660.5560	21.0186	6076101.6900
l = 13	1890.3069	10375044	-10140228	0.0007	0.0247	Y	4.3581	247647.3060	21.0186	6075793.7800
l = 14		10374877	-10140228	0.0001	0.0047	Y	3.2189	247649.6040	21.0186	6075846.9800
l = 15		10374911	-10140228	0.0000	0.0005	98 / 2	32.6859	247649.7160	21.0186	6075848.9900
l = 16		10374913	-10140228	0.0000	0.0004	99 / 1	8.6218	247649.7050	21.0186	6075848.9800
l = 17	1890.3069	10374911	-10140228	0.0000	0.0004	99 / 1	4.9341	247649.7150	21.0186	6075848.8800
l = 18		10374912	-10140228	0.0000	0.0003	97 / 3	12.4505	247649.7150	21.0186	6075848.9900
l = 19		10374912	-10140228	0.0000	0.0004	98 / 2	6.0645	247649.7160	21.0186	6075848.9700
l = 20		10374912	-10140228	0.0000	0.0004	98 / 2	25.7410	247649.7140	21.0186	6075848.9800

Z tabuľky 3.9. vyplýva, že pri použitý prvej stratégie nedochádza ku konvergencii k optimálnemu riešeniu a bariérový parameter ostáva natoľko veľký, že aproximácia optimálneho riešenia je „veľmi ďaleko“ od optimálneho riešenia. Pri testovaní tejto stratégie sme ani v jednom prípade nedosiahli dostatočne presnú aproximáciu optimálneho riešenia. Na druhej strane $r_c, r_\mu \to 0$, teda problémom ostáva fakt, že μ_k nekonverguje k nule.

V ďalších experimentoch odporúčame počiatočný bariérový parameter zmenšiť niekoľko násobne (napr. 100 krát, 1000 krát,...)

2.stratégia.

Spoločné vstupné parametre pre tabuľky 2.10. – 2.12.:

$\mu_1 = \frac{1}{m}\sum_{i=1}^{m} 2y_i^0\sqrt{z_i^0}$, rozmery úloh $(n\,x\,m) = (10\,x\,20)$, $\rho = 100$, počet iterácií zlatého rezu $= 30$, počet Newtonovských iterácií $= 4$

Pri použití 2 stratégie sme vypočítaný počiatočný bariérový parameter zmenšili vždy 1000 krát. Redukčný parameter σ sme postupne menili. Výsledky experimentu sú zhrnuté v tabuľkách 3.10. a 3.11..

Tab 3.10. *Dosiahnutá presnosť v závislosti od redukčného parametra* σ

	redukčný parameter σ											
	2			3			4			5		
iteracia	mi	‖x(l)-xopt‖	‖y(l)-yopt‖	mi	‖x(l)-xopt‖	‖y(l)-yopt‖	mi	‖x(l)-xopt‖	‖y(l)-yopt‖	mi	‖x(l)-xopt‖	‖y(l)-yopt‖
l = 1		211.9215	26.5489		211.9215	26.5489		211.9215	26.5489		211.9215	26.5489
l = 2	2.7029	223.0234	19.7839	2.7029	223.0234	19.7839	2.7029	223.0234	19.7839	2.7029	223.0234	19.7839
l = 3		168.7619	17.3258		168.7619	17.3258		168.7619	17.3258		168.7619	17.3258
l = 4		106.6119	16.6460		106.6119	16.6460		106.6119	16.6460		106.6119	16.6460
l = 5		54.9364	16.2508		54.8954	16.2467		54.9336	16.2462		55.1047	16.2419
l = 6	1.3515	24.3870	15.2530	0.9010	24.3791	15.2462	0.6757	24.5109	15.2578	0.5406	24.6971	15.2686
l = 7		9.1455	13.8382		10.2185	13.8283		10.3079	13.8147		10.3640	13.7913
l = 8		3.7127	12.3104		3.8591	12.3441		4.0067	12.3830		4.0467	12.3943
l = 9		2.1338	10.9077		2.3045	11.0701		2.3724	11.1475		2.4568	11.2246
l = 10	0.6757	1.6574	9.4243	0.3003	1.7010	9.6055	0.1689	1.7567	9.7075	0.1081	1.7748	9.8180
l = 11		0.9464	7.8522		0.9953	8.0555		1.0555	8.2542		1.0765	8.4020
l = 12		0.5599	6.4449		0.5885	6.3145		0.5720	6.5579		0.6154	6.7125
l = 13		0.4036	5.4939		0.3765	5.3842		0.3807	5.6914		0.3889	5.8650
l = 14	0.3379	0.3431	4.7198	0.1001	0.3237	4.4901	0.0422	0.2953	4.6971	0.0216	0.3088	4.7983
l = 15		0.1786	3.9288		0.2504	3.5432		0.2702	3.6648		0.2806	3.7516
l = 16		0.0878	3.4831		0.0909	2.9495		0.1022	2.8812		0.1039	2.9880
l = 17		0.0543	2.9950		0.0432	2.6190		0.0502	2.4946		0.0523	2.6263
l = 18	0.1689	0.0293	2.4048	0.0334	0.0243	2.0247	0.0106	0.0300	1.9872	0.0043	0.0302	2.1246
l = 19		0.0140	1.9866		0.0191	1.4687		0.0195	1.4698		0.0195	1.5719
l = 20		0.0140	1.7573		0.0024	1.0754		0.0012	1.0481		0.0000	1.0939

Tab 3.11. *Dosiahnutá presnosť v závislosti od redukčného parametra* σ

	redukčný parameter σ											
	6			8			10			20		
iteracia	mi	‖x(l)-xopt‖	‖y(l)-yopt‖	mi	‖x(l)-xopt‖	‖y(l)-yopt‖	mi	‖x(l)-xopt‖	‖y(l)-yopt‖	mi	‖x(l)-xopt‖	‖y(l)-yopt‖
l = 1		211.9215	26.5489		211.9215	26.5489		211.9215	26.5489		211.9215	26.5489
l = 2	2.7029	223.0234	19.7839	2.7029	223.0234	19.7839	2.7029	223.0234	19.7839	2.7029	223.0234	19.7839
l = 3		168.7619	17.3258		168.7619	17.3258		168.7619	17.3258		168.7619	17.3258
l = 4		106.6119	16.6460		106.6119	16.6460		106.6119	16.6460		106.6119	16.6460
l = 5		55.4229	16.2363		56.2985	16.2263		56.9176	16.2214		58.7547	16.2167
l = 6	0.4505	24.9524	15.2684	0.3379	25.6105	15.2747	0.2703	26.1264	15.2872	0.1351	27.0965	15.3474
l = 7		10.4735	13.7926		10.8512	13.8385		11.0784	13.8565		11.5144	13.9170
l = 8		4.1358	12.4053		4.5974	12.4587		4.7218	12.4346		4.9776	12.4497
l = 9		2.4907	11.2680		2.5659	11.4143		2.5970	11.4307		2.8614	11.5360
l = 10	0.0751	1.7944	9.8529	0.0422	1.7714	9.9262	0.0270	1.8266	9.9746	0.0068	1.9730	10.1894
l = 11		1.0592	8.4188		1.3325	8.5010		1.3467	8.5921		1.3997	8.9013
l = 12		0.6332	6.7389		0.6642	7.0475		0.6819	7.2421		0.8041	7.8035
l = 13		0.3991	5.9239		0.4645	6.2450		0.4649	6.4074		0.5663	6.9664
l = 14	0.0125	0.3034	4.7785	0.0053	0.3813	5.1490	0.0027	0.3900	5.3081	0.0003	0.4551	5.9031
l = 15		0.2864	3.8259		0.2898	4.1922		0.3009	4.3795		0.3491	5.0475
l = 16		0.1373	3.0288		0.1492	3.3009		0.1453	3.5357		0.3132	4.2595
l = 17		0.0762	2.6812		0.0935	2.8986		0.1059	3.1419		0.1818	3.6703
l = 18	0.0021	0.0247	2.1752	0.0007	0.0470	2.3915	0.0003	0.0430	2.6162	2.E-05	0.1112	3.0947
l = 19		0.0195	1.6435		0.0302	1.9152		0.0302	2.0765		0.0558	2.5476
l = 20		0.0040	1.1443		0.0086	1.4922		0.0241	1.6177		0.0335	2.0746

Z experimentu vyplýva ako najvhodnejšia voľba redukčného parametra $\sigma = 5$. Ako ukazuje tabuľka 3.12. pri použití redukčného parametra v intervale 3 - 5 metóda konverguje najrýchlejšie.

Tab 3.12. *Poč etnosť prípadov v závislosti od parametra* σ

	redukčný parameter σ							
	2	3	4	5	6	8	10	20
iteracia	počet prípadov v %							
1	100	100	100	100	100	100	100	100
2	100	100	100	100	100	100	100	100
3	100	100	100	100	100	100	100	100
4	100	100	100	100	100	100	100	100
5	100	100	100	100	100	100	100	100
6	100	100	100	100	100	100	100	100
7	100	99	100	100	100	100	100	100
8	100	99	99	99	99	99	99	100
9	100	99	99	98	98	98	97	97
10	100	96	95	95	95	95	95	95
11	100	92	93	92	92	91	91	92
12	100	85	82	76	76	77	80	85
13	100	72	62	61	63	66	71	74
14	100	50	43	46	49	53	55	69
15	100	37	30	32	34	35	38	47
16	100	31	22	23	23	24	30	38
17	100	22	16	14	18	18	20	28
18	100	10	10	10	11	13	14	20
19	98	5	6	5	5	8	7	12
20	91	3	2	2	2	5	5	10
21	71	1	1	1	1	2	3	6
22	49	1	1	1	1	1	2	5
23	24	1	1	1	1	1	1	3
24	8	1	0	1	1	1	1	1
25	3	0	0	0	1	1	1	1

Zhrnutie – závery numerických experimentov

- Experimenty naznačili, že aproximácia optimálneho riešenia sa v prvej vonkajšej iterácii vzdiali od optimálneho riešenia $\hat{x}$.
- Ako vhodný počet iterácii metódy zlatého rezu sa javí počet 30. Použitie väčšieho počtu je zbytočné.
- Monotónnosť funkčnej hodnoty účelovej funkcie $c^T x$ ako aj transformačnej funkcie $T(x, z, \mu)$.
- Zvyšovaním počtu vnútorných iterácií zvyšujeme presnosť dosiahnutej aproximácie optimálneho riešenia $\hat{x}$, resp. $\hat{y}$.
- Ako vhodný počet Newtonovských iterácií sa javí počet $= 4$ .Pri tomto počte už nedochádza k nespádovosti smeru ako v prípadoch 2 a 3 vnútorných iterácií.
- S rastom rozmerov úlohy lineárneho programovania (P) klesá vplyv zvoleného počtu iterácií metódy zlatého rezu na celkovej výpočtovej zložitosti algoritmu.
- Potreba redukcie štartovacieho bariérového parametra μ_1 za účelom zrýchlenia konvergencii algoritmu.
- Najrýchlejšiu konvergenciu experimenty rozmerov $(10\,x\,20)$ ukázali pri voľbe σ v rozmezí 3 – 5, pričom pri použití redukčného parametra $\sigma = 4$ po 20 iteráciách dostávame „superpresné“ optimálne riešenie $\hat{x}$.
- Pomalšia konvergencia duálnej premennej y ako v prípade primárnej premennej x

4. Porovnanie dvoch testovaných metód.

V numerických experimentoch primárnej a primárno - duálnej metódy riešenia úlohy lineárneho programovania sme si vytýčili určité ciele, ktoré sme pomocou numerických experimentov chceli dosiahnuť. Teraz sa na ich výsledky pozrieme z pohľadu porovnania týchto dvoch testovaných metód.

Analýza počtu iterácií metódy zlatého rezu.

Cieľom tohto experimentu bolo sledovanie správania sa metód v závislosti od presnosti výpočtu optimálnej dĺžky kroku λ. Numerické experimenty odhalili nasledujúce skutočnosti:

- V obidvoch prípadoch pri použití dvoch Newtonovských iterácií sa zdá byť použitie metódy zlatého rezu zbytočné.
- Pri použití viac ak 2 Newtonovských iterácií je pri primárno – duálnej metóde potrebný menší počet iterácií metódy zlatého rezu na dosiahnutie dostatočne presnej aproximácie bodu centrálnej trajektórie ako v prípade primárnej metódy.
- V obidvoch prípadoch experimenty ukázali počiatočné vzdialenie aproximácie od bodu optimálneho riešenia, ale v prípade primárnej metódy sa toto oddialenie javí menšie.

Voľba vhodného počtu iterácií Newtonovej metódy.

V tomto experiment sme sa zaoberali sledovaním správania s metód pre rôzne počty vnútorných Newtonovských iterácií. Porovnaním záverom môžeme vysloviť nasledujúce tvrdenia.

- V obidvoch prípadoch metóda zlatého rezu použitá pre určenie optimálnej dĺžky kroku zabezpečila monotónnosť účelovej funkcie $c^T x$ ako aj transformačnej funkcie $T(x,\mu)$, resp. $T(x,z,\mu)$.
- Z pohľadu konvergencie duálnej premennej y k optimálnemu duálnemu riešeniu $\hat{y}$ pri použití primárno – duálnej metódy platí $y(\mu_k) \underset{\mu_k \downarrow 0}{\rightarrow} \hat{y}$. V primárnej metóde sme túto premennú len odhadovali a ku konvergencii nedochádzalo.Tento odhad navyše pri použití viac ako 3 vnútorných Newtonovských iterácii často krát zlyhával , čo bolo indikované porušením slabej vety o dualite, t.j. vzťahu $b^T y \le c^T x$.
- Experimenty ukázali, že v prípade primárnej metódy sa ako vhodný počet vnútorných iterácií javí použitie 3 iterácií, kým v prípade primárno – duálnej metódy je to 4.
- Pri použití nami navrhovaných parametrov metód (počet iterácií metódy zlatého rezu, počet vnútorných Newtonovských iterácií) je pre úlohy rozmerov $(10\,x\,20)$ rýchlosť konvergencie $x(\mu_k) \underset{\mu_k \downarrow 0}{\rightarrow} \hat{x}$ v oboch prípadoch porovnateľná, t.j. v prípade primárnej metódy 95% úloh skonverguje celkovo za 22 iterácií a v prípade primárno – duálnej metódy za 21-22 iterácií. Ale pre úlohy veľkých rozmerov sa už prejavuje väčšia výpočtová zložitosť primárno – duálnej metódy a táto konverguje pomalšie. Na druhej strane je jej výhoda v konvergencii duálnej premennej, t.j. $y(\mu_k) \underset{\mu_k \downarrow 0}{\rightarrow} \hat{y}$.

Porovnanie výpočtovej zložitosti Newtonovského smeru a optimálneho kroku.

V predchádzajúcich kapitolách sa zaoberali porovnaním výpočtovej zložitosti Newtonovského smeru a optimálneho kroku pre jednu konkrétnu metódu. Teraz sa na tento problém pozrieme z pohľadu porovnania 2 metód.

- Výpočet Newtonovského smeru.

  Primárna metóda:
  - operácia násobenia: $mn^2 + n^3 + 5mn + 3.5n^2 + 3m + 3n + 2.5$
  - operácia odmocňovania: $3m + n$

  Primárno- duálna metóda:
  - operácia násobenia: $2m^2n + mn^2 + n^3 + 4mn + 3.5n^2 + m^2 + 5m + 2n + 1.5$
  - operácia odmocňovania: $2m + n$

  Ako vidíme z hore uvedeného prehľadu, výpočtová zložitosť Newtonovského smeru pre operáciu násobenia je v primárno – duálnej metóde väčšia o rádovo m^2n operácií. Výpočtová zložitosť pre operáciu odmocňovania je porovnateľná pre obe metódy.

- Výpočet optimálneho kroku.

  Primárna metóda:
  - operácia násobenia: $(k+2)mn + (k+1)m + (k+3)n + k + 2$
  - operácia odmocňovania: $(k+1)m + 1$

  Primárno- duálna metóda:
  - operácia násobenia: $(2k+4)m + (2k+4)n + k + 2$
  - operácia odmocňovania: $(k+2)m + 1$

  Naopak, výpočtová zložitosť optimálneho kroku pre operáciu násobenia je v primárno – duálnej metóde menšia o rádovo kmn operácii, kde k je počet iterácií metódy zlatého rezu. Pre operáciu odmocňovania je výpočtová zložitosť metód opäť porovnateľná.

Môžeme povedať, že v obidvoch metódach nárastom rozmerov úlohy klesá vplyv voľby počtu iterácií metódy zlatého rezu na celkovej výpočtovej zložitosti algoritmu, s tým, že výpočet Newtonovského smeru je „lacnejší“ v prípade primárnej metódy a výpočet optimálnej dĺžky kroku v prípade primárno – duálnej metódy.

Citlivosť metódy na voľbu postupnosti bariérového parametra μ.

Cieľom tohto experimentu bolo sledovaťsprávanie sa metód pre rôzne stratégie výberu bariérového parametra μ. Numerické experimenty odhalili nasledujúce skutočnosti.

- V oboch metódach pri použití 1. stratégie výberu postupnosti $\{\mu_k\}_{k=1}^{\infty}$ bariérového parametra μ nedochádza ku konvergencii $x^l(\mu_k) \to \hat{x}$.
- Pri použití 2. stratégie $\mu_{k+1} = \frac{\mu_k}{\sigma}$, v prípade primárno - duálnej metódy riešenie úlohy dosahuje najlepšiu presnosť pri použití redukčného parametra $\sigma = 5$. Pri metóde primárnej je najlepšia presnosť dosahovaná pri použití čo najmenšieho σ, ale pre hodnoty $\sigma < 10$ často dochádza k zlyhaniu odhadu y (v podmienke prípustnosti).

- Kým primárna metóda dosahuje najrýchlejšiu konvergenciu $x^t(\mu_k) \to \hat{x}$ pre $\sigma \in \langle 10,30 \rangle$, metóda primárno – duálna pre $\sigma \in \langle 3,5 \rangle$.

Záver.

V tejto práci sme sa zaoberali odmocninovou transformačnou funkciou aplikovanou na dve metódy riešena úlohy lineárneho programovania: primárnu a primárno – duálnu.

Numerické experimenty nám potvrdili, že na rozdiel od logaritmickej transformačnej funkcie odmocninová transformačná funkcia nezachováva symetriu primárno – duálnych vzťahov (Tab. 3.5. – 3.7.). Navyše pri použití primárnej metódy riešenia úlohy lineárneho programovania, kde duálnu premennú len odhadujeme, nám experimenty odhalili, že ku konvergencii k optimálnej hodnote duálnej premennej nedochádza a dokonca odhad duálnej premennej môže zlyhať(v podmienke prípustnosti). Tieto závery nám potvrdzujú tabuľky 2.8., 2.10. a 2.11..

Experimentálnym porovnaním týchto dvoch metód sme prišli k záveru, že pre úlohy malých rozmerov je rýchlosť konvergencie primárnej premennej u oboch metód porovnateľná (Tab. 2.12. a 3.12.), ale záleží aj na nastavení vhodných parametrov pre jednotlivé metódy. Pri úlohách veľkých rozmerov, ktorých výsledky sú zobrazené v prílohe A, sa ukázalo, že pri použití primárnej metódy primárna premenná konverguje rýchlejšie (napr. tabuľka na str. 54 verzus tabuľka na str. 65).

Z hľadiska výpočtovej zložitosti algoritmov, pri väčšom rozmere úloh, kde ťažisko výpočtov preberá výpočet Newtonovského smeru, je náročnejšia primárno – duálna metóda. Vyplýva to aj z porovnania priemerného času trvania riešenia jednej úlohy (tabuľky v prílohe A).

Čo sa týka správania sa metód v závislosti od presnosti výpočtu optimálnej dĺžky kroku, experimenty naznačili. že pri použití primárnej metódy je potrebný väčší počet iterácii metódy zlatého rezu na dosiahnutie dostatočne presnej aproximácie bodu centrálnej trajektórie ako pri použití primárno – duálnej metódy. Dokonca v prípade použitia 2 vnútorných iterácii sa výpočet optimálnej dĺžky kroku v oboch metódach javí zbytočný (Tab. 2.1 – 2.3 a 3.1. - 3.3.).

Porovnaním stratégií výberu postupnosti bariérového parametra μ sa ukázalo, že redukčný parameter σ, ktorý zabezpečí najlepšiu konvergenciu je pri použití primárnej metódy väčší ako pri použití primárno – duálnej metódy (Tab. 2.12. a 3.12.).

Použitá literatúra.

[1] den Hertog D., Ross C., Terlaky T., *Inverse berrier methods for linear programming,* Report No. 91-27, Faculty of Mathematics and Informatics/Computer Science, Delft University of Technology, (1991)

[2] Freund R. M., Mizuno S., *Interior point methods: Current status and future directions*, OPTIMA – Mathematical Programming Society Newsletter, č. 51, 1 – 9, (1996)

[3] Goceliaková, Z., *Metódy vnútorného bodu na riešenie úloh lineárneho programovania*, diplomová práca, MFF UK, (2000)

[4] Gondzio J. ,Terlaky T, *A Compuuttational View of Interior-Point Methods for Linear Programming,*, Advances in Linear and Integer Programming, Oxford Lecture Series, č.4, Clarendon Press, Oxford, 103 – 144, (1994)

[5] Halická, M., *Dvadsať rokov moderných metód vnútorného bodu*, Pokroky matematiky, fyziky a astronómie, ročník 49, č.3, 234 – 244, (2004).

[6] Halická M., Hamala M., *Duality of transformation functions in the interior point methods*, Acta Math. Univ. Comenianae Vol. LXV, 229 – 245, 2(1996)

[6] Hamala, M., *Prednášky z nelineárneho programovanie*, prednášky 3.ročník, (1998).

[7] Vanderbei, R. J., *Linear programming: Foundations and Extensions*, Princeton, http://princeton.edu/~rvdb/LPbook/, (2001)

A. Tabuľky numerických experimentov pre úlohy rozmeru 50 x 100

metóda: primárna

vstup: počet riešených úloh = **40**

rozmer úlohy $(n\,x\,m) = (50\,x\,100)$

vzdialenosť štartovacieho bodu x^0 od optimálneho riešenia $\hat{x}$ je $\rho = 100$

kritérium presnosti: $\|x_i(\mu_k) - \hat{x}\| < 0.001$

počet vnútorných Newtonovkých iterácií $= 2$

počet iterácií metódy zlatého rezu $= 25$

veľkosť redukčného parametra $\sigma = 10$

výstup:

Úloha číslo	počet vonkajších iterácií	počiatočné mí	c*x-c*xopt	||gradient||	||hessian||	||x-xopt||	||y-yopt||	priemerné lambda	CPU time
1	12	2.1012	0.9547	38176.8000	2.15E+19	2.48E-06	84.7290	0.000084	1.783
2	10	2.0412	0.1159	35193.7000	4.15E+10	6.05E-04	78.5875	0.000095	1.552
3	12	2.0768	0.1454	37125.9000	8.23E+17	1.62E-06	82.3711	0.000065	1.772
4	13	2.1898	0.0087	42750.8000	4.08E+19	1.27E-06	88.6905	0.000000	1.973
5	13	2.1956	0.1618	41995.2000	1.13E+18	7.96E-06	87.3499	0.000077	1.923
6	10	2.1565	0.0028	40698.6000	2.96E+15	5.16E-06	84.4808	0.000119	1.483
7	18	2.0733	1.4465	36523.4000	5.67E+17	1.29E-06	82.0366	0.000055	2.694
8	20	2.1506	4.3204	40397.3000	3.39E+10	3.30E-05	85.7205	0.000045	2.995
9	11	2.1633	0.0298	40606.4000	4.24E+14	3.50E-04	91.0494	0.000092	1.672
10	10	2.1500	0.0923	40688.2000	8.20E+12	4.61E-04	89.7051	0.000089	1.563
11	12	2.0984	0.0156	37792.8000	1.56E+16	3.26E-04	83.1264	0.000092	1.772
12	13	2.1548	2.9084	40047.1000	8.91E+14	2.48E-06	87.7781	0.000067	1.933
13	11	2.1042	0.0495	38227.2000	2.50E+13	5.02E-05	82.7224	0.000080	1.602
14	10	2.1190	3.6853	39072.6000	3.59E+10	1.10E-04	84.9647	0.000076	1.502
15	15	2.1733	3.6149	42045.1000	1.89E+20	1.15E-05	90.3881	0.000059	2.293
16	13	2.0726	0.1516	37365.4000	3.51E+16	3.53E-06	81.9451	0.000066	1.993
17	24	2.0717	4.2001	36230.0000	1.16E+02	5.71E-07	79.4984	0.000044	3.625
18	8	2.0149	0.0510	34409.5000	1.16E+18	5.27E-05	75.5179	0.000125	1.222
19	9	2.0817	0.3231	36380.2000	5.80E+15	9.47E-04	81.6945	0.000099	1.362
20	11	2.1266	0.0340	39396.6000	1.55E+11	6.02E-04	86.9368	0.000073	1.603
21	12	2.1230	0.0043	38319.7000	3.53E+19	4.28E-06	83.9107	0.000077	1.852
22	11	2.2749	0.0272	46590.9000	1.30E+12	7.66E-05	97.3653	0.000090	1.633
23	9	2.1195	0.0357	39489.6000	1.95E+14	2.58E-04	86.0871	0.000097	1.382
24	12	2.1981	1.1513	42046.3000	1.15E+10	3.09E-04	92.7362	0.000072	1.853
25	14	2.1203	0.5421	38351.3000	1.14E+23	1.18E-06	84.9765	0.000070	2.083
26	9	2.0480	0.0658	36069.9000	1.17E+16	1.58E-04	77.8717	0.000112	1.372
27	10	2.0637	0.0324	36357.5000	7.95E+13	3.07E-04	80.9630	0.000088	1.552
28	14	2.0725	2.4424	36766.2000	2.42E+24	1.74E-06	81.9329	0.000063	2.143
29	9	2.0544	0.3926	35372.2000	9.21E+12	2.34E-04	79.4040	0.000111	1.372
30	13	2.0187	0.0174	34357.0000	2.49E+12	4.90E-04	78.2752	0.000069	1.912
31	11	2.2416	0.0579	45367.7000	2.14E+12	1.02E-04	93.3970	0.000082	1.603
32	13	2.1621	0.1248	40983.2000	8.19E+13	5.44E-05	87.7211	0.000061	1.933
33	11	2.1222	0.0198	39655.4000	6.95E+14	2.44E-04	85.0059	0.000097	1.692
34	11	2.1198	0.1062	39243.7000	1.90E+12	1.42E-04	83.0482	0.000104	1.622
35	10	2.0439	0.3709	35364.7000	8.06E+13	5.21E-04	80.1873	0.000128	1.452
36	23	2.1095	0.0838	38448.5000	5.17E+09	1.04E-07	82.6438	0.000052	3.455
37	15	2.1390	0.5622	40218.1000	2.34E+21	4.47E-07	85.0059	0.000062	2.323
38	10	2.1748	0.0190	41842.5000	5.61E+13	4.52E-04	90.7414	0.000097	1.542
39	11	2.2324	0.6986	43878.6000	2.04E+13	2.07E-05	92.6121	0.000090	1.612
40	16	2.0766	0.6548	36982.5000	7.19E+20	6.50E-07	80.2496	0.000063	2.463
priemer	**12.475**	**2.1208**	**0.7430**	**39020.7075**	**6.35E+22**	**1.74E-04**	**84.8357**	**0.000080**	**1.8792**

metóda: primárna

vstup: počet riešených úloh = **40**

rozmer úlohy $(n\,x\,m) = (50\,x\,100)$

vzdialenosť štartovacieho bodu x^0 od optimálneho riešenia $\hat{x}$ je $\rho = 100$

kritérium presnosti: $\|x_i(\mu_k) - \hat{x}\| < 0.001$

počet vnútorných Newtonovských iterácií $= 3$

počet iterácií metódy zlatého rezu $= 25$

veľkosť redukčného parametra $\sigma = 10$

výstup:

Úloha číslo	počet vonkajších iterácií	počiatočné mí	c*x-c*xopt	||gradient||	||hessian||	||x-xopt||	||y-yopt||	priemerné lambda	CPU time
1	11	2.0805	1.3812	36988.0000	4.05E+27	7.34E-07	81.8533	0.00019	2.624
2	12	2.1371	0.0915	40835.8000	9.15E+27	3.99E-08	86.3597	0.00020	3.815
3	6	2.1395	0.3273	39685.3000	2.16E+21	9.09E-04	85.9891	0.00036	1.412
4	16	2.1245	1.1782	39823.7000	9.17E+20	4.20E-07	84.7231	0.00015	3.675
5	7	2.1718	0.2648	41494.0000	1.93E+16	8.78E-04	89.4918	0.00034	1.482
6	8	2.1290	0.7642	39099.1000	2.08E+22	6.49E-05	82.9312	0.00031	1.782
7	12	2.0939	0.1606	37184.7000	2.82E+23	1.22E-04	82.9096	0.00022	3.475
8	8	2.1012	0.9688	38168.4000	2.11E+27	2.50E-06	84.7138	0.00032	1.793
9	7	2.0412	0.1155	35192.8000	4.16E+16	6.08E-04	78.5865	0.00030	1.542
10	8	2.0768	0.0514	37121.0000	9.57E+25	1.13E-06	82.3600	0.00023	1.782
11	9	2.1898	0.0087	42748.6000	2.26E+28	1.14E-06	88.6855	0.00021	2.013
12	9	2.1956	0.1681	41993.8000	1.13E+26	8.09E-06	87.3472	0.00027	1.922
13	7	2.1565	0.0027	40692.6000	3.04E+21	5.08E-06	84.4736	0.00045	1.492
14	10	2.0733	1.9033	36521.2000	1.94E+31	1.70E-06	82.0331	0.00022	2.444
15	12	2.1506	4.5503	40397.3000	1.73E+25	1.28E-05	85.7205	0.00017	2.894
16	13	2.2842	1.4570	46141.8000	3.48E+26	3.92E-07	94.1435	0.00020	3.185
17	8	2.1633	0.0299	40605.6000	4.12E+20	3.52E-04	91.0482	0.00030	1.702
18	7	2.1500	0.0897	40686.2000	9.22E+18	4.55E-04	89.7017	0.00030	1.593
19	8	2.0984	0.0145	37790.9000	1.56E+24	3.25E-04	83.1214	0.00032	1.843
20	9	2.1548	2.9040	40046.8000	8.78E+22	2.40E-06	87.7776	0.00024	1.923
21	7	2.1042	0.0512	38223.6000	2.40E+21	5.39E-05	82.7153	0.00029	1.622
22	7	2.1190	3.6907	39071.2000	3.60E+16	1.10E-04	84.9632	0.00027	1.563
23	11	2.1733	1.3709	42044.8000	5.04E+30	7.80E-07	90.3877	0.00018	2.494
24	9	2.0726	0.1204	37364.7000	2.43E+25	3.23E-06	81.9433	0.00021	2.023
25	15	2.0717	5.4919	36230.0000	1.31E+19	7.57E-07	79.4984	0.00016	3.525
26	6	2.0149	0.0500	34390.4000	1.17E+22	5.19E-05	75.4908	0.00051	1.262
27	6	2.0817	0.3214	36364.4000	5.80E+21	9.50E-04	81.6672	0.00039	1.402
28	7	2.1266	0.0338	39394.0000	1.55E+19	5.97E-04	86.9338	0.00027	1.602
29	8	2.1230	0.0042	38309.7000	3.74E+27	5.18E-06	83.8918	0.00027	1.902
30	7	2.2749	0.0277	46586.9000	1.30E+20	7.66E-05	97.3597	0.00033	1.642
31	6	2.1195	0.0357	39472.4000	1.93E+20	2.54E-04	86.0650	0.00037	1.422
32	8	2.1981	1.3352	42045.9000	1.15E+18	2.90E-04	92.7357	0.00023	1.913
33	6	2.0480	0.0670	36050.7000	1.19E+22	1.49E-04	77.8458	0.00042	1.402
34	7	2.0637	0.0322	36355.5000	7.93E+19	3.09E-04	80.9605	0.00033	1.621
35	6	2.0544	0.3926	35360.1000	9.21E+18	2.34E-04	79.3880	0.00040	1.412
36	7	2.2416	0.0578	45364.9000	2.15E+20	1.02E-04	93.3932	0.00029	1.772
37	9	2.1621	0.1066	40983.0000	8.32E+21	5.53E-05	87.7206	0.00021	2.063
38	8	2.1222	0.0199	39654.7000	6.97E+20	2.44E-04	85.0048	0.00033	1.753
39	7	2.1198	0.1060	39241.4000	1.91E+20	1.43E-04	83.0445	0.00036	1.752
40	7	2.0439	0.3473	35361.6000	7.79E+19	5.33E-04	80.1827	0.00047	1.613
priemer	**8.525**	**2.1262**	**0.7524**	**39277.1875**	**6.11E+29**	**1.98E-04**	**85.1291**	**0.00029**	**2.0038**

metóda: primárna

vstup: počet riešených úloh = **40**

rozmer úlohy $(n\,x\,m) = (50\,x\,100)$

vzdialenosť štartovacieho bodu x^0 od optimálneho riešenia $\hat{x}$ je $\rho = 100$

kritérium presnosti: $\|x_i(\mu_k) - \hat{x}\| < 0.001$

počet vnútorných Newtonovských iterácií $= 4$

počet iterácií metódy zlatého rezu $= 25$

veľkosť redukčného parametra $\sigma = 10$

výstup:

Úloha číslo	počet vonkajších iterácií	počiatočné mí	c*x-c*xopt	||gradient||	||hessian||	||x-xopt||	||y-yopt||	priemerné lambda	CPU time
1	6	2.1195	0.5630	38736.5	4.85E+20	7.61E-04	83.3536	0.00076	1.772
2	6	2.0616	0.7560	36467.9	5.56E+27	1.62E-04	81.4941	0.00090	1.772
3	5	2.0567	0.0203	35658.1	7.87E+21	1.80E-04	77.6496	0.00138	1.592
4	7	2.2194	0.9198	44027.8	3.76E+26	1.97E-06	92.6186	0.00073	2.013
5	6	2.2731	0.3634	45097.1	7.49E+30	1.69E-06	95.3323	0.00093	1.873
6	6	2.1875	2.6676	42499.9	6.07E+17	4.10E-04	91.7097	0.00073	1.822
7	10	2.0805	0.9167	36982.6	3.22E+33	4.47E-07	81.8425	0.00048	3.044
8	10	2.1371	0.8853	40830.7	3.74E+33	1.05E-05	86.3503	0.00048	3.195
9	5	2.1395	0.3282	39521.6	2.11E+23	8.73E-04	85.7451	0.00124	1.422
10	5	2.1718	0.2663	41396.9	1.90E+20	8.84E-04	89.3779	0.00092	1.522
11	6	2.1290	0.7562	38969.2	2.08E+26	6.49E-05	82.6941	0.00087	1.753
12	9	2.0939	0.1838	37179.5	5.64E+28	9.39E-05	82.9009	0.00061	2.844
13	6	2.1012	0.9602	37340.2	2.06E+31	2.48E-06	83.3854	0.00115	1.903
14	5	2.0412	0.1144	35106.2	4.22E+20	6.14E-04	78.4833	0.00086	1.693
15	6	2.0768	0.1078	36654.1	8.08E+29	1.44E-06	81.3631	0.00083	1.892
16	7	2.1898	0.0088	42530.1	2.24E+32	1.57E-06	88.2079	0.00061	2.324
17	7	2.1956	0.1801	41854.9	1.13E+30	8.50E-06	87.0759	0.00079	2.033
18	5	2.1565	0.0027	40104.1	3.05E+25	4.67E-06	83.8121	0.00160	1.602
19	9	2.1506	4.6132	40387.3	1.55E+32	1.21E-05	85.6962	0.00051	3.045
20	11	2.2842	1.7666	46141.5	1.45E+29	9.58E-07	94.1432	0.00049	3.615
21	6	2.1633	0.0300	40528.7	3.96E+24	3.56E-04	90.9321	0.00079	1.882
22	5	2.1500	0.0884	40488.3	9.68E+22	4.54E-04	89.3768	0.00083	1.683
23	6	2.0984	0.0119	37597.3	1.56E+28	3.14E-04	82.6407	0.00091	1.953
24	7	2.1548	2.8985	40009.7	8.70E+26	2.35E-06	87.7257	0.00081	2.063
25	6	2.1042	0.0534	38191.3	2.27E+23	5.93E-05	82.6524	0.00083	1.743
26	5	2.1190	3.6847	38931.0	3.61E+20	1.10E-04	84.8123	0.00076	1.563
27	8	2.1733	0.0029	41679.0	4.65E+37	1.44E-04	90.0248	0.00050	2.484
28	7	2.0726	0.0699	37306.6	1.29E+28	3.09E-06	81.8354	0.00059	2.032
29	12	2.0717	5.1538	36230.0	5.39E+26	7.84E-07	79.4984	0.00042	3.825
30	4	2.0149	0.0521	32466.3	1.18E+26	6.90E-05	74.2273	0.00191	1.483
31	5	2.0817	0.3200	36222.6	5.83E+23	9.36E-04	81.4311	0.00126	1.472
32	6	2.1266	0.0340	39371.1	1.55E+21	6.02E-04	86.9072	0.00077	1.662
33	6	2.1230	0.0029	37545.9	3.90E+30	5.03E-06	82.8230	0.00095	1.973
34	6	2.2749	0.0283	46550.8	1.22E+22	7.29E-05	97.3100	0.00099	1.733
35	9	2.1830	0.0278	42578.7	3.32E+32	1.07E-08	89.9320	0.00044	3.074
36	5	2.1195	0.0357	39317.6	1.88E+22	2.44E-04	85.8692	0.00119	1.452
37	6	2.1981	1.4729	42011.1	1.17E+22	2.54E-04	92.6878	0.00059	1.913
38	8	2.1203	0.5416	37983.1	7.73E+36	1.18E-06	84.3919	0.00118	2.373
39	5	2.0480	0.0675	35878.7	1.19E+24	1.46E-04	77.6254	0.00128	1.453
40	5	2.0637	0.0323	36158.1	7.91E+23	3.09E-04	80.7304	0.00097	1.612
priemer	**6.6**	**2.1332**	**0.7747**	**39363.3025**	**1.35E+36**	**2.04E-04**	**85.4167**	**0.00087**	**2.0540**

metóda: primárna

vstup: počet riešených úloh = **40**

rozmer úlohy $(n\,x\,m) = (50\,x\,100)$

vzdialenosť štartovacieho bodu x^0 od optimálneho riešenia $\hat{x}$ je $\rho = 100$

kritérium presnosti: $\|x_i(\mu_k) - \hat{x}\| < 0.001$

počet vnútorných Newtonovských iterácií $= 2$

počet iterácií metódy zlatého rezu $= 25$

veľkosť redukčného parametra $\sigma = 10$

výstup:

Úloha číslo	počet vonkajších iterácií	počiatočné mí	c*x-c*xopt	||gradient||	||hessian||	||x-xopt||	||y-yopt||	priemerné lambda	CPU time
1	12	2.1012	0.9531	38176.8	5.13E+12	2.48E-06	84.7290	0.000063	1.832
2	10	2.0412	0.1160	35193.7	1.58E+05	6.04E-04	78.5875	0.000076	1.703
3	12	2.0768	0.1594	37125.9	1.93E+11	1.67E-06	82.3711	0.000050	1.802
4	13	2.1898	0.0086	42750.8	2.44E+12	1.22E-06	88.6905	0.000051	2.284
5	13	2.1956	0.1387	41995.2	6.76E+10	7.79E-06	87.3499	0.000058	1.913
6	10	2.1565	0.0028	40698.6	1.12E+10	5.18E-06	84.4808	0.000087	1.573
7	16	2.0733	0.5438	36523.4	1.53E+12	4.85E-07	82.0366	0.000048	2.503
8	21	2.1506	3.4756	40397.3	2.02E-04	9.49E-05	85.7205	0.000033	3.626
9	19	2.2842	0.0904	46141.8	1.60E+02	2.67E-08	94.1435	0.000040	3.335
10	11	2.1633	0.0298	40606.4	4.05E+08	3.50E-04	91.0494	0.000070	1.743
11	10	2.1500	0.0925	40688.2	3.10E+07	4.62E-04	89.7051	0.000068	1.702
12	12	2.0984	0.0157	37792.8	3.72E+09	3.26E-04	83.1264	0.000069	1.792
13	13	2.1548	2.9089	40047.1	5.32E+07	2.48E-06	87.7781	0.000053	1.983
14	11	2.1042	0.0493	38227.2	2.40E+07	4.98E-05	82.7224	0.000059	1.643
15	10	2.1190	3.6846	39072.6	1.37E+05	1.10E-04	84.9647	0.000057	1.733
16	15	2.1733	3.6146	42045.1	1.37E+12	7.97E-06	90.3881	0.000045	2.313
17	13	2.0726	0.1552	37365.4	2.13E+09	3.55E-06	81.9451	0.000052	2.133
18	24	2.0717	4.6681	36230.0	1.72E-12	6.34E-07	79.4984	0.000033	3.826
19	8	2.0149	0.0510	34409.7	7.08E+13	5.27E-05	75.5182	0.000090	1.292
20	9	2.0817	0.3237	36380.2	8.84E+10	9.48E-04	81.6946	0.000072	1.462
21	11	2.1266	0.0340	39396.6	1.48E+05	6.03E-04	86.9368	0.000055	1.752
22	12	2.1230	0.0038	38319.7	8.38E+12	4.25E-06	83.9107	0.000059	1.903
23	11	2.2749	0.0272	46590.9	1.25E+06	7.66E-05	97.3653	0.000067	1.693
24	19	2.1830	2.3821	42593.3	2.03E+03	6.93E-07	89.9500	0.000031	3.024
25	9	2.1195	0.0357	39489.6	2.98E+09	2.58E-04	86.0872	0.000072	1.522
26	12	2.1981	1.1316	42046.3	2.74E+03	3.10E-04	92.7362	0.000056	1.973
27	9	2.0480	0.0657	36069.9	1.79E+11	1.59E-04	77.8717	0.000084	1.442
28	10	2.0637	0.0325	36357.5	3.04E+08	3.07E-04	80.9630	0.000064	1.582
29	9	2.0544	0.3927	35372.2	1.40E+08	2.34E-04	79.4040	0.000082	1.472
30	13	2.0187	0.0160	34357.0	1.49E+05	4.39E-04	78.2752	0.000053	2.043
31	11	2.2416	0.0579	45367.7	2.04E+06	1.02E-04	93.3970	0.000062	1.772
32	13	2.1621	0.1263	40983.2	4.88E+06	5.43E-05	87.7211	0.000047	2.013
33	11	2.1222	0.0198	39655.4	6.63E+08	2.44E-04	85.0059	0.000073	1.782
34	11	2.1198	0.1062	39243.7	1.81E+06	1.42E-04	83.0482	0.000078	1.762
35	10	2.0439	0.3742	35364.7	3.10E+08	5.20E-04	80.1873	0.000096	1.603
36	22	2.1095	0.4174	38448.5	1.54E-03	5.21E-07	82.6438	0.000040	3.605
37	10	2.1748	0.0191	41842.6	2.18E+08	4.56E-04	90.7414	0.000072	1.683
38	11	2.2324	0.6988	43878.6	1.94E+07	2.08E-05	92.6121	0.000067	1.652
39	19	2.1210	3.2575	38923.3	6.63E+03	5.80E-07	84.8057	0.000041	2.934
40	17	2.0766	1.2580	36982.5	1.34E+09	1.25E-06	80.2496	0.000045	2.624
priemer	**12.8**	**2.1272**	**0.7885**	**39328.7850**	**2.25E+12**	**1.74E-04**	**85.2603**	**0.000060**	**2.0507**

metóda: primárna

vstup: počet riešených úloh = **40**

rozmer úlohy $(n\,x\,m) = (50\,x\,100)$

vzdialenosť štartovacieho bodu x^0 od optimálneho riešenia $\hat{x}$ je $\rho = 100$

kritérium presnosti: $\|x_i(\mu_k) - \hat{x}\| < 0.001$

počet vnútorných Newtonovských iterácií $= 2$

počet iterácií metódy zlatého rezu $= 30$

veľkosť redukčného parametra $\sigma = 20$

výstup:

Úloha číslo	počet vonkajších iterácií	počiatočné mí	c*x-c*xopt	||gradient||	||hessian||	||x-xopt||	||y-yopt||	priemerné lambda	CPU time
1	10	2.1441	0.1334	40472.7	8.60E+07	0.000235	87.4929	0.000062	1.663
2	10	2.1425	0.0317	39576.7	2.48E+14	0.000663	85.0470	0.000065	1.642
3	11	2.1195	0.5615	38755.9	6.18E+07	7.43E-04	83.3847	0.000062	1.733
4	12	2.0616	0.8218	36683.3	1.07E+17	2.81E-06	81.8291	0.000056	1.953
5	10	2.0567	0.0190	35823.2	4.71E+09	9.83E-05	77.8460	0.000094	1.562
6	13	2.2194	0.9234	44058.5	2.78E+11	1.16E-06	92.6769	0.000053	2.033
7	11	2.2731	0.3951	45636.7	8.42E+17	3.13E-06	96.4469	0.000065	1.832
8	11	2.1875	0.0295	42511.4	2.07E+10	8.09E-05	91.7224	0.000051	1.833
9	9	2.1395	0.0078	39703.8	4.50E+13	1.57E-04	86.0174	0.000068	1.492
10	23	2.1245	0.9119	39823.7	5.76E-10	5.16E-07	84.7231	0.000034	3.805
11	10	2.1718	0.2456	41495.0	9.80E+07	9.20E-04	89.4930	0.000074	1.512
12	11	2.1290	0.7452	39100.4	1.97E+13	5.90E-05	82.9337	0.000059	1.822
13	12	2.1012	0.9411	38176.8	4.92E+16	8.19E-07	84.7290	0.000063	1.913
14	10	2.0412	0.1099	35193.7	2.14E+08	6.06E-04	78.5875	0.000076	1.653
15	12	2.0768	0.1231	37125.9	5.26E+11	1.74E-06	82.3711	0.000050	1.852
16	13	2.1956	0.0031	41995.2	9.19E+13	7.49E-06	87.3499	0.000058	2.063
17	10	2.1565	0.0026	40698.6	1.53E+13	1.35E-05	84.4808	0.000087	1.562
18	11	2.1633	0.0295	40606.4	5.53E+11	1.93E-04	91.0494	0.000070	1.823
19	10	2.1500	0.0891	40688.2	4.09E+10	4.48E-04	89.7051	0.000068	1.632
20	12	2.0984	0.0168	37792.8	5.06E+12	3.30E-04	83.1264	0.000069	2.013
21	11	2.1042	0.0478	38227.2	3.24E+10	3.34E-05	82.7224	0.000059	1.763
22	10	2.1190	3.6824	39072.6	1.83E+08	1.09E-04	84.9647	0.000057	1.612
23	13	2.0726	0.1017	37365.4	2.35E+13	3.35E-06	81.9451	0.000052	2.154
24	8	2.0149	0.0515	34409.7	9.63E+16	5.29E-05	75.5182	0.000090	1.312
25	9	2.0817	0.3189	36380.2	1.20E+14	9.58E-04	81.6946	0.000072	1.382
26	11	2.1266	0.0334	39396.6	2.00E+08	5.90E-04	86.9368	0.000055	1.732
27	12	2.1230	0.1285	38319.7	6.77E+15	4.41E-07	83.9107	0.000062	1.833
28	10	2.2749	0.4941	46590.9	5.29E+11	1.93E-04	97.3653	0.000070	1.682
29	9	2.1195	0.0339	39489.6	3.92E+12	2.46E-04	86.0872	0.000072	1.482
30	12	2.1981	1.1214	42046.3	3.65E+06	2.48E-04	92.7362	0.000056	2.013
31	9	2.0480	0.0614	36069.9	2.44E+14	1.59E-04	77.8717	0.000084	1.512
32	10	2.0637	0.0312	36357.5	4.14E+11	3.03E-04	80.9630	0.000064	1.662
33	9	2.0544	0.3859	35372.2	1.88E+11	2.62E-04	79.4040	0.000082	1.472
34	12	2.0187	1.0360	34357.0	8.00E+10	8.36E-04	78.2752	0.000066	1.912
35	11	2.2416	0.0550	45367.7	2.77E+09	9.84E-05	93.3970	0.000062	1.653
36	13	2.1621	0.1380	40983.2	6.64E+09	5.04E-05	87.7211	0.000047	2.003
37	11	2.1222	0.0195	39655.4	9.03E+11	2.29E-04	85.0059	0.000073	1.782
38	11	2.1198	0.1004	39243.7	2.46E+09	1.54E-04	83.0482	0.000078	1.793
39	10	2.0439	0.3739	35364.7	4.23E+11	5.11E-04	80.1873	0.000096	1.673
40	10	2.1748	0.0185	41842.6	2.99E+11	4.60E-04	90.7414	0.000072	1.602
priemer	**11.05**	**2.1259**	**0.3594**	**39295.7750**	**2.76E+16**	**2.51E-04**	**85.2877**	**0.000066**	**1.7856**

metóda: primárna

vstup: počet riešených úloh = **40**

rozmer úlohy $(n\,x\,m) = (50\,x\,100)$

vzdialenosť štartovacieho bodu x^0 od optimálneho riešenia $\hat{x}$ je $\rho = 100$

kritérium presnosti: $\|x_i(\mu_k) - \hat{x}\| < 0.001$

počet vnútorných Newtonovských iterácií $= 2$

počet iterácií metódy zlatého rezu $= 30$

veľkosť redukčného parametra $\sigma = 10$

výstup:

Úloha číslo	počet vonkajších iterácií	počiatočné mí	c*x-c*xopt	||gradient||	||hessian||	||x-xopt||	||y-yopt||	priemerné lambda	CPU time
1	10	2.1441	0.1331	40472.7	2.24E+13	2.34E-04	87.4929	0.000083	1.662
2	10	2.1425	0.0317	39576.7	6.48E+19	6.63E-04	85.0470	0.000089	1.622
3	11	2.1195	0.5616	38755.9	6.48E+13	7.43E-04	83.3847	0.000080	1.722
4	12	2.0616	0.8217	36683.3	4.54E+23	3.03E-06	81.8291	0.000075	1.923
5	10	2.0567	0.0190	35823.2	1.22E+15	9.81E-05	77.8460	0.000126	1.553
6	13	2.2194	0.9292	44058.5	4.75E+18	1.08E-06	92.6769	0.000072	2.083
7	11	2.2731	0.3944	45636.6	8.29E+23	3.11E-06	96.4468	0.000085	1.762
8	11	2.1875	0.0296	42511.4	2.17E+16	7.85E-05	91.7224	0.000066	1.793
9	9	2.1395	0.0077	39703.7	2.95E+18	1.56E-04	86.0173	0.000090	1.492
10	10	2.1718	0.2459	41495.0	2.57E+13	9.20E-04	89.4930	0.000098	1.562
11	11	2.1290	0.7464	39100.4	2.07E+19	5.90E-05	82.9337	0.000080	1.783
12	12	2.1012	0.9482	38176.8	2.06E+23	7.85E-07	84.7290	0.000084	1.913
13	10	2.0412	0.1098	35193.7	5.61E+13	6.06E-04	78.5875	0.000095	1.642
14	12	2.0768	0.1119	37125.9	2.23E+18	1.70E-06	82.3711	0.000065	1.843
15	13	2.1898	0.0086	42750.8	5.30E+22	1.96E-07	88.6905	0.000065	2.003
16	13	2.1956	0.5055	41995.2	1.52E+21	5.03E-06	87.3499	0.000077	2.073
17	10	2.1565	0.0026	40698.6	4.02E+18	1.35E-05	84.4808	0.000119	1.542
18	11	2.1633	0.0295	40606.4	5.79E+17	1.93E-04	91.0494	0.000092	1.742
19	10	2.1500	0.0890	40688.2	1.08E+16	4.47E-04	89.7051	0.000089	1.733
20	12	2.0984	0.0167	37792.8	2.12E+19	3.30E-04	83.1264	0.000092	1.922
21	11	2.1042	0.0480	38227.2	3.38E+16	3.36E-05	82.7224	0.000080	1.753
22	10	2.1190	3.6831	39072.6	4.80E+13	1.09E-04	84.9647	0.000076	1.642
23	13	2.0726	0.1565	37365.4	3.86E+20	8.84E-07	81.9451	0.000066	2.073
24	8	2.0149	0.0515	34409.1	1.58E+21	5.31E-05	75.5175	0.000125	1.282
25	9	2.0817	0.3183	36380.1	7.89E+18	9.58E-04	81.6945	0.000099	1.482
26	11	2.1266	0.0333	39396.6	2.09E+14	5.90E-04	86.9368	0.000073	1.723
27	12	2.1230	0.1400	38319.7	2.85E+22	3.96E-07	83.9107	0.000080	1.943
28	10	2.2749	0.4940	46590.9	1.39E+17	1.93E-04	97.3653	0.000094	1.612
29	9	2.1195	0.0339	39489.6	2.57E+17	2.46E-04	86.0871	0.000097	1.512
30	12	2.1981	1.1407	42046.3	1.53E+13	2.48E-04	92.7362	0.000072	1.963
31	9	2.0480	0.0616	36069.9	1.60E+19	1.57E-04	77.8716	0.000112	1.542
32	10	2.0637	0.0311	36357.5	1.08E+17	3.03E-04	80.9629	0.000088	1.603
33	9	2.0544	0.3859	35372.2	1.24E+16	2.62E-04	79.4040	0.000111	1.473
34	12	2.0187	1.0523	34357.0	3.38E+17	9.65E-04	78.2752	0.000072	1.963
35	11	2.2416	0.0550	45367.7	2.91E+15	9.84E-05	93.3970	0.000082	1.762
36	13	2.1621	0.1345	40983.2	1.12E+17	5.05E-05	87.7211	0.000061	2.043
37	11	2.1222	0.0195	39655.4	9.46E+17	2.29E-04	85.0059	0.000097	1.843
38	11	2.1198	0.1003	39243.7	2.58E+15	1.54E-04	83.0482	0.000104	1.742
39	10	2.0439	0.3705	35364.7	1.10E+17	5.12E-04	80.1873	0.000128	1.562
40	10	2.1748	0.0184	41842.5	7.70E+16	4.57E-04	90.7414	0.000097	1.683
priemer	**10.8**	**2.1275**	**0.3518**	**39368.9275**	**3.94E+22**	**2.54E-04**	**85.3869**	**0.000088**	**1.7393**

metóda: primárna

vstup: počet riešených úloh = **40**

rozmer úlohy $(n\,x\,m) = (50\,x\,100)$

vzdialenosť štartovacieho bodu x^0 od optimálneho riešenia $\hat{x}$ je $\rho = 100$

kritérium presnosti: $\|x_i(\mu_k) - \hat{x}\| < 0.001$

počet vnútorných Newtonovských iterácií $= 3$

počet iterácií metódy zlatého rezu $= 25$

veľkosť redukčného parametra $\sigma = 20$

výstup:

Úloha číslo	počet vonkajších iterácií	počiatočné mí	c*x-c*xopt	||gradient||	||hessian||	||x-xopt||	||y-yopt||	priemerné lambda	CPU time
1	11	2.0805	1.3613	36988.2	7.19E+21	7.54E-07	81.8535	0.00012	2.553
2	13	2.1371	1.8829	40835.9	1.00E+19	5.23E-07	86.3597	0.00012	3.044
3	6	2.1395	0.3269	39703.2	2.11E+18	9.23E-04	86.0165	0.00021	1.332
4	16	2.1245	0.9622	39823.7	1.34E+06	1.01E-06	84.7231	0.00010	3.695
5	7	2.1718	0.2644	41495.0	4.70E+12	8.81E-04	89.4930	0.00022	1.592
6	8	2.1290	0.7617	39100.4	1.27E+18	6.49E-05	82.9337	0.00019	1.793
7	12	2.0939	0.3114	37184.7	8.38E+16	9.32E-05	82.9096	0.00014	2.794
8	8	2.1012	0.9620	38176.7	1.30E+23	2.49E-06	84.7289	0.00019	2.003
9	7	2.0412	0.1157	35193.7	1.01E+13	6.07E-04	78.5875	0.00019	1.563
10	8	2.0768	0.0763	37125.9	5.62E+21	1.31E-06	82.3710	0.00014	1.793
11	9	2.1898	0.0087	42750.8	3.45E+23	7.40E-07	88.6905	0.00014	2.063
12	9	2.1956	0.1772	41995.1	1.72E+21	8.13E-06	87.3499	0.00017	1.982
13	7	2.1565	0.0027	40698.5	7.36E+17	5.08E-06	84.4807	0.00027	1.502
14	13	2.1506	4.3979	40397.3	1.59E+15	8.75E-06	85.7205	0.00010	2.974
15	8	2.1633	0.0298	40606.4	2.56E+16	3.51E-04	91.0494	0.00020	1.933
16	7	2.1500	0.0911	40688.2	2.13E+15	4.58E-04	89.7050	0.00020	1.973
17	8	2.0984	0.0151	37792.8	9.52E+19	3.26E-04	83.1264	0.00020	1.822
18	9	2.1548	2.9056	40047.1	1.35E+18	2.42E-06	87.7781	0.00014	2.033
19	7	2.1042	0.0505	38227.2	5.94E+17	5.24E-05	82.7223	0.00018	1.713
20	7	2.1190	3.6895	39072.6	8.78E+12	1.10E-04	84.9647	0.00017	1.903
21	16	2.0743	6.9881	36930.5	2.62E+11	1.98E-06	80.1561	0.00008	4.186
22	10	2.1733	3.6157	42045.1	1.72E+26	1.09E-05	90.3880	0.00013	2.835
23	9	2.0726	0.1336	37365.4	3.98E+20	3.30E-06	81.9451	0.00014	2.023
24	15	2.0717	5.3976	36230.0	3.34E+10	7.48E-07	79.4984	0.00011	3.795
25	6	2.0149	0.0508	34409.1	1.14E+19	5.30E-05	75.5173	0.00029	1.292
26	6	2.0817	0.3205	36379.7	5.67E+18	9.46E-04	81.6937	0.00023	1.392
27	7	2.1266	0.0339	39396.6	3.79E+15	5.98E-04	86.9367	0.00017	1.662
28	8	2.1230	0.0043	38319.6	2.22E+23	4.63E-06	83.9105	0.00017	1.973
29	7	2.2749	0.0274	46590.8	3.18E+16	7.66E-05	97.3652	0.00020	1.793
30	12	2.1830	1.6846	42593.3	7.63E+16	4.85E-07	89.9500	0.00009	2.784
31	6	2.1195	0.0357	39489.1	1.89E+17	2.56E-04	86.0865	0.00022	1.402
32	8	2.1981	1.2662	42046.3	7.03E+13	3.00E-04	92.7362	0.00015	1.903
33	9	2.1203	0.5420	38351.2	6.75E+28	1.17E-06	84.9763	0.00016	2.183
34	6	2.0480	0.0666	36069.3	1.16E+19	1.52E-04	77.8709	0.00026	1.392
35	7	2.0637	0.0322	36357.5	1.94E+16	3.09E-04	80.9629	0.00020	1.612
36	10	2.0725	2.4397	36766.2	1.57E+26	1.74E-06	81.9329	0.00013	2.243
37	6	2.0544	0.3925	35371.9	8.99E+15	2.34E-04	79.4035	0.00025	1.602
38	9	2.0187	0.0096	34357.0	1.68E+16	2.34E-05	78.2752	0.00014	2.123
39	7	2.2416	0.0578	45367.6	5.25E+16	1.02E-04	93.3969	0.00019	1.662
40	9	2.1621	0.1147	40983.2	1.26E+17	5.51E-05	87.7211	0.00013	2.003
priemer	**8.825**	**2.1218**	**1.0402**	**39083.0700**	**1.70E+27**	**1.76E-04**	**84.9072**	**0.00017**	**2.0980**

metóda: primárna

vstup: počet riešených úloh = **40**

rozmer úlohy $(n\,x\,m) = (50\,x\,100)$

vzdialenosť štartovacieho bodu x^0 od optimálneho riešenia $\hat{x}$ je $\rho = 100$

kritérium presnosti: $\|x_i(\mu_k) - \hat{x}\| < 0.001$

počet vnútorných Newtonovských iterácií $= 3$

počet iterácií metódy zlatého rezu $= 30$

veľkosť redukčného parametra $\sigma = 10$

výstup:

Úloha číslo	počet vonkajších iterácií	počiatočné mí	c*x-c*xopt	||gradient||	||hessian||	||x-xopt||	||y-yopt||	priemerné lambda	CPU time
1	7	2.1441	0.1268	40469.3	2.15E+19	2.29E-04	87.4896	0.00030	1.772
2	7	2.1425	0.0323	39560.9	6.40E+25	6.63E-04	85.0112	0.00037	2.483
3	7	2.1195	0.5623	38750.6	6.49E+21	7.43E-04	83.3768	0.00026	1.953
4	8	2.0616	0.8208	36659.0	3.52E+30	2.94E-06	81.8062	0.00027	1.982
5	7	2.0567	0.0189	35819.1	1.15E+21	9.88E-05	77.8426	0.00047	1.612
6	9	2.2194	0.9069	44057.5	4.94E+26	1.38E-06	92.6750	0.00026	2.144
7	8	2.2731	0.3872	45626.2	1.83E+28	3.17E-06	96.4283	0.00030	1.743
8	8	2.1875	0.0295	42510.4	2.18E+22	8.20E-05	91.7214	0.00023	1.763
9	6	2.1395	0.3246	39650.9	2.89E+24	9.65E-04	85.9435	0.00036	1.412
10	7	2.1718	0.2471	41492.1	2.59E+19	9.14E-04	89.4900	0.00034	1.612
11	8	2.1290	0.7549	39097.0	2.07E+25	5.89E-05	82.9280	0.00031	1.841
12	8	2.1012	0.9654	38142.8	2.07E+31	8.93E-07	84.6726	0.00032	1.943
13	7	2.0412	0.1094	35191.3	5.63E+19	6.09E-04	78.5852	0.00030	1.662
14	8	2.0768	0.0310	37116.0	5.66E+26	1.28E-06	82.3526	0.00023	2.023
15	11	2.1898	0.0059	42750.7	6.52E+27	2.24E-07	88.6903	0.00018	2.554
16	9	2.1956	0.5175	41991.7	1.52E+29	8.98E-06	87.3434	0.00027	3.024
17	7	2.1565	0.0026	40683.5	4.13E+24	1.33E-05	84.4666	0.00045	1.582
18	8	2.1633	0.0296	40604.5	5.63E+23	1.93E-04	91.0471	0.00030	1.842
19	7	2.1500	0.0867	40682.3	1.21E+22	4.42E-04	89.6956	0.00030	1.622
20	8	2.0984	0.0154	37787.4	2.12E+27	3.28E-04	83.1123	0.00032	1.923
21	12	2.1548	2.0180	40047.1	2.36E+27	1.59E-07	87.7781	0.00017	2.883
22	7	2.1042	0.0496	38216.5	3.24E+24	3.52E-05	82.7019	0.00029	1.752
23	7	2.1190	3.6886	39068.6	4.82E+19	1.09E-04	84.9611	0.00027	1.632
24	9	2.0726	0.1277	37363.6	3.27E+28	8.73E-07	81.9403	0.00021	2.173
25	6	2.0149	0.0511	34356.6	1.59E+25	5.26E-05	75.4511	0.00051	1.302
26	6	2.0817	0.3165	36333.7	7.89E+24	9.60E-04	81.6185	0.00039	1.491
27	7	2.1266	0.0332	39389.6	2.10E+22	5.83E-04	86.9304	0.00027	1.712
28	8	2.1230	0.1833	38301.7	2.80E+30	2.45E-07	83.8709	0.00028	1.862
29	7	2.2749	0.4926	46581.2	1.38E+23	1.94E-04	97.3527	0.00034	1.662
30	6	2.1195	0.0340	39444.0	2.54E+23	2.45E-04	86.0400	0.00037	1.453
31	8	2.1981	1.3230	42045.2	1.54E+21	2.34E-04	92.7348	0.00023	1.942
32	6	2.0480	0.0627	36014.0	1.63E+25	1.46E-04	77.8042	0.00042	1.433
33	7	2.0637	0.0309	36351.9	1.08E+23	3.04E-04	80.9572	0.00033	1.603
34	6	2.0544	0.3859	35336.6	1.24E+22	2.62E-04	79.3615	0.00040	1.492
35	7	2.2416	0.0550	45359.4	2.92E+23	9.83E-05	93.3866	0.00029	1.733
36	9	2.1621	0.1139	40982.6	1.13E+25	5.15E-05	87.7196	0.00021	2.073
37	8	2.1222	0.0196	39653.5	9.49E+23	2.29E-04	85.0029	0.00033	1.803
38	7	2.1198	0.1002	39236.7	2.59E+23	1.55E-04	83.0373	0.00036	1.732
39	7	2.0439	0.3470	35356.5	1.07E+23	5.31E-04	80.1770	0.00047	1.542
40	7	2.1748	2.5134	41835.9	6.71E+22	8.73E-04	90.7348	0.00039	1.582
priemer	**7.55**	**2.1309**	**0.4480**	**39497.9525**	**6.82E+29**	**2.61E-04**	**85.6060**	**0.00032**	**1.8337**

metóda: primárna

vstup: počet riešených úloh = **40**

rozmer úlohy $(n\,x\,m) = (50\,x\,100)$

vzdialenosť štartovacieho bodu x^0 od optimálneho riešenia $\hat{x}$ je $\rho = 100$

kritérium presnosti: $\|x_i(\mu_k) - \hat{x}\| < 0.001$

počet vnútorných Newtonovských iterácií $= 3$

počet iterácií metódy zlatého rezu $= 30$

veľkosť redukčného parametra $\sigma = 20$

výstup:

Úloha číslo	počet vonkajších iterácií	počiatočné mí	c*x-c*xopt	||gradient||	||hessian||	||x-xopt||	||y-yopt||	priemerné lambda	CPU time
1	7	2.1441	0.1308	40472.7	5.35E+15	2.32E-04	87.4928	0.00019	1.663
2	7	2.1425	0.0320	39576.5	1.57E+22	6.61E-04	85.0465	0.00021	1.602
3	7	2.1195	0.5622	38755.8	1.58E+18	7.43E-04	83.3845	0.00017	1.683
4	8	2.0616	0.8211	36683.0	6.12E+27	2.84E-06	81.8287	0.00017	2.083
5	7	2.0567	0.0190	35823.1	2.89E+17	9.74E-05	77.8459	0.00029	1.592
6	9	2.2194	0.9231	44058.5	6.93E+21	1.19E-06	92.6769	0.00016	2.043
7	8	2.2731	0.3897	45636.5	5.06E+25	3.00E-06	96.4466	0.00019	1.882
8	8	2.1875	0.0298	42511.4	1.33E+18	6.75E-05	91.7224	0.00014	1.812
9	6	2.1395	0.3242	39702.1	2.83E+21	9.77E-04	86.0151	0.00021	1.823
10	15	2.1245	1.0900	39823.7	1.71E+13	3.93E-08	84.7231	0.00010	6.069
11	7	2.1718	0.2469	41494.9	6.32E+15	9.17E-04	89.4930	0.00022	1.773
12	8	2.1290	0.7525	39100.4	1.26E+21	5.89E-05	82.9337	0.00019	1.822
13	8	2.1012	0.9476	38176.5	1.26E+27	7.35E-07	84.7285	0.00019	1.952
14	7	2.0412	0.1095	35193.7	1.37E+16	6.08E-04	78.5875	0.00019	1.642
15	8	2.0768	0.0572	37125.8	2.21E+25	1.45E-06	82.3708	0.00014	1.893
16	10	2.1898	0.0086	42750.8	1.92E+23	2.66E-07	88.6905	0.00012	2.524
17	9	2.1956	0.3872	41995.1	2.29E+24	9.19E-06	87.3498	0.00017	2.043
18	7	2.1565	0.0026	40698.4	9.99E+20	1.33E-05	84.4805	0.00027	1.552
19	8	2.1633	0.0295	40606.4	3.49E+19	1.93E-04	91.0494	0.00020	1.823
20	7	2.1500	0.0879	40688.1	2.81E+18	4.45E-04	89.7049	0.00020	1.733
21	8	2.0984	0.0161	37792.8	1.30E+23	3.29E-04	83.1263	0.00020	1.953
22	10	2.1548	0.1725	40047.1	2.76E+27	5.41E-08	87.7781	0.00012	2.324
23	7	2.1042	0.0489	38227.1	8.02E+20	3.45E-05	82.7221	0.00018	1.722
24	7	2.1190	3.6874	39072.6	1.17E+16	1.09E-04	84.9646	0.00017	1.713
25	9	2.0726	0.1241	37365.4	5.34E+23	2.81E-07	81.9451	0.00014	2.133
26	6	2.0149	0.0513	34408.0	1.54E+22	5.38E-05	75.5161	0.00029	1.341
27	6	2.0817	0.3157	36378.7	7.73E+21	9.57E-04	81.6921	0.00023	1.452
28	7	2.1266	0.0332	39396.5	5.12E+18	5.85E-04	86.9367	0.00017	1.773
29	8	2.1230	0.1650	38319.6	1.77E+26	1.23E-07	83.9104	0.00018	2.063
30	7	2.2749	0.4938	46590.8	3.37E+19	1.88E-04	97.3651	0.00021	1.612
31	6	2.1195	0.0339	39488.2	2.49E+20	2.45E-04	86.0857	0.00022	1.452
32	8	2.1981	1.2548	42046.3	9.37E+16	2.41E-04	92.7362	0.00015	1.993
33	6	2.0480	0.0624	36068.2	1.58E+22	1.50E-04	77.8695	0.00026	1.442
34	7	2.0637	0.0310	36357.4	2.64E+19	3.04E-04	80.9629	0.00020	1.632
35	6	2.0544	0.3858	35371.1	1.21E+19	2.62E-04	79.4027	0.00025	1.492
36	9	2.0187	0.0105	34357.0	5.13E+20	6.19E-05	78.2752	0.00015	2.113
37	7	2.2416	0.0550	45367.6	7.12E+19	9.84E-05	93.3968	0.00019	1.702
38	9	2.1621	0.1251	40983.2	1.72E+20	5.11E-05	87.7211	0.00013	2.033
39	8	2.1222	0.0195	39655.3	5.78E+19	2.29E-04	85.0059	0.00021	1.843
40	7	2.0439	0.3568	35364.5	2.64E+19	5.20E-04	80.1871	0.00029	1.532
priemer	**7.725**	**2.1271**	**0.3599**	**39338.2700**	**2.60E+26**	**2.36E-04**	**85.3543**	**0.00019**	**1.9082**

metóda: primárna

vstup: počet riešených úloh = **40**

rozmer úlohy $(n\,x\,m) = (50\,x\,100)$

vzdialenosť štartovacieho bodu x^0 od optimálneho riešenia $\hat{x}$ je $\rho = 100$

kritérium presnosti: $\|x_i(\mu_k) - \hat{x}\| < 0.001$

počet vnútorných Newtonovských iterácií $= 4$

počet iterácií metódy zlatého rezu $= 25$

veľkosť redukčného parametra $\sigma = 20$

výstup:

Úloha číslo	počet vonkajších iterácií	počiatočné mí	c*x-c*xopt	||gradient||	||hessian||	||x-xopt||	||y-yopt||	priemerné lambda	CPU time
1	5	2.1441	1.4573	40466.1	5.86E+17	8.80E-04	87.4853	0.00052	1.542
2	5	2.1425	0.0364	39544.2	1.84E+24	7.46E-04	84.9738	0.00060	1.583
3	6	2.1195	0.5651	38755.3	4.74E+17	7.63E-04	83.3837	0.00045	1.662
4	6	2.0616	0.7597	36676.5	5.45E+24	1.55E-04	81.8183	0.00049	1.782
5	5	2.0567	0.0202	35812.8	3.24E+19	2.00E-04	77.8335	0.00075	1.553
6	7	2.2194	0.9205	44058.0	6.63E+22	2.21E-06	92.6759	0.00042	1.963
7	6	2.2731	0.3768	45619.7	5.84E+27	2.05E-06	96.4088	0.00051	1.783
8	6	2.1875	2.6648	42511.0	5.84E+14	6.20E-04	91.7220	0.00041	1.693
9	5	2.1395	0.3282	39692.3	8.37E+20	8.81E-04	85.9998	0.00061	1.352
10	5	2.1718	0.2655	41488.8	7.51E+17	8.79E-04	89.4857	0.00054	1.522
11	6	2.1290	0.7645	39096.3	2.03E+23	6.48E-05	82.9260	0.00049	1.743
12	6	2.1012	0.9742	38150.4	2.08E+28	2.50E-06	84.6816	0.00055	1.773
13	5	2.0412	0.1149	35188.3	1.63E+18	6.12E-04	78.5810	0.00049	1.572
14	6	2.0768	0.0584	37110.6	9.29E+26	1.21E-06	82.3365	0.00041	1.903
15	7	2.1898	0.0088	42747.4	5.49E+28	8.29E-07	88.6828	0.00034	2.113
16	7	2.1956	0.1489	41993.0	2.75E+26	7.96E-06	87.3457	0.00043	1.983
17	5	2.1565	0.0027	40661.3	1.21E+23	4.80E-06	84.4358	0.00076	1.433
18	8	2.0733	1.9079	36521.5	3.20E+31	1.70E-06	82.0338	0.00036	2.293
19	9	2.1506	4.5923	40397.3	1.83E+26	8.57E-06	85.7204	0.00028	2.874
20	10	2.2842	0.5822	46141.8	5.24E+25	1.59E-07	94.1435	0.00029	3.185
21	6	2.1633	0.0298	40603.9	4.00E+21	3.53E-04	91.0457	0.00047	1.723
22	8	2.1991	0.5242	43278.6	3.04E+33	9.03E-07	89.5394	0.00033	2.383
23	5	2.1500	0.0899	40675.8	3.58E+20	4.57E-04	89.6843	0.00049	1.583
24	6	2.0984	0.0139	37786.7	1.52E+25	3.23E-04	83.1108	0.00052	1.803
25	7	2.1548	2.9000	40046.6	2.11E+23	2.35E-06	87.7773	0.00040	2.013
26	6	2.1042	0.0523	38226.1	2.26E+20	5.68E-05	82.7202	0.00046	1.642
27	5	2.1190	3.6918	39063.8	1.41E+18	1.10E-04	84.9551	0.00042	1.573
28	12	2.0743	6.9430	36930.5	1.44E+22	1.97E-06	80.1561	0.00021	3.765
29	8	2.1733	3.6165	42044.2	9.03E+29	1.16E-05	90.3865	0.00030	2.393
30	7	2.0726	0.0975	37364.3	5.29E+25	3.29E-06	81.9424	0.00034	2.103
31	12	2.0717	5.3785	36230.0	1.95E+18	7.31E-07	79.4984	0.00026	3.595
32	4	2.0149	0.0495	34165.7	1.84E+24	7.48E-05	75.1941	0.00086	1.312
33	5	2.0817	0.3155	36370.3	2.29E+21	9.36E-04	81.6774	0.00068	1.442
34	6	2.1266	0.0338	39395.8	1.52E+18	5.95E-04	86.9358	0.00044	1.723
35	6	2.1230	0.0034	38296.0	4.28E+27	5.25E-06	83.8765	0.00046	1.953
36	6	2.2749	0.0274	46589.7	1.25E+19	7.52E-05	97.3635	0.00053	1.733
37	10	2.1830	1.6488	42593.3	2.23E+27	4.77E-07	89.9500	0.00022	3.175
38	8	2.1278	1.7093	39322.2	8.96E+32	9.13E-07	85.1216	0.00033	2.564
39	5	2.1195	0.0357	39478.8	7.50E+19	2.49E-04	86.0733	0.00060	1.462
40	6	2.1981	1.4323	42045.2	1.14E+19	2.67E-04	92.7347	0.00037	1.943
priemer	**6.575**	**2.1386**	**1.1286**	**39828.5025**	**9.93E+31**	**2.34E-04**	**86.0604**	**0.00046**	**1.9798**

metóda: primárna

vstup: počet riešených úloh = **40**

rozmer úlohy $(n\,x\,m) = (50\,x\,100)$

vzdialenosť štartovacieho bodu x^0 od optimálneho riešenia $\hat{x}$ je $\rho = 100$

kritérium presnosti: $\|x_i(\mu_k) - \hat{x}\| < 0.001$

počet vnútorných Newtonovských iterácií $= 4$

počet iterácií metódy zlatého rezu $= 30$

veľkosť redukčného parametra $\sigma = 10$

výstup:

Úloha číslo	počet vonkajších iterácií	počiatočné mí	c*x-c*xopt	||gradient||	||hessian||	||x-xopt||	||y-yopt||	priemerné lambda	CPU time
1	5	2.1441	0.3843	40098.1	2.53E+23	1.05E-04	87.1453	0.00092	1.713
2	5	2.1425	0.0326	38002.1	6.36E+29	6.87E-04	82.5134	0.00117	1.662
3	6	2.1195	0.5593	38702.8	6.49E+23	7.40E-04	83.3066	0.00076	1.771
4	6	2.0616	0.8185	32981.8	1.05E+36	3.11E-06	82.6179	0.00090	2.003
5	5	2.0567	0.0189	35416.3	1.05E+25	6.66E-05	77.5346	0.00138	1.611
6	7	2.2194	0.9242	43944.9	9.91E+30	1.06E-06	92.4680	0.00073	2.093
7	6	2.2731	0.3651	43722.9	1.60E+34	2.46E-06	92.9842	0.00093	1.883
8	6	2.1875	0.0273	42416.2	2.20E+26	1.80E-04	91.6201	0.00070	1.833
9	5	2.1395	0.3254	39180.3	2.83E+26	9.33E-04	85.3683	0.00124	1.372
10	5	2.1718	0.2485	41206.9	2.55E+23	9.20E-04	89.2047	0.00092	1.583
11	6	2.1290	0.7465	38764.9	2.07E+29	5.90E-05	82.4045	0.00087	1.853
12	6	2.1012	0.9509	34795.8	2.05E+35	8.64E-07	83.0245	0.00110	2.023
13	5	2.0412	0.1083	34955.9	5.70E+23	6.14E-04	78.3690	0.00086	1.672
14	6	2.0768	0.0762	36353.7	2.17E+30	1.55E-06	81.0879	0.00083	1.882
15	7	2.1956	0.3220	41646.6	1.51E+33	6.11E-06	86.7376	0.00073	2.253
16	5	2.1565	0.0026	39195.4	4.14E+28	1.39E-05	83.6431	0.00160	1.572
17	6	2.1633	0.0297	40417.0	5.41E+27	1.94E-04	90.8313	0.00079	1.842
18	5	2.1500	0.0855	40095.8	1.27E+26	4.41E-04	88.8307	0.00083	1.683
19	6	2.0984	0.0127	37247.1	2.12E+31	3.19E-04	81.8786	0.00091	1.943
20	9	2.1548	0.0573	40008.4	2.60E+34	6.33E-08	87.7225	0.00069	2.803
21	6	2.1042	0.0518	38121.1	3.06E+26	3.80E-05	82.5228	0.00083	1.803
22	5	2.1190	3.6827	38669.3	4.83E+23	1.09E-04	84.6215	0.00076	1.682
23	8	2.0726	0.0447	37294.6	1.19E+33	7.50E-07	81.7802	0.00053	2.433
24	4	2.0149	0.0516	29086.6	1.62E+29	6.46E-05	84.5452	0.00190	1.342
25	5	2.0817	0.3150	35916.6	7.94E+26	9.47E-04	81.0456	0.00126	1.522
26	6	2.1266	0.0333	39327.0	2.09E+24	5.89E-04	86.8744	0.00077	1.782
27	6	2.1230	0.0036	31795.0	1.28E+38	5.49E-07	94.7811	0.00096	1.973
28	5	2.2749	0.4860	45633.1	1.27E+27	2.44E-04	96.3004	0.00104	1.722
29	5	2.1195	0.0339	39048.6	2.47E+25	2.41E-04	85.6543	0.00118	1.462
30	6	2.1981	1.4584	41941.4	1.56E+25	2.04E-04	92.6034	0.00059	2.023
31	5	2.0480	0.0631	35512.4	1.63E+27	1.43E-04	77.3374	0.00128	1.582
32	5	2.0637	0.0311	35797.9	1.08E+27	3.04E-04	80.5332	0.00097	1.693
33	5	2.0544	0.3863	35015.8	1.24E+24	2.61E-04	78.9987	0.00112	1.582
34	7	2.0187	0.0103	34289.0	3.36E+29	6.30E-05	78.1326	0.00064	2.304
35	6	2.2416	0.0550	45285.0	2.91E+25	9.82E-05	93.2952	0.00074	1.752
36	7	2.1621	0.0781	40925.6	1.16E+29	5.11E-05	87.5644	0.00061	2.254
37	6	2.1222	0.0197	39474.8	9.48E+27	2.33E-04	84.7244	0.00089	1.832
38	7	2.1198	0.1000	39174.0	2.58E+25	1.52E-04	82.9412	0.00101	1.773
39	5	2.0439	0.3220	34553.7	1.03E+27	5.61E-04	79.3438	0.00145	1.603
40	5	2.1748	2.6308	41196.3	5.59E+26	8.01E-04	90.2286	0.00129	1.612
priemer	**5.775**	**2.1267**	**0.3988**	**38430.2675**	**3.23E+36**	**2.60E-04**	**85.3280**	**0.00097**	**1.8195**

metóda: primárna

vstup: počet riešených úloh = **40**

rozmer úlohy $(n\,x\,m) = (50\,x\,100)$

vzdialenosť štartovacieho bodu x^0 od optimálneho riešenia $\hat{x}$ je $\rho = 100$

kritérium presnosti: $\|x_i(\mu_k) - \hat{x}\| < 0.001$

počet vnútorných Newtonovských iterácií $= 4$

počet iterácií metódy zlatého rezu $= 30$

veľkosť redukčného parametra $\sigma = 20$

výstup:

Úloha číslo	počet vonkajších iterácií	počiatočné mí	c*x-c*xopt	||gradient||	||hessian||	||x-xopt||	||y-yopt||	priemerné lambda	CPU time
1	5	2.1441	0.0930	40451.9	8.03E+20	2.98E-04	87.4725	0.00050	1.682
2	5	2.1425	0.0324	39478.0	2.49E+27	6.77E-04	84.8265	0.00060	1.642
3	6	2.1195	0.5615	38754.2	6.34E+20	7.42E-04	83.3822	0.00045	1.773
4	6	2.0616	0.8202	36606.9	3.53E+31	3.02E-06	81.7585	0.00045	2.163
5	5	2.0567	0.0188	35797.4	4.31E+22	1.04E-04	77.8246	0.00075	1.673
6	7	2.2194	0.9391	44056.7	1.49E+27	8.31E-07	92.6734	0.00042	2.033
7	6	2.2731	0.3788	45580.4	1.09E+31	2.86E-06	96.3111	0.00051	1.993
8	6	2.1875	0.0287	42508.4	2.13E+23	1.22E-04	91.7191	0.00039	1.812
9	5	2.1395	0.3254	39670.9	1.12E+24	9.40E-04	85.9711	0.00061	1.392
10	5	2.1718	0.2477	41476.9	1.01E+21	9.14E-04	89.4742	0.00054	1.582
11	6	2.1290	0.7550	39089.9	2.02E+26	5.89E-05	82.9159	0.00049	1.812
12	6	2.1012	0.9633	38070.6	2.01E+32	8.26E-07	84.5553	0.00055	1.953
13	5	2.0412	0.1087	35178.8	2.20E+21	6.12E-04	78.5731	0.00049	1.652
14	6	2.0768	0.5776	37079.5	3.54E+30	2.13E-06	82.2576	0.00041	1.913
15	7	2.1898	0.0068	42733.9	1.69E+32	7.07E-08	88.6622	0.00033	2.283
16	7	2.1956	0.4991	41988.8	2.03E+30	7.91E-07	87.3361	0.00043	2.103
17	5	2.1565	0.0025	40603.9	1.65E+26	1.28E-05	84.3938	0.00076	1.583
18	6	2.1633	0.0295	40600.4	5.46E+24	1.94E-04	91.0422	0.00047	1.803
19	5	2.1500	0.0869	40651.4	4.72E+23	4.43E-04	89.6463	0.00049	1.622
20	6	2.0984	0.0147	37775.7	2.07E+28	3.26E-04	83.0823	0.00052	1.953
21	8	2.1548	1.1182	40044.8	4.49E+31	1.08E-07	87.7742	0.00033	3.024
22	6	2.1042	0.0507	38223.9	3.05E+23	3.67E-05	82.7160	0.00046	2.193
23	5	2.1190	3.6898	39047.4	1.88E+21	1.09E-04	84.9420	0.00042	1.622
24	7	2.0726	0.0863	37362.6	7.15E+28	7.59E-07	81.9378	0.00034	2.133
25	4	2.0149	0.0502	33745.4	2.51E+27	5.51E-05	74.9122	0.00086	1.332
26	5	2.0817	0.3105	36351.1	3.12E+24	9.46E-04	81.6467	0.00068	1.443
27	6	2.1266	0.0332	39394.4	2.05E+21	5.82E-04	86.9348	0.00044	1.702
28	7	2.1230	0.0082	38314.4	2.12E+30	8.32E-07	83.9014	0.00041	2.223
29	5	2.2749	0.4903	46530.5	5.22E+24	2.20E-04	97.2876	0.00056	1.653
30	5	2.1195	0.0339	39461.1	9.88E+22	2.43E-04	86.0576	0.00060	1.522
31	6	2.1981	1.4187	42043.0	1.51E+22	2.15E-04	92.7320	0.00037	2.053
32	5	2.0480	0.0629	36035.0	6.39E+24	1.43E-04	77.8291	0.00069	1.502
33	5	2.0637	0.0307	36322.5	4.23E+24	3.05E-04	80.9277	0.00054	1.652
34	5	2.0544	0.3856	35349.9	4.83E+21	2.62E-04	79.3774	0.00064	1.512
35	7	2.0187	0.0159	34355.9	8.12E+25	7.30E-05	78.2729	0.00037	2.103
36	6	2.2416	0.0549	45365.1	2.85E+22	9.83E-05	93.3938	0.00045	1.713
37	7	2.1621	0.0986	40982.3	2.81E+25	5.18E-05	87.7187	0.00035	2.083
38	6	2.1222	0.0196	39649.7	9.23E+24	2.29E-04	84.9967	0.00053	1.862
39	6	2.1198	0.1002	39241.5	2.53E+22	1.57E-04	83.0448	0.00061	2.284
40	5	2.1748	0.3377	35313.3	4.23E+24	5.41E-04	80.1230	0.00077	1.673
priemer	**5.775**	**2.1303**	**0.3721**	**39282.2100**	**1.17E+31**	**2.43E-04**	**85.2601**	**0.00051**	**1.8427**

metóda: primárno - duálna

vstup: počet riešených úloh = **40**

rozmer úlohy $(n\,x\,m) = (50\,x\,100)$

vzdialenosť štartovacieho bodu x^0 od optimálneho riešenia $\hat{x}$ je $\rho = 100$

kritérium presnosti: $\|x_i(\mu_k) - \hat{x}\| < 0.001$

počet vnútorných Newtonovských iterácií $= 3$

počet iterácií metódy zlatého rezu $= 25$

veľkosť redukčného parametra $\sigma = 3$

výstup:

Úloha číslo	počet vonkajších iterácií	počiatočné mí	||rc||	||rmi||	||x-xopt||	||y-yopt||	||z-yopt||	priemerné lambda	CPU time
1	15	2.1198	25.0884	4.5020	3.89E-04	6.2218	0.0155	0.1488	10.726
2	16	2.0976	9.0559	1.7317	2.61E-05	7.3224	0.0014	0.1594	10.975
3	16	2.2048	22.9670	3.7753	8.17E-04	11.3932	0.0340	0.1487	10.966
4	11	2.0439	11.3897	1.8053	2.27E-05	1.6477	0.0010	0.2346	7.421
5	14	2.1095	8.7354	1.4702	1.42E-04	12.0437	0.0052	0.1865	9.573
6	15	2.0977	0.3532	0.0539	4.69E-07	4.3440	0.0000	0.2213	10.505
7	16	2.0909	2.5515	0.4226	1.56E-05	3.5110	0.0007	0.1912	10.975
8	12	2.0441	8.6474	1.4885	4.29E-04	8.3027	0.0187	0.2180	8.332
9	19	2.1662	8.3136	1.5983	9.70E-04	9.9823	0.0402	0.1373	13.37
10	12	2.1643	9.7165	1.7621	8.07E-05	2.3413	0.0032	0.2138	8.602
11	16	2.1390	22.1523	4.2975	8.14E-04	9.1807	0.0350	0.1409	11.456
12	12	2.1748	12.6715	2.2207	1.05E-04	4.8177	0.0049	0.2013	8.352
13	17	2.1178	2.1370	0.3466	1.53E-05	8.7638	0.0007	0.1783	11.707
14	13	2.2324	15.0672	2.9369	4.35E-05	8.2660	0.0023	0.1841	9.313
15	12	2.0847	24.9463	4.5589	3.60E-04	10.8094	0.0149	0.1927	8.352
16	11	2.1258	9.4402	1.6381	4.46E-04	3.9386	0.0186	0.2257	7.892
17	12	2.0826	17.2782	2.8954	1.48E-04	3.4955	0.0069	0.2052	8.111
18	11	2.1198	11.4527	1.9060	4.07E-05	5.0159	0.0014	0.2224	7.872
19	16	2.1176	8.1900	1.3522	4.28E-05	3.7831	0.0019	0.1654	10.996
20	14	2.1210	9.8134	1.8592	1.58E-04	2.6153	0.0073	0.1849	9.794
21	15	2.0988	4.2498	0.6821	8.28E-05	3.3603	0.0028	0.1847	10.735
22	12	2.0774	3.2761	0.5935	9.33E-04	3.0592	0.0497	0.2342	8.362
23	14	2.0915	7.7177	1.1987	1.37E-05	9.2242	0.0005	0.1896	9.554
24	14	2.0389	1.7319	0.2777	3.98E-05	9.0455	0.0020	0.2085	9.754
25	14	2.0766	5.2576	0.8221	9.51E-06	5.6602	0.0003	0.2014	9.554
26	14	2.1307	8.0909	1.5285	3.37E-06	3.2325	0.0001	0.1873	10.024
27	17	2.0210	2.0992	0.3399	3.72E-04	7.5026	0.0180	0.1715	11.957
28	14	2.1408	11.4050	2.1494	3.48E-04	10.8894	0.0152	0.1843	9.564
29	18	2.0638	3.4811	0.7271	1.45E-05	3.7978	0.0006	0.1526	12.618
30	13	2.1315	20.6812	3.6458	1.34E-04	14.9517	0.0066	0.1757	9.303
31	12	2.1366	18.2787	2.9877	6.43E-04	3.4239	0.0216	0.2097	8.132
32	9	2.1213	19.6465	2.8266	6.44E-04	2.2653	0.0234	0.2593	6.439
33	12	2.0803	17.0097	2.8617	5.29E-04	3.6536	0.0230	0.2008	8.583
34	11	2.0801	6.9388	0.9853	2.36E-04	6.6460	0.0096	0.2352	7.871
35	13	2.0553	6.7182	1.2476	2.00E-04	1.0505	0.0090	0.2143	8.843
36	13	2.0789	3.2970	0.4822	9.40E-04	25.2339	0.0389	0.2126	9.063
37	12	2.2068	11.2941	1.7301	4.47E-06	3.2788	0.0002	0.2174	8.122
38	13	2.1763	10.8329	1.7238	4.92E-04	4.7224	0.0205	0.1934	9.073
39	14	2.1345	19.8874	4.1296	2.04E-04	7.5025	0.0093	0.1716	9.794
40	9	2.1102	23.3049	3.9700	9.91E-04	6.2556	0.0398	0.2489	6.199
priemer	**13.575**	**2.1126**	**11.1291**	**1.9383**	**2.98E-04**	**6.5638**	**0.0126**	**0.1953**	**9.4709**

metóda: primárno - duálna

vstup: počet riešených úloh = **40**

rozmer úlohy $(n\,x\,m) = (50\,x\,100)$

vzdialenosť štartovacieho bodu x^0 od optimálneho riešenia $\hat{x}$ je $\rho = 100$

kritérium presnosti: $\|x_i(\mu_k) - \hat{x}\| < 0.001$

počet vnútorných Newtonovských iterácií $= 3$

počet iterácií metódy zlatého rezu $= 30$

veľkosť redukčného parametra $\sigma = 3$

výstup:

Úloha číslo	počet vonkajších iterácií	počiatočné mí	||rc||	||rmi||	||x-xopt||	||y-yopt||	||z-yopt||	priemerné lambda	CPU time
1	15	2.1198	25.0951	4.4986	3.88E-04	6.2542	0.0154	0.1521	10.666
2	14	2.0976	9.1721	1.7503	2.31E-05	7.3409	0.0013	0.1833	10.054
3	15	2.2048	22.2800	3.6350	4.25E-04	9.2733	0.0180	0.1587	10.896
4	11	2.0439	9.8272	1.5369	1.60E-05	1.4280	0.0007	0.2381	7.711
5	13	2.1095	9.2825	1.5446	3.58E-04	13.6475	0.0132	0.1929	9.944
6	14	2.0977	0.3930	0.0536	4.27E-08	4.3512	0.0000	0.2414	12.458
7	14	2.0909	2.6427	0.4355	1.23E-05	3.4990	0.0005	0.2136	10.556
8	12	2.0441	7.3854	1.2585	4.16E-04	8.3131	0.0181	0.2246	9.714
9	18	2.1662	15.0446	0.9573	2.26E-04	9.6000	0.0092	0.1533	14.38
10	12	2.1643	9.9299	1.7928	5.64E-05	2.3927	0.0022	0.2132	10.285
11	16	2.1390	18.7592	3.5468	3.12E-04	6.8135	0.0133	0.1526	12.228
12	12	2.1748	12.5422	2.1581	8.87E-05	4.6782	0.0041	0.2004	9.143
13	16	2.1178	1.8270	0.2925	1.09E-05	8.7554	0.0005	0.1860	11.847
14	13	2.2324	12.1109	2.3049	2.01E-05	8.1229	0.0010	0.1897	9.604
15	12	2.0847	24.8676	4.5391	3.58E-04	10.8051	0.0148	0.1927	9.043
16	11	2.1258	9.5069	1.6576	4.60E-04	3.8885	0.0192	0.2259	9.254
17	11	2.0826	17.4011	2.9032	1.49E-04	3.5633	0.0070	0.2107	9.423
18	11	2.1198	9.8155	1.6207	4.83E-05	4.7311	0.0017	0.2239	8.362
19	13	2.1210	9.8481	1.8632	1.57E-04	2.6221	0.0073	0.1994	10.925
20	16	2.0988	4.7295	0.7542	6.95E-05	3.3306	0.0024	0.1820	12.427
21	11	2.0774	3.8758	0.6963	7.94E-06	3.2941	0.0004	0.2449	8.072
22	14	2.0915	7.7041	1.1947	1.11E-05	9.1991	0.0004	0.1896	9.974
23	13	2.0389	1.7195	0.2749	4.99E-05	9.4614	0.0025	0.2301	9.744
24	12	2.0766	5.6240	0.8750	9.02E-06	5.7116	0.0003	0.2226	8.963
25	13	2.1307	8.3897	1.5954	4.16E-06	3.3982	0.0001	0.2005	9.484
26	16	2.0210	2.0471	0.3346	4.13E-05	7.4936	0.0020	0.1832	11.647
27	12	2.1408	13.6107	2.5404	2.69E-04	11.4540	0.0118	0.2098	8.482
28	13	2.1315	18.7943	3.4648	1.34E-04	15.0465	0.0066	0.1858	9.534
29	11	2.1366	20.2316	3.3111	7.23E-04	3.8685	0.0242	0.2130	8.021
30	9	2.1213	21.5777	3.1012	7.63E-04	2.3423	0.0278	0.2564	6.51
31	11	2.0803	14.5040	2.3798	4.65E-04	3.6760	0.0201	0.2183	7.951
32	11	2.0801	6.9472	0.9861	2.26E-04	6.6571	0.0092	0.2424	7.951
33	12	2.0553	6.7996	1.2589	1.98E-04	1.0488	0.0089	0.2262	8.702
34	13	2.0789	3.1128	0.4563	3.34E-04	22.7357	0.0138	0.2208	9.864
35	11	2.2068	10.0467	1.5151	1.32E-05	3.4071	0.0005	0.2197	8.513
36	13	2.1763	10.8406	1.7239	4.93E-04	4.7161	0.0206	0.1986	9.303
37	13	2.1345	18.1442	3.7287	8.33E-05	7.3881	0.0034	0.1909	9.794
38	9	2.1102	24.0767	4.0577	7.62E-04	6.4437	0.0295	0.2564	6.57
39	10	2.1599	16.2520	2.3780	5.08E-04	6.0491	0.0213	0.2485	7.05
40	11	2.1942	22.5610	4.4127	8.04E-04	7.7243	0.0324	0.2154	7.981
priemer	**12.675**	**2.1169**	**11.7330**	**1.9847**	**2.37E-04**	**6.6131**	**0.0096**	**0.2077**	**9.5758**

metóda: primárno - duálna

vstup: počet riešených úloh = **40**

rozmer úlohy $(n\,x\,m) = (50\,x\,100)$

vzdialenosť štartovacieho bodu x^0 od optimálneho riešenia $\hat{x}$ je $\rho = 100$

kritérium presnosti: $\|x_i(\mu_k) - \hat{x}\| < 0.001$

počet vnútorných Newtonovských iterácií $= 4$

počet iterácií metódy zlatého rezu $= 25$

veľkosť redukčného parametra $\sigma = 3$

výstup:

Úloha číslo	počet vonkajších iterácií	počiatočné mí	||rc||	||rmi||	||x-xopt||	||y-yopt||	||z-yopt||	priemerné lambda	CPU time
1	11	2.1198	24.5888	4.4622	4.08E-04	6.2558	0.0166	0.1633	10.024
2	10	2.0976	8.7868	1.6885	2.20E-05	7.3463	0.0012	0.1832	9.744
3	11	2.2048	22.5680	3.6900	3.41E-04	9.1879	0.0144	0.1625	10.334
4	14	2.2013	2.8414	0.4813	8.81E-06	4.0115	0.0004	0.1572	14.641
5	8	2.0439	12.4900	1.9863	2.30E-05	1.8830	0.0011	0.2383	7.210
6	10	2.1095	8.6392	1.4544	1.65E-04	12.8041	0.0061	0.1847	9.604
7	11	2.0977	0.3467	0.0515	3.34E-07	4.3343	0.0000	0.2321	10.074
8	10	2.0909	1.9515	0.3205	1.45E-05	3.5308	0.0006	0.2195	9.604
9	9	2.0441	6.8374	1.1967	4.84E-04	8.3038	0.0215	0.2322	7.922
10	12	2.1662	9.9457	1.9023	8.52E-04	9.9658	0.0351	0.1597	11.506
11	9	2.1643	10.8124	1.9854	1.77E-04	2.5569	0.0068	0.2110	8.643
12	11	2.1390	18.0637	3.4700	2.83E-04	6.8105	0.0121	0.1716	9.844
13	12	2.0423	0.6994	0.1113	4.00E-07	5.0557	0.0000	0.2114	10.805
14	8	2.1748	10.9389	1.9835	4.59E-05	4.3549	0.0022	0.2207	7.671
15	11	2.1178	1.7932	0.2925	1.14E-05	8.7232	0.0005	0.1983	10.565
16	9	2.2324	11.8374	2.2795	1.77E-05	8.0989	0.0009	0.2003	8.633
17	9	2.0847	23.9065	4.3324	3.32E-04	10.8647	0.0139	0.1927	8.372
18	9	2.1258	7.3409	1.2648	2.76E-04	3.3348	0.0115	0.2303	7.891
19	8	2.0826	16.8449	2.8596	1.34E-04	3.6646	0.0064	0.2381	6.960
20	8	2.1198	10.1122	1.7056	3.57E-05	4.8452	0.0012	0.2367	7.431
21	11	2.1176	11.0439	2.0040	5.47E-05	3.8951	0.0025	0.1759	10.315
22	10	2.1210	8.1005	1.5188	1.34E-04	2.7081	0.0062	0.2001	9.083
23	11	2.0988	3.9315	0.6336	6.13E-05	3.3702	0.0021	0.1896	10.575
24	12	2.0530	0.5043	0.0827	2.89E-07	4.7545	0.0000	0.2055	11.036
25	9	2.0774	3.3861	0.6335	9.91E-06	3.4793	0.0005	0.2476	7.902
26	10	2.0915	7.4302	1.1822	8.73E-06	9.2101	0.0003	0.2045	8.872
27	9	2.0389	1.6669	0.2732	4.38E-05	9.0116	0.0022	0.2442	8.382
28	14	2.1542	0.8185	0.1602	1.61E-06	5.3598	0.0001	0.1772	12.958
29	9	2.0766	5.4660	0.8594	2.60E-06	5.7794	0.0001	0.2283	8.362
30	10	2.1307	8.4079	1.6163	3.77E-06	3.4482	0.0001	0.2049	9.124
31	11	2.0210	2.2579	0.3766	4.21E-05	8.0528	0.0020	0.2008	10.064
32	10	2.1408	8.3787	1.5812	1.36E-04	8.7096	0.0059	0.2053	8.883
33	12	2.0638	4.1368	0.6933	2.68E-05	5.8927	0.0011	0.1778	11.046
34	9	2.1315	16.5468	2.9174	9.36E-05	14.6243	0.0045	0.1972	8.392
35	9	2.1366	15.4404	2.5303	6.28E-04	3.4804	0.0211	0.2155	7.891
36	7	2.1213	23.1954	3.4497	9.33E-04	2.1352	0.0340	0.2746	5.999
37	9	2.0803	13.1482	2.0394	3.88E-04	3.6336	0.0168	0.2156	7.911
38	8	2.0801	6.4095	0.9229	2.05E-04	6.7212	0.0083	0.2507	7.411
39	8	2.0553	9.2085	1.7568	7.45E-04	1.5739	0.0334	0.2365	7.681
40	9	2.0789	3.4487	0.5152	6.92E-04	24.3100	0.0286	0.2355	8.172
priemer	**9.925**	**2.1082**	**9.1068**	**1.5816**	**1.96E-04**	**6.4021**	**0.0081**	**0.2083**	**9.1884**

metóda: primárno - duálna

vstup: počet riešených úloh = **40**

rozmer úlohy $(n\,x\,m) = (50\,x\,100)$

vzdialenosť štartovacieho bodu x^0 od optimálneho riešenia $\hat{x}$ je $\rho = 100$

kritérium presnosti: $\|x_i(\mu_k) - \hat{x}\| < 0.001$

počet vnútorných Newtonovských iterácií $= 4$

počet iterácií metódy zlatého rezu $= 30$

veľkosť redukčného parametra $\sigma = 3$

výstup:

Úloha číslo	počet vonkajších iterácií	počiatočné mí	||rc||	||rmi||	||x-xopt||	||y-yopt||	||z-yopt||	priemerné lambda	CPU time
1	10	2.1198	24.4530	4.4382	4.03E-04	6.2942	0.0164	0.1674	10.165
2	10	2.0976	10.8912	2.1177	2.26E-05	7.5694	0.0012	0.1968	8.933
3	11	2.2048	24.8598	4.0135	8.48E-04	12.6874	0.0349	0.1631	9.994
4	13	2.2013	2.8699	0.4817	5.15E-06	4.0510	0.0002	0.1692	12.087
5	8	2.0439	9.0163	1.4006	1.09E-05	1.4318	0.0005	0.2461	7.180
6	10	2.1095	8.7662	1.4746	3.23E-04	14.1403	0.0120	0.1969	9.864
7	11	2.0977	0.3433	0.0511	3.90E-08	4.3339	0.0000	0.2306	10.816
8	10	2.0909	1.9477	0.3198	1.38E-05	3.5307	0.0006	0.2193	10.415
9	8	2.0441	9.1819	1.6098	4.51E-04	8.3074	0.0201	0.2471	7.781
10	11	2.1662	9.8099	1.8727	8.51E-04	9.9911	0.0352	0.1739	11.456
11	9	2.1643	11.2857	2.0644	1.04E-04	2.7243	0.0040	0.2102	9.254
12	10	2.1390	21.6847	4.1759	8.01E-04	9.1957	0.0344	0.1832	9.133
13	12	2.0423	0.6831	0.1085	8.27E-08	5.1342	0.0000	0.2069	11.246
14	8	2.1748	11.4983	2.0851	6.21E-05	4.5163	0.0029	0.2193	8.242
15	11	2.1178	1.7474	0.2861	9.07E-06	8.5094	0.0004	0.2131	12.197
16	9	2.2324	11.8683	2.2881	1.84E-05	8.1086	0.0010	0.2059	9.944
17	9	2.0847	24.3055	4.4126	3.43E-04	11.0880	0.0144	0.1977	8.903
18	8	2.1258	9.3711	1.6231	4.66E-04	4.1703	0.0194	0.2316	8.161
19	8	2.0826	16.9058	2.8670	1.38E-04	3.6971	0.0065	0.2373	7.280
20	8	2.1198	10.1148	1.7062	3.58E-05	4.8467	0.0012	0.2366	7.761
21	10	2.1176	10.6675	1.9132	5.77E-05	3.8625	0.0026	0.1939	9.894
22	10	2.1210	8.0959	1.5244	1.44E-04	2.6867	0.0067	0.2007	9.483
23	10	2.0988	3.8663	0.6205	5.74E-05	3.4887	0.0020	0.2187	10.215
24	11	2.0530	0.5036	0.0825	2.94E-07	4.7548	0.0000	0.2149	11.967
25	8	2.0774	3.8381	0.7154	1.54E-05	3.6620	0.0008	0.2529	8.302
26	9	2.0915	7.4810	1.1853	9.63E-06	9.3539	0.0004	0.2096	9.394
27	9	2.0389	1.6246	0.2681	1.22E-04	9.4398	0.0062	0.2570	10.084
28	13	2.1542	0.8076	0.1594	1.23E-07	5.3330	0.0000	0.1937	13.259
29	9	2.0766	5.4689	0.8586	2.33E-06	5.7881	0.0001	0.2280	9.474
30	10	2.1307	8.2805	1.5976	4.25E-06	3.4233	0.0001	0.2056	10.906
31	10	2.0210	2.2937	0.3838	6.51E-05	7.9611	0.0031	0.2110	12.508
32	9	2.1408	11.0646	2.1121	4.97E-04	10.8941	0.0219	0.2171	10.294
33	12	2.0638	3.6467	0.6102	2.19E-05	5.8018	0.0009	0.1834	11.105
34	9	2.1315	23.2337	4.1731	7.84E-05	17.1309	0.0038	0.2062	8.021
35	9	2.1366	15.3454	2.5151	5.19E-04	3.1966	0.0174	0.2173	8.042
36	7	2.1213	23.2006	3.4503	9.33E-04	2.1338	0.0340	0.2746	6.089
37	8	2.0803	12.8401	1.9893	5.89E-04	3.5779	0.0256	0.2228	7.831
38	8	2.0801	7.1592	1.0346	1.99E-04	7.6277	0.0081	0.2552	7.310
39	8	2.0553	7.3152	1.3844	2.27E-04	1.2245	0.0102	0.2430	7.812
40	9	2.0789	3.4106	0.5118	7.76E-04	24.3230	0.0321	0.2363	9.073
priemer	**9.55**	**2.1082**	**9.5437**	**1.6622**	**2.31E-04**	**6.7498**	**0.0095**	**0.2149**	**9.5469**

metóda: primárno - duálna

vstup: počet riešených úloh = **40**

rozmer úlohy $(n\,x\,m) = (50\,x\,100)$

vzdialenosť štartovacieho bodu x^0 od optimálneho riešenia $\hat{x}$ je $\rho = 100$

kritérium presnosti: $\|x_i(\mu_k) - \hat{x}\| < 0.001$

počet vnútorných Newtonovských iterácií $= 5$

počet iterácií metódy zlatého rezu $= 25$

veľkosť redukčného parametra $\sigma = 3$

výstup:

Úloha číslo	počet vonkajších iterácií	počiatočné mí	||rc||	||rmi||	||x-xopt||	||y-yopt||	||z-yopt||	priemerné lambda	CPU time
1	8	2.1198	23.9048	4.4080	4.35E-04	6.1902	0.0179	0.1721	9.634
2	8	2.0976	8.5582	1.6993	3.30E-05	7.3732	0.0017	0.2042	8.653
3	8	2.2048	30.4009	5.0831	9.06E-04	13.6662	0.0373	0.1696	9.383
4	10	2.2013	2.7505	0.4698	4.66E-07	4.0518	0.0000	0.1762	11.557
5	6	2.0439	11.6830	1.8904	3.08E-05	1.8943	0.0014	0.2553	6.740
6	8	2.1095	8.7067	1.4988	4.07E-04	14.2363	0.0152	0.1973	8.892
7	8	2.0977	0.2457	0.0483	7.77E-06	4.2712	0.0004	0.2571	8.933
8	8	2.0909	1.7631	0.3091	3.42E-05	3.5234	0.0015	0.2332	9.163
9	7	2.0441	7.1835	1.3177	6.77E-04	8.2557	0.0304	0.2330	7.931
10	9	2.1662	9.5470	1.8733	9.02E-04	9.9423	0.0372	0.1831	10.115
11	7	2.1643	10.6047	2.0047	9.56E-05	2.5799	0.0036	0.2236	8.171
12	12	2.1531	3.9324	0.7352	2.95E-05	8.2703	0.0015	0.1403	14.211
13	8	2.1390	17.2678	3.3371	4.39E-05	5.9342	0.0017	0.1882	9.143
14	10	2.0423	0.6672	0.1085	2.61E-06	5.0656	0.0001	0.2062	11.076
15	7	2.1748	10.4271	2.1086	6.10E-05	4.4623	0.0029	0.2233	7.691
16	8	2.1178	1.8643	0.3391	9.71E-04	8.6572	0.0435	0.2261	9.384
17	7	2.2324	11.4372	2.2849	2.17E-05	8.0747	0.0011	0.2131	8.161
18	7	2.0847	23.9793	4.4443	3.19E-04	10.8748	0.0133	0.1986	8.181
19	7	2.1258	8.0641	1.4474	3.52E-04	4.1951	0.0147	0.2336	7.691
20	6	2.0826	13.9125	2.4814	8.98E-05	3.4940	0.0043	0.2599	6.490
21	6	2.1198	10.2284	1.7905	3.16E-05	4.9005	0.0010	0.2440	7.210
22	8	2.1176	10.4329	1.9634	8.07E-05	3.8724	0.0037	0.2001	9.143
23	8	2.1210	8.6991	1.6747	1.21E-04	2.7564	0.0056	0.2102	8.673
24	8	2.0988	3.4162	0.5528	3.45E-05	3.4881	0.0012	0.2190	9.173
25	9	2.0530	0.4687	0.0829	2.24E-06	4.7224	0.0001	0.2312	9.854
26	7	2.0774	3.6605	0.7516	3.65E-05	4.2400	0.0020	0.2530	7.692
27	7	2.0915	6.9950	1.1757	2.13E-05	9.4347	0.0010	0.2305	7.941
28	7	2.0389	1.5053	0.2661	1.45E-04	8.9121	0.0073	0.2522	8.212
29	10	2.1542	0.7667	0.1563	1.64E-06	5.3492	0.0001	0.1984	11.566
30	7	2.0766	5.2756	0.8618	9.78E-06	5.8577	0.0004	0.2346	8.202
31	7	2.1307	8.1485	1.5946	6.36E-04	4.4954	0.0244	0.2175	8.402
32	8	2.0210	2.0814	0.3702	6.62E-05	7.8406	0.0032	0.2225	9.154
33	8	2.1408	9.3507	1.8220	2.32E-04	10.2125	0.0102	0.2070	8.672
34	9	2.0638	5.4994	0.9512	5.03E-05	6.2431	0.0021	0.1901	10.325
35	7	2.1315	21.5560	4.0403	8.41E-05	17.0690	0.0034	0.2076	7.931
36	6	2.1366	25.3051	4.3252	9.41E-04	7.2488	0.0318	0.2177	7.240
37	5	2.1213	20.4312	3.0922	7.09E-04	1.7896	0.0260	0.2795	6.029
38	7	2.0803	12.8103	2.0086	3.95E-04	3.5697	0.0171	0.2278	7.470
39	7	2.0801	6.0143	0.9248	9.44E-05	7.2326	0.0038	0.2449	7.702
40	6	2.0553	7.0325	1.4194	3.62E-04	1.4063	0.0162	0.2593	7.210
priemer	**7.65**	**2.1101**	**9.4144**	**1.6928**	**2.37E-04**	**6.3913**	**0.0098**	**0.2185**	**8.7250**

metóda: primárno - duálna

vstup: počet riešených úloh = **40**

rozmer úlohy $(n\,x\,m) = (50\,x\,100)$

vzdialenosť štartovacieho bodu x^0 od optimálneho riešenia $\hat{x}$ je $\rho = 100$

kritérium presnosti: $\|x_i(\mu_k) - \hat{x}\| < 0.001$

počet vnútorných Newtonovských iterácií $= 5$

počet iterácií metódy zlatého rezu $= 30$

veľkosť redukčného parametra $\sigma = 3$

výstup:

Úloha číslo	počet vonkajších iterácií	počiatočné mí	||rc||	||rmi||	||x-xopt||	||y-yopt||	||z-yopt||	priemerné lambda	CPU time
1	5	2.0480	15.4674	3.0135	2.50E-04	3.3388	0.0086	0.2720	6.249
2	9	2.0364	0.7800	0.1306	1.58E-06	6.8156	0.0001	0.2173	10.355
3	8	2.0595	10.9304	2.2118	9.19E-04	9.2599	0.0375	0.1993	9.123
4	8	2.1215	6.0119	1.0526	6.18E-04	10.7680	0.0226	0.2149	8.653
5	7	2.1442	14.0492	2.7683	8.92E-04	7.3267	0.0329	0.2252	7.691
6	8	2.1405	0.4395	0.0644	4.57E-06	18.5446	0.0002	0.2484	9.373
7	7	2.0922	4.1899	0.7361	9.67E-06	7.7523	0.0004	0.2383	8.162
8	7	2.0637	9.0827	1.5671	8.59E-04	1.2643	0.0343	0.2404	7.671
9	6	2.2222	18.7220	3.2921	7.42E-04	11.8260	0.0270	0.2331	7.200
10	7	2.0990	9.2644	1.4453	1.44E-04	3.1349	0.0060	0.2210	8.161
11	7	2.0935	7.3010	1.2409	3.04E-04	5.4151	0.0104	0.2335	7.932
12	9	2.1914	3.1979	0.6059	1.09E-05	3.8467	0.0004	0.1953	10.084
13	6	2.0725	21.7339	4.3833	2.35E-04	6.3863	0.0102	0.2385	6.710
14	8	2.0324	1.5085	0.2695	4.49E-07	3.3404	0.0000	0.2240	9.363
15	7	2.0544	12.7274	2.3835	1.05E-04	2.8535	0.0035	0.2158	8.423
16	11	2.0099	1.0455	0.1847	4.80E-08	3.3383	0.0000	0.1777	12.988
17	7	2.0187	8.3719	1.5501	2.12E-05	3.5844	0.0009	0.2199	8.162
18	7	2.2416	12.2453	1.9241	1.29E-04	4.7564	0.0052	0.2323	7.711
19	7	2.1621	11.4197	1.9672	2.46E-04	10.1191	0.0097	0.2146	8.162
20	7	2.1222	10.0306	1.6979	1.22E-04	3.9932	0.0050	0.2324	7.921
21	8	2.1198	23.8353	4.3961	4.28E-04	6.2256	0.0177	0.1766	9.123
22	8	2.0976	8.5960	1.7028	3.19E-05	7.3819	0.0016	0.2036	8.662
23	8	2.2048	29.9260	4.9957	9.68E-04	13.2104	0.0398	0.1741	9.143
24	10	2.2013	2.8690	0.4902	6.00E-07	4.1005	0.0000	0.1832	11.036
25	6	2.0439	11.6871	1.8907	3.06E-05	1.8958	0.0014	0.2552	6.720
26	7	2.1095	10.2040	1.7896	6.74E-04	13.3492	0.0253	0.2091	8.392
27	8	2.0977	0.2418	0.0483	8.52E-06	4.2689	0.0004	0.2636	8.883
28	8	2.0909	1.7441	0.3056	3.30E-05	3.5229	0.0014	0.2333	9.133
29	7	2.0441	7.2088	1.3223	6.76E-04	8.2561	0.0303	0.2330	7.951
30	9	2.1662	7.0739	1.3607	5.57E-04	8.6229	0.0230	0.1774	10.576
31	7	2.1643	11.9881	2.2662	7.76E-05	2.7717	0.0029	0.2214	8.141
32	12	2.1531	3.9180	0.7286	2.79E-05	8.2723	0.0014	0.1475	13.479
33	7	2.1390	20.0391	3.8099	3.67E-04	9.1540	0.0160	0.1935	8.402
34	10	2.0423	0.6736	0.1095	2.20E-06	5.0606	0.0001	0.2061	11.026
35	7	2.1748	10.4288	2.1089	6.08E-05	4.4604	0.0029	0.2233	7.691
36	8	2.1178	1.5275	0.2777	1.43E-05	8.5903	0.0006	0.2207	9.634
37	7	2.2324	11.4455	2.2856	2.12E-05	8.0768	0.0011	0.2131	8.182
38	7	2.0847	26.4560	4.9119	3.82E-04	12.2626	0.0160	0.2012	7.931
39	7	2.1258	7.2131	1.3046	2.96E-04	3.5413	0.0123	0.2377	7.691
40	6	2.0826	14.0118	2.4992	9.31E-05	3.4980	0.0044	0.2599	6.490
priemer	**7.625**	**2.1130**	**9.7402**	**1.7773**	**2.59E-04**	**6.6047**	**0.0103**	**0.2182**	**8.7095**

metóda: primárno - duálna

vstup: počet riešených úloh = **40**

rozmer úlohy $(n\,x\,m) = (50\,x\,100)$

vzdialenosť štartovacieho bodu x^0 od optimálneho riešenia $\hat{x}$ je $\rho = 100$

kritérium presnosti: $\|x_i(\mu_k) - \hat{x}\| < 0.001$

počet vnútorných Newtonovských iterácií $= 4$

počet iterácií metódy zlatého rezu $= 25$

veľkosť redukčného parametra $\sigma = 5$

výstup:

Úloha číslo	počet vonkajších iterácií	počiatočné mí	||rc||	||rmi||	||x-xopt||	||y-yopt||	||z-yopt||	priemerné lambda	CPU time
1	12	2.1198	24.6629	4.4475	4.17E-04	6.2716	0.0167	0.1486	12.388
2	12	2.0976	13.3865	1.7569	2.96E-05	6.8664	0.0016	0.1587	12.358
3	12	2.2048	21.6808	3.5664	7.21E-04	10.6828	0.0302	0.1474	15.692
4	7.5	2.0439	10.8975	1.7876	2.37E-05	1.6292	0.0011	0.2442	8.702
5	11	2.1095	10.0614	1.6582	2.35E-04	12.6280	0.0087	0.1729	18.266
6	12	2.0909	2.5238	0.4238	1.56E-05	3.5001	0.0007	0.1958	12.177
7	13	2.1390	19.7939	3.6683	3.09E-04	6.9217	0.0133	0.1373	12.228
8	9	2.1748	12.6619	2.1922	1.78E-04	4.7469	0.0082	0.1999	8.371
9	14	2.1178	1.6434	0.2463	7.43E-06	8.7942	0.0003	0.1653	12.959
10	11	2.2324	23.2970	0.9515	2.29E-06	8.9145	0.0001	0.1776	10.565
11	10	2.0847	23.3875	4.2533	3.12E-04	9.8383	0.0129	0.1794	9.143
12	9	2.1258	8.9157	1.5168	4.00E-04	3.8300	0.0167	0.2075	8.642
13	9	2.0826	18.1012	3.0222	1.76E-04	3.6690	0.0083	0.2096	7.922
14	8	2.1198	11.5756	1.9268	3.83E-05	5.0319	0.0013	0.2292	7.671
15	12	2.0988	4.1890	0.6791	7.40E-05	3.4000	0.0025	0.1734	11.507
16	10	2.0774	2.8646	0.5175	2.13E-06	2.2036	0.0001	0.2279	8.873
17	11	2.0915	7.7233	1.1960	1.12E-05	9.2202	0.0004	0.1847	9.814
18	10	2.0389	1.7116	0.2731	3.77E-05	9.4620	0.0019	0.2296	8.863
19	10	2.0766	5.4527	0.8585	1.01E-05	5.7099	0.0003	0.2013	9.594
20	11	2.1307	7.9265	1.5504	4.16E-06	3.2522	0.0001	0.1828	10.305
21	12	2.0210	2.1980	0.3452	9.10E-04	7.6645	0.0439	0.1783	11.517
22	11	2.1408	7.7828	1.4310	9.45E-05	7.4990	0.0041	0.1755	10.575
23	10	2.1315	20.4747	3.8801	1.78E-04	15.2935	0.0087	0.1716	9.584
24	9	2.1366	18.1342	2.9894	8.55E-04	3.3528	0.0287	0.2045	8.412
25	7	2.1213	13.9535	1.9852	2.47E-04	1.3580	0.0090	0.2628	6.709
26	10	2.0803	18.6562	2.9837	6.93E-04	3.6675	0.0302	0.1931	8.883
27	9	2.0801	6.8269	0.9781	2.11E-04	6.6316	0.0085	0.2293	8.162
28	10	2.0553	7.2733	1.3690	6.44E-04	1.2811	0.0288	0.2065	9.123
29	11	2.0789	3.2814	0.4859	8.33E-04	24.2067	0.0345	0.1999	9.844
30	9	2.2068	11.7411	1.7663	1.06E-05	3.3881	0.0004	0.2079	8.402
31	10	2.1763	10.7752	1.7171	6.53E-04	5.4278	0.0263	0.1856	9.354
32	11	2.1345	19.1577	3.9675	1.84E-04	7.4704	0.0083	0.1718	9.854
33	7	2.1102	23.5201	3.9910	8.83E-04	6.2588	0.0358	0.2480	6.239
34	8	2.1599	15.5004	2.2929	5.11E-04	5.6802	0.0208	0.2353	7.201
35	9	2.1942	16.1538	3.0816	6.79E-04	6.5920	0.0276	0.1965	8.642
36	13	2.1248	9.2249	1.6614	3.05E-04	4.4735	0.0123	0.1554	11.737
37	12	2.0974	12.7921	2.3832	1.58E-04	0.8990	0.0066	0.1618	11.276
38	7	2.1514	9.3698	1.3024	1.75E-04	1.1977	0.0078	0.2739	6.730
39	13	2.2200	16.8281	3.0693	8.42E-05	4.5499	0.0037	0.1446	12.228
40	13	2.0378	10.1522	1.9423	5.93E-05	1.3863	0.0025	0.1553	12.028
priemer	**10.3625**	**2.1179**	**12.1563**	**2.0029**	**2.84E-04**	**6.1213**	**0.0118**	**0.1933**	**10.0635**

metóda: primárno - duálna

vstup: počet riešených úloh = **40**

rozmer úlohy $(n\,x\,m) = (50\,x\,100)$

vzdialenosť štartovacieho bodu x^0 od optimálneho riešenia $\hat{x}$ je $\rho = 100$

kritérium presnosti: $\|x_i(\mu_k) - \hat{x}\| < 0.001$

počet vnútorných Newtonovských iterácií $= 4$

počet iterácií metódy zlatého rezu $= 30$

veľkosť redukčného parametra $\sigma = 5$

výstup:

Úloha číslo	počet vonkajších iterácií	počiatočné mí	||rc||	||rmi||	||x-xopt||	||y-yopt||	||z-yopt||	priemerné lambda	CPU time
1	7	2.0480	16.9160	3.2074	6.16E-04	3.1810	0.0216	0.2509	7.080
2	11	2.0595	11.4148	2.2267	9.50E-04	9.4107	0.0384	0.1845	9.844
3	11	2.1215	4.8548	0.7154	3.69E-04	9.6453	0.0135	0.1827	10.615
4	11	2.1442	12.8545	2.4407	9.63E-04	4.4530	0.0355	0.1744	10.325
5	11	2.0922	3.9790	0.6769	1.06E-05	7.7931	0.0005	0.1992	9.864
6	10	2.0637	4.8297	0.7786	2.25E-04	0.6339	0.0090	0.2219	8.893
7	8	2.2222	17.6397	3.0033	8.62E-04	7.5046	0.0374	0.2245	7.721
8	9	2.0990	8.1434	1.2282	6.00E-05	2.5642	0.0026	0.2257	8.172
9	9	2.0935	7.7317	1.2239	2.39E-04	5.5999	0.0081	0.2138	8.652
10	9	2.0725	16.8962	3.0627	4.99E-05	3.6648	0.0023	0.1945	8.653
11	12	2.0324	1.5684	0.2717	8.52E-07	3.3473	0.0000	0.1835	11.556
12	10	2.0544	13.9969	2.5800	5.97E-04	3.5114	0.0202	0.1941	9.094
13	10	2.0187	11.2222	2.0256	3.66E-05	3.6917	0.0016	0.1850	9.603
14	9.5	2.2416	11.5151	1.7126	2.58E-04	4.4150	0.0105	0.1965	9.134
15	11	2.1621	15.8536	2.7376	8.02E-04	11.7367	0.0340	0.1733	10.094
16	8.5	2.1222	11.0570	1.7832	5.49E-04	3.9232	0.0221	0.2208	8.172
17	12	2.1198	19.5296	4.2330	4.22E-04	6.3620	0.0171	0.1492	10.836
18	11	2.0976	9.6410	1.7675	1.92E-05	7.3791	0.0011	0.1745	10.094
19	11	2.2048	22.9013	4.2935	3.67E-04	11.2304	0.0163	0.1580	10.335
20	8	2.0439	10.6735	1.7524	2.58E-05	1.5659	0.0012	0.2449	7.200
21	11	2.1095	8.6401	1.4526	1.34E-04	11.7387	0.0049	0.1827	9.884
22	12	2.0909	3.1459	0.3440	1.31E-05	3.5378	0.0006	0.1835	11.507
23	9	2.0441	9.2258	1.5922	4.03E-04	8.3074	0.0176	0.2251	8.172
24	10	2.1643	10.2198	1.8513	1.02E-04	2.4401	0.0040	0.2069	8.913
25	12	2.1390	18.6726	3.5443	2.97E-04	6.8175	0.0127	0.1527	11.056
26	9	2.1748	12.6364	2.1817	1.90E-04	4.7735	0.0085	0.1997	8.412
27	13	2.1178	2.0952	0.3355	1.30E-05	8.7504	0.0006	0.1770	11.777
28	10	2.2324	9.5435	2.3092	2.05E-05	8.1339	0.0011	0.1803	9.644
29	9	2.0847	24.9540	4.5517	3.56E-04	10.8747	0.0147	0.1925	8.392
30	9	2.1258	9.2677	1.6057	4.30E-04	4.0173	0.0179	0.2123	8.412
31	9	2.0826	18.0065	3.0037	1.66E-04	3.7601	0.0077	0.2092	7.932
32	8	2.1198	11.3500	1.8868	4.79E-05	5.0100	0.0017	0.2289	7.691
33	10	2.1210	29.7333	1.1253	7.84E-05	3.7883	0.0036	0.1957	9.633
34	13	2.0988	3.7864	0.1966	9.35E-07	3.2097	0.0000	0.1780	12.498
35	9	2.0774	3.2642	0.5917	7.52E-06	2.4968	0.0004	0.2380	8.412
36	10	2.0915	7.6549	1.1942	1.46E-05	9.1802	0.0005	0.1899	9.614
37	9	2.0389	2.4342	0.3893	7.32E-05	9.4287	0.0037	0.2410	8.412
38	10	2.0766	5.5732	0.8692	8.85E-06	5.7173	0.0003	0.2167	8.903
39	10	2.1307	8.8813	1.8235	9.70E-04	5.2688	0.0373	0.1981	9.133
40	11	2.0210	2.2929	0.3486	4.37E-05	7.4554	0.0021	0.1933	10.535
priemer	**10.05**	**2.1064**	**10.8649**	**1.8229**	**2.70E-04**	**5.9080**	**0.0108**	**0.1988**	**9.3717**

B. Zdrojový kód programu.

Jednotlivé metódy a funkcie boli naprogramované v jazyku C++ v prostredí Builder 5.

```
//---------------------------------------------------------------------------

#include <vcl.h>
#pragma hdrstop
#include <math.h>
#include <time.h>
#include <stdlib.h>
#include <Math.hpp>
#include <fstream.h>
#include "Diplomovka.h"
//---------------------------------------------------------------------------
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
//---------------------------------------------------------------------------
__fastcall TForm1::TForm1(TComponent* Owner)
        : TForm(Owner)
{
}
//---------------------------------------------------------------------------
```

//-----definicia premennych-----

```
long double lambopt;
long double fxopt;
long double fx;
long double lambpom;
DynamicArray < long double > xopt;
DynamicArray < long double > x;
DynamicArray < long double > b;
DynamicArray < long double > c;
DynamicArray < long double > g;
DynamicArray < long double > s;
DynamicArray < long double > yypsilon;
DynamicArray < long double > y;
DynamicArray < long double > yopt;
DynamicArray < long double > z;
DynamicArray < long double > zopt;
DynamicArray < long double > rb;
DynamicArray < long double > rc;
DynamicArray < long double > rmi;
DynamicArray < long double > r1;
DynamicArray < long double > r;
DynamicArray < long double > deltax;
DynamicArray < long double > deltay;
DynamicArray < long double > deltaz;
DynamicArray < DynamicArray < long double > > A;
DynamicArray < DynamicArray < long double > > H;
DynamicArray < DynamicArray < long double > > D1;
DynamicArray < DynamicArray < long double > > D2;
DynamicArray < DynamicArray < long double > > D2inv;
```

```
DynamicArray < DynamicArray < long double > > POM;
DynamicArray < DynamicArray < long double > > POM2;
DynamicArray < DynamicArray < long double > > ADDA;

//-----absolutna hodnota-----
long double ABS(long double a) {
 if(a<0) return -a;
 else return a;
}

//-----inverzna matica pre maticu D2
void InverznaMaticaD2(DynamicArray < DynamicArray < long double > > A,DynamicArray < DynamicArray
< long double > > I) {
  int m = A[0].Length;
  if (A.Length!=A[0].Length) ShowMessage("matica nie je mxm");
  for (int i=0; i<m; i++) {
    for (int j=0;j<m;j++) {
      if (i==j) {
        if (A[i][j]==0) ShowMessage("zla D2");
        else I[i][j]=1/A[i][j];
      }
      else I[i][j]=0;
    }
  }
}

//Generator uloh LP s vopred zadanym optimalnym riesenim
void Generator(const n, const m, int ro,DynamicArray < DynamicArray < long double > > A,
   DynamicArray < DynamicArray < long double > > H,DynamicArray < long double > xopt,
   DynamicArray < long double > x,DynamicArray < long double > y,DynamicArray < long double > yopt,
   DynamicArray < long double > z,DynamicArray < long double > zopt,DynamicArray < long double > b,
   DynamicArray < long double > c,DynamicArray < long double > g,DynamicArray < long double > s, int
xnula) {
   if(n>m) {
     ShowMessage("rozmer n musi byt mensi ako rozmer m!!!!!");
     return;
   }
   DynamicArray < long double > ypsilon;
   ypsilon.set_length(n);
   DynamicArray < long double > yypsilon;
   yypsilon.set_length(m);

   for (int i=0;i<n;i++)                                    //tvorba A
     for (int j=0;j<m;j++) {
       A[i][j]=random(20)+1;
     }

   for (int i=0;i<n;i++) {                                  //vytvotim si optimalne x
     xopt[i]=random(20)+1;
   }

   int pocety=0;
   for (int j=0;j<m;j++) {                                  //tvorba z optimalne
     if ((m-j-1+pocety)< m/2) {
       zopt[j]=0;
       pocety++;
     }
     else {
       if (random(2)==0 && pocety<(m/2))  {
         zopt[j]=0;
```

```
      pocety++;
     }
     else {
      zopt[j]=random(20)+1;
     }
    }
   }

   for (int j=0;j<m;j++) {                                  //tvorba b
    b[j]=-zopt[j];
    for (int i=0;i<n;i++) b[j]=b[j] + A[i][j]*xopt[i];
   }


   for (int j=0;j<m;j++) {                                  //vypocet y optimalneho
    if (zopt[j]==0) {
     yopt[j]=random(20)+1;
    }
    else yopt[j]=0;
   }

   for (int i=0;i<n;i++) {                                  //tvorba c
    c[i]=0;
    for (int j=0;j<m;j++) {
     c[i]=c[i] + A[i][j]*yopt[j];
    }
   }
   srand(xnula);   //bacha!!!!!!!!!!!!!!!!!!!1
   long double v;
   long double v2=0;
   for (int i=0;i<n;i++) {
    ypsilon[i]=random(2000)+1;                               //vytvorim pomocny vektor ypsilon
    v2=v2+(ypsilon[i] - xopt[i])*(ypsilon[i] - xopt[i]);
   }

   v=ro/(sqrtl(v2));                                         //pomocna v na tvorbu x nula

   for(int i=0;i<n;i++) {                                    //tvorba x nula
    x[i]=xopt[i] + v*(ypsilon[i] - xopt[i]);
   }
   v2=0;
   for(int j=0;j<m;j++) {
    yypsilon[j]=random(2000)+1;                              //vytvorim pomocny vektor yypsilon
    v2=v2+(yypsilon[j] - yopt[j])*(yypsilon[j] - yopt[j]);
   }
   v=ro/(sqrtl(v2));                                         //pomocna v na tvorbu y nula

   for (int j=0;j<m;j++) {
    y[j]=yopt[j]+v*(yypsilon[j]-yopt[j]);                    //tvorba y nula
   }
   for(int j=0;j<m;j++) {                                    // tvorba z nula
    z[j]=-b[j];
    for(int i=0;i<n;i++) z[j]=z[j] + A[i][j]*x[i];
   }
}

//-----funkcna hodnota pri primarnej metode-----
long double Txmi(DynamicArray < DynamicArray < long double > > A,DynamicArray < long double > b,
            DynamicArray < long double > c,DynamicArray < long double > x,long double mi)
{
```

```
 int pozor=0;
 long double Txmi=0;
 long double aix;
 int n = c.Length;
 int m = A[0].Length;
 for(int j=0;j<n;j++) Txmi=Txmi+c[j]*x[j];
 for(int i=0;i<m;i++) {
  aix=-b[i];
  for(int j=0;j<n;j++) {
   aix=aix+A[j][i]*x[j];
  }
  if(aix<0) {
   ShowMessage("chyba pri aix");
  }
  aix=sqrtl(aix);
  Txmi=Txmi - aix*mi;
 }
return Txmi;
}

//-----funkcna hodnota prim-dual-----
long double Txmi2(DynamicArray < long double > z,DynamicArray < long double > c,DynamicArray < long
double > x, long double mi)
{
 long double Txmi=0;
 long double aix;
 int n = c.Length;
 int m = z.Length;
 for(int j=0;j<n;j++) Txmi=Txmi+c[j]*x[j];
 for(int i=0;i<m;i++) {
  if(z[i]<0) {
   ShowMessage("chyba pri aix");
  }
  aix=sqrtl(z[i]);
  Txmi=Txmi - aix*mi;
 }
 return Txmi;
}

//-----gradient-----
void gTxmi(DynamicArray < DynamicArray < long double > > A,DynamicArray < long double > b,
          DynamicArray < long double > c,DynamicArray < long double > x,
          DynamicArray < long double > g,long double mi)
{
 long double aix;
 int n = c.Length;
 int m = A[0].Length;
 for(int i=0;i<n;i++) {
  g[i]=c[i];
 }
 for(int j=0;j<m;j++) {
  aix=-b[j];
  for(int i=0;i<n;i++) aix=aix + A[i][j]*x[i];
  if(aix<=0) {
   ShowMessage("chyba v aix pri vypocte gradientu");
   return;
  }
  aix=2*sqrt(aix);
  for(int i=0;i<n;i++) {
   g[i]=g[i] - mi*A[i][j]/aix;
```

```
    }
   }
}

//-----hessova matica-----
void hTxmi(DynamicArray < DynamicArray < long double > > A,DynamicArray < long double > b,
          DynamicArray < long double > x,DynamicArray < DynamicArray < long double > > H,
          long double mi) {
  long double alfa,beta;
  DynamicArray < long double > V;
  int n = x.Length;
  int m = A[0].Length;
  V.set_length(n);
  for (int i=0;i<n;i++)
    for(int j=0;j<n;j++)
      H[i][j]=0;
  for(int i=0;i<m;i++) {
    alfa=-b[i];
    for(int j=0;j<n;j++) alfa=alfa + A[j][i]*x[j];
    if(alfa<=0) {
      ShowMessage("chyba v alfe pri vypocte hessianu");
      return;
    }
    long double q=powl(alfa,3);
    q=sqrtl(q);
    beta=(0.25*mi)/q;
    for(int j=0;j<n;j++) {
      V[j]=beta*A[j][i];
      for(int k=0;k<n;k++) H[k][j]=H[k][j] +V[j]*A[k][i];
    }
  }
}

//-----Q-R rozklad metodou Hansona-Lawsona-----
void HansLaws(DynamicArray <long double> s,DynamicArray < long double > g,
             DynamicArray < DynamicArray < long double > > H)
{
  int n = g.Length;
  int n1=n+1;
  DynamicArray < DynamicArray < long double > > a;
  a.set_length(n1);
  for (int j=0;j<n1;j++) a[j].set_length(n);
  long double u,sinv,sum,alfa,alfinv,xx,aa;

  for(int j=0;j<n;j++) {
    for(int i=0;i<n;i++) {
      a[i][j]=H[i][j];
    }
    a[n][j]=-g[j];
  }

  for(int j=0;j<n;j++) {
    u=0;
    for(int i=j;i<n;i++) {
      aa=ABS(a[j][i]);
      if(aa>u) u=aa;
    }
    if(u==0) {
      ShowMessage(IntToStr(j) + ". stlpec je nulovy");
      return;
```

```
    }
    sinv=1/u;
    sum=0;
    for(int i=j;i<n;i++) sum=sum + (a[j][i]*sinv)*(a[j][i]*sinv);
    u=u*sqrtl(sum);
    if(a[j][j]>0) u=-u;
    a[j][j]=a[j][j]-u;
    alfa=u*a[j][j];
    if(alfa!=0) {
      alfinv=1/alfa;
      for(int k=j+1;k<n+1;k++) {
        sum=0;
        for(int i=j;i<n;i++) sum=sum + a[j][i]*a[k][i];
        sum=sum*alfinv;
        for(int i=j;i<n;i++) a[k][i]=a[k][i] + sum*a[j][i];
      }
      a[j][j]=u;
    }
  }
  s[n-1]=a[n][n-1]/a[n-1][n-1];
  for(int i=n-2;i>=0;i--) {
    xx=a[n][i];
    for(int j=i+1;j<n;j++) {
      xx=xx-a[j][i]*s[j];
    }
    s[i]=xx/a[i][i];
  }
}
```

//-----zlaty rez pri primarnej metode-----

```
long double ZlatyRez(DynamicArray < DynamicArray < long double > > A,DynamicArray < long double > b,
            DynamicArray < long double > c,DynamicArray < long double > x,
            DynamicArray < long double > s,long double mi,long double aa,
            long double bb, int zlrez)
{
  int pozor=0;
  int n=x.Length;
  int k=0;
  long double tau=(sqrt(5)+1)/2;
  long double z1,z2,c1,c2,f1,f2;
  z1=2-tau;
  z2=tau-1;
  c1=aa + z1*(bb-aa);
  DynamicArray < long double > xlamb;
  xlamb.set_length(n);
  for (int i= 0;i<n;i++) xlamb[i]=x[i]+c1*s[i];
  f1=Txmi(A,b,c,xlamb,mi);
  c2=aa + z2*(bb-aa);
  for (int i= 0;i<n;i++) xlamb[i]=x[i]+c2*s[i];
  f2=Txmi(A,b,c,xlamb,mi);
  while( k!=zlrez) {
    if(f1<f2) {
      bb=c2;
      c2=c1;
      f2=f1;
      c1=aa + z1*(bb-aa);
      for (int i= 0;i<n;i++) xlamb[i]=x[i]+c1*s[i];
      f1=Txmi(A,b,c,xlamb,mi);
    }
    else {
```

```
    aa=c1;
    c1=c2;
    f1=f2;
    c2=aa + z2*(bb-aa);
    for (int i= 0;i<n;i++) xlamb[i]=x[i]+c2*s[i];
    f2=Txmi(A,b,c,xlamb,mi);
   }
   k++;
  }
  return c1;
}

//-----zlaty rez prim-dual-----
long double ZlatyRez2(DynamicArray < long double > x,DynamicArray < long double > deltax,DynamicArray
< long double > z,
                DynamicArray < long double > deltaz,DynamicArray < long double > c,long double mi,long
double aa,
                long double bb, int zlrez)
{
  int n=x.Length;
  int m=z.Length;
  int k=0;
  long double tau=(sqrt(5)+1)/2;
  long double z1,z2,c1,c2,f1,f2;
  z1=2-tau;
  z2=tau-1;
  c1=aa + z1*(bb-aa);
  DynamicArray < long double > xlamb;
  DynamicArray < long double > zlamb;
  xlamb.set_length(n);
  zlamb.set_length(m);
  for (int i= 0;i<n;i++) xlamb[i]=x[i]+c1*deltax[i];
  for (int j= 0;j<m;j++) zlamb[j]=z[j]+c1*deltaz[j];
  f1=Txmi2(zlamb,c,xlamb,mi);
  c2=aa + z2*(bb-aa);
  for (int i= 0;i<n;i++) xlamb[i]=x[i]+c2*deltax[i];
  for (int j= 0;j<m;j++) zlamb[j]=z[j]+c2*deltaz[j];
  f2=Txmi2(zlamb,c,xlamb,mi);
  while( k!=zlrez) {
   if(f1<f2) {
    bb=c2;
    c2=c1;
    f2=f1;
    c1=aa + z1*(bb-aa);
    for (int i= 0;i<n;i++) xlamb[i]=x[i]+c1*deltax[i];
    for (int j= 0;j<m;j++) zlamb[j]=z[j]+c1*deltaz[j];
    f1=Txmi2(zlamb,c,xlamb,mi);
   }
   else {
    aa=c1;
    c1=c2;
    f1=f2;
    c2=aa + z2*(bb-aa);
    for (int i= 0;i<n;i++) xlamb[i]=x[i]+c2*deltax[i];
    for (int j= 0;j<m;j++) zlamb[j]=z[j]+c1*deltaz[j];
    f2=Txmi2(zlamb,c,xlamb,mi);
   }
   k++;
  }
  return c1;
```

```
}

//-----faza I pri primarnej metode-----
long double FazaI(DynamicArray < DynamicArray < long double > > A,DynamicArray < long double > b,
        DynamicArray < long double > c,DynamicArray < long double > x,
        DynamicArray < long double > s,long double mi,int zlrez)
{
 int n=x.Length;
 long double tau=(sqrt(5)+1)/2;
 long double lamb2,lamb1,lamb0;
 lamb2=powl(tau,0);
 lamb1=0;
 lamb0=0;
 int i=1;
 DynamicArray < long double > xlamb2;
 DynamicArray < long double > xlamb1;
 xlamb1.set_length(n);
 xlamb2.set_length(n);
 for (int i=0;i<n;i++) xlamb1[i]=x[i] + lamb1*s[i];
 for (int i=0;i<n;i++) xlamb2[i]=x[i] + lamb2*s[i];
 while(Txmi(A,b,c,xlamb2,mi)<Txmi(A,b,c,xlamb1,mi)) {
  lamb0=lamb1;
  lamb1=lamb2;
  lamb2=lamb2+powl(tau,i);
  i++;
  for (int i=0;i<n;i++) xlamb1[i]=x[i] + lamb1*s[i];
  for (int i=0;i<n;i++) xlamb2[i]=x[i] + lamb2*s[i];
 }
 return ZlatyRez(A,b,c,x,s,mi,lamb0,lamb2,zlrez);
}

//-----faza I prim-dual-----
long double FazaI2(DynamicArray < long double > x,DynamicArray < long double > deltax,DynamicArray <
long double > z,
            DynamicArray < long double > deltaz,DynamicArray < long double > c,long double mi,int zlrez)
{
 int n=x.Length;
 int m=z.Length;
 long double tau=(sqrt(5)+1)/2;
 long double lamb2,lamb1,lamb0;
 lamb2=powl(tau,0);
 lamb1=0;
 lamb0=0;
 int k=1;
 DynamicArray < long double > xlamb2;
 DynamicArray < long double > xlamb1;
 DynamicArray < long double > zlamb2;
 DynamicArray < long double > zlamb1;
 xlamb1.set_length(n);
 xlamb2.set_length(n);
 zlamb1.set_length(m);
 zlamb2.set_length(m);
 for (int i=0;i<n;i++) xlamb1[i]=x[i] + lamb1*deltax[i];
 for (int i=0;i<n;i++) xlamb2[i]=x[i] + lamb2*deltax[i];
 for (int j=0;j<m;j++) zlamb1[j]=z[j] + lamb1*deltaz[j];
 for (int j=0;j<m;j++) zlamb2[j]=z[j] + lamb2*deltaz[j];
 while(Txmi2(zlamb2,c,xlamb2,mi)<Txmi2(zlamb1,c,xlamb1,mi)) {
  lamb0=lamb1;
  lamb1=lamb2;
  lamb2=lamb2+powl(tau,k);
```

```
   k++;
   for (int i=0;i<n;i++) xlamb1[i]=x[i] + lamb1*deltax[i];
   for (int i=0;i<n;i++) xlamb2[i]=x[i] + lamb2*deltax[i];
   for (int j=0;j<m;j++) zlamb1[j]=z[j] + lamb1*deltaz[j];
   for (int j=0;j<m;j++) zlamb2[j]=z[j] + lamb2*deltaz[j];
  }
  return ZlatyRez2(x,deltax,z,deltaz,c,mi,lamb0,lamb2,zlrez);
}

//-----vypocet lambdy pri primarnej metode-----
long double Lambda(DynamicArray < DynamicArray < long double > > A,DynamicArray < long double > b,
          DynamicArray < long double > c,DynamicArray < long double > s,
          DynamicArray < long double > x,long double mi,int zlrez)
{
  int n = s.Length;
  int m = A[0].Length;
  int stav=0;
  long double lamb0=0;
  long double lamb_=10000000000;
  for(int i=0;i<m;i++) {
    long double ais=0;
    for(int j=0;j<n;j++) {
      ais=ais + A[j][i]*s[j];
    }
    if(ais<0) {
      long double aix=-b[i];
      for(int j=0;j<n;j++)  aix=aix + A[j][i]*x[j];
      lamb0=-aix/ais;
      stav=1;
      if(lamb0<lamb_) lamb_=lamb0;
    }
  }
  if(stav==0) {
    lambpom=-1;                              //aby sme videli na vystupe, kedy nastala FAZAI
    return FazaI(A,b,c,x,s,mi,zlrez);
  }
  else {
    lambpom=lamb_;
    return ZlatyRez(A,b,c,x,s,mi,0,lamb_,zlrez);
  }
}

//-----vypocet lambdy prim-dual-----
long double Lambda2(DynamicArray < long double > x,DynamicArray < long double > deltax,DynamicArray <
long double > y,
              DynamicArray < long double > deltay,DynamicArray < long double > z,DynamicArray < long
double > deltaz,
              DynamicArray < long double > c,DynamicArray < long double > b,long double mi,int zlrez)
{
  int m = z.Length;
  int stav=0;
  long double lamb0=0;
  long double lamb_=10000000000;
  for(int j=0;j<m;j++) {
    if (y[j]<0) ShowMessage("ypsilon zaporny");
    if(deltay[j] < 0) {
      lamb0=-y[j]/deltay[j];
      stav=1;
      if(lamb0<lamb_) lamb_=lamb0;
    }
```

```
  }
  for(int j=0;j<m;j++) {
   if (z[j]<0) ShowMessage("zetko zaporne");
   if(deltaz[j] < 0) {
    lamb0=-z[j]/deltaz[j];
    stav=1;
    if(lamb0<lamb_) lamb_=lamb0;
   }
  }
  if(stav==0) {
   return FazaI2(x,deltax,z,deltaz,c,mi,zlrez);
  }
  else {
   return ZlatyRez2(x,deltax,z,deltaz,c,mi,0,lamb_,zlrez);
  }
}

//-----matica D1-----
void Djedna(DynamicArray < long double > y,DynamicArray < long double > z,
   DynamicArray < DynamicArray < long double > > D1) {
   int m = D1[0].Length;
   for (int i=0;i<m;i++) {
    for(int j=0;j<m;j++) {
     if (i==j) {
      D1[i][j]=y[i]/sqrtl(z[i]);
     }
     else D1[i][j]=0;
    }
   }
}

//-----matica D2-----
void Ddva(DynamicArray < long double > z,DynamicArray < DynamicArray < long double > > D2) {

  int m = D2[0].Length;
  for (int i=0;i<m;i++) {
    for(int j=0;j<m;j++) {
     if (i==j) {
      D2[i][j]=2*sqrtl(z[i]);
     }
     else D2[i][j]=0;
    }
   }
}

//-----primarna metoda-----
void __fastcall TForm1::Button1Click(TObject *Sender)
{
if(OpenDialog1->Execute()) {
  for(int i=0;i<OpenDialog1->FileName.Length();i++) {
   fname[i]=OpenDialog1->FileName[i+1];
  }
  fname[OpenDialog1->FileName.Length()]=0;
  ofstream G(fname);

  srand(StrToInt(Edit3->Text));                    //seed random
  int        xnula  = StrToInt(Edit4->Text);       //seed random pre x0
  const      n      = StrToInt(Edit1->Text);       //stanovenie rozmerov ulohy
  const      m      = StrToInt(Edit2->Text);
  long double ro = StrToFloat(Edit6->Text);        //vzdialenost start. bodu od opt. bodu
```

```
int        zlrez   = StrToInt(Edit7->Text);                    //stanovenie poctu iteracii zlateho rezu
int        stopka = StrToInt(Edit16->Text)+1;                  //stanovenie poctu vnutornych iteracii
clock_t start,end,dokopy;                                       //inicializacia CPU timera

//-----stanovenie rozmerov premennych-----

xopt.set_length(n);
x.set_length(n);
yopt.set_length(m);
y.set_length(m);
zopt.set_length(m);
z.set_length(m);
A.set_length(n);
H.set_length(n);
b.set_length(m);
c.set_length(n);
g.set_length(n);
s.set_length(n);
for (int i=0;i<n;i++) A[i].set_length(m);
for (int i=0;i<n;i++) H[i].set_length(n);


Generator(n,m,ro,A,H,xopt,x,y,yopt,z,zopt,b,c,g,s,xnula);    //vygenerovanie ulohy LP

long double mi=0;                                               //vypocet barieroveho parametra
long double pomm=0;
if (StrToInt(Edit14->Text)==1) {
  mi = StrToFloat(Edit5->Text);
}
else {
  for (int j=0;j<m;j++) {
    pomm = - b[j];
    for (int i=0;i<n;i++) {
      pomm=pomm + A[i][j]*x[i];
    }
    mi = mi + 2* sqrtl(pomm) * y[j];
  }
  mi=mi / (m*1000);
}


//-----vstupy-----
G << "VSTUPY:";
G.put('\n');
G << "ro:"; G << ro; G.put('\t'); G << "random:"; G << (StrToInt(Edit3->Text));  G.put('\t'); G << xnula;
G.put('\n');
G << "xopt:";
G.put('\n');
for (int i=0;i<n;i++) {
  G << xopt[i];
  if(i!=(n-1)) G.put('\t');
}
G.put('\n');
G.put('\n');

G << "yopt:";
G.put('\n');
for (int j=0;j<m;j++) {
  G << yopt[j];
  if(j!=(m-1)) G.put('\t');
```

```
}
G.put('\n');
G.put('\n');

G << "zopt:";
G.put('\n');
for (int j=0;j<m;j++) {
  G << zopt[j];
  if(j!=(m-1)) G.put('\t');
}
G.put('\n');
G.put('\n');

G << "x0:";
G.put('\n');
for(int i=0;i<n;i++) {
  G << x[i];
  if(i!=(n-1)) G.put('\t');
}
G.put('\n');
G.put('\n');

G << "y0:";
G.put('\n');
for(int i=0;i<m;i++) {
  G << y[i];
  if(i!=(m-1)) G.put('\t');
}
G.put('\n');
G.put('\n');

G << "A:";
for(int j=0;j<m;j++) {
  G.put('\n');
  for(int i=0;i<n;i++)  {
    G << A[i][j];
    if(i!=(n-1)) G.put('\t');
  }
}
G.put('\n');
G.put('\n');

G << "b:";
G.put('\n');
for(int i=0;i<m;i++) {
  G << b[i];
  if(i!=(m-1)) G.put('\t');
}
G.put('\n');
G.put('\n');

G << "c:";
G.put('\n');
for(int i=0;i<n;i++) {
  G << c[i];
  if(i!=(n-1)) G.put('\t');
}
G.put('\n');
G.put('\n');
```

```
G << "T(xopt):";
G.put('\t');
long double Txopt=Txmi(A,b,c,xopt,mi);
G << Txopt;
G.put('\n');
G.put('\n');

fxopt=0;
for(int i=0;i<n;i++) {
  fxopt = fxopt + c[i]*xopt[i];
}
G << "cxopt:";
G.put('\t');
G << fxopt;
G.put('\n');
G.put('\n');

//-----VYSTUP-----
G << "TELO:";
G.put('\n');
G.put('\n');

G << "mi";
G.put('\t');
G <<"iZR";
G.put('\t');
G << "iteracia";
G.put('\t');
G << "cx";
G.put('\t');
G << "Txmi";
G.put('\t');
G << "gradient";
G.put('\t');
G << "hessian";
G.put('\t');
G << "smer";
G.put('\t');
G << "lambda";
G.put('\t');
G << "smer*lambda";
G.put('\t');
G << "lambda_";
G.put('\t');
G << "smer*lambda_";
G.put('\t');
G << "c*x(l)-c*xopt";
G.put('\t');
G << "c*x(l)-b*y(l)";
G.put('\t');
G << "PLATI?";
G.put('\t');
G << "y-yopt";
G.put('\t');
G << "x-xopt";
G.put('\t');

G.put('\n');
G << mi;
G.put('\t');
```

```
G << zlrez;
G.put('\t');
G.put('1');
G.put('\t');

long double pom1, pom2, ax, smer, rozdiel, pomrozdiel;

dokopy=0;//CPU
//-----vnutorne iteracie-----
for (int f=1;f<stopka;f++) {
 start=clock();

 gTxmi(A,b,c,x,g,mi);                          //vypocet gradientu funkcie Txmi
 hTxmi(A,b,x,H,mi);                            //vypocet hessovej matice funkcie Txmi
 HansLaws(s,g,H);                              //vypocet smeru
 lambopt = Lambda(A,b,c,s,x,mi,zlrez);         //vypocet optimalnej dlzky smeru

  if(StrToInt(Edit12->Text)==1) {              //pri zapnuti podmienky, v pripade ze optimalna....
  if (lambopt>1) lambopt=1;                    //... dlzka kroku je vacsia ako nula, ohranici sa...
                                               //...jednotkou
 }

 for(int j=0;j<m;j++) {
  ax=-b[j];
  for(int i=0;i<n;i++) ax=ax + A[i][j]*x[i];
  if(ax<=0) {
    ShowMessage("chyba v aix pri vypocte gradientu");
    return;
  }
  ax=2*sqrt(ax);
  y[j]=mi/ax;
 }

 end=clock();//CPU
 dokopy = dokopy + end - start;//CPU

 //-----vystupy-----

 fx=0;                                         //c * x
 for(int i=0;i<n;i++) {
  fx = fx + c[i]*x[i];
 }
 G << fx;
 G.put('\t');

 G << Txmi(A,b,c,x,mi);                        // funkcna hodnota transformacnej funkcie
 G.put('\t');

 pom1=0;                                       //norma gradientu pre vystup
 pom2=0;
 for (int i=0;i<n;i++) {
  pom2=g[i]*g[i];
  pom1=pom1+pom2;
 }
 pom1=sqrtl(pom1);
 G << pom1;
 G.put('\t');

 pom1=0;                                       //norma hessovej matice pre vystup
 pom2=0;
```

```
for (int i=0;i<n;i++) {
  for (int j=0;j<n;j++) {
    pom2=H[i][j]*H[i][j];
    pom1=pom1+pom2;
  }
}
G << pom1;
G.put('\t');

smer=0;                                          //norma smeru pre vystup
pom2=0;
for (int i=0;i<n;i++) {
  smer=s[i]*s[i];
  smer=smer+pom2;
}
smer=sqrtl(smer);
G << smer;
G.put('\t');

G << lambopt;                                    //zobrazenie optimalnej dlzky kroku
G.put('\t');

pom1=lambopt*smer;                               // lambda optimalna * smer kroku
G << pom1;
G.put('\t');

if (lambpom==-1) G << "FAZAI";                   //velkost maximalnej dlzky kroku
else G << lambpom;
G.put('\t');

pom1=lambpom*smer;                               //neredukovana dlzka krok * smer kroku
if (lambpom==-1) G << "FAZAI";
else G << pom1;
G.put('\t');

G << (fx-fxopt);                                 // c * x - c * x optimalne
G.put('\t');

pom1=0;                                          //dualna medzera a jej platnost
for (int j=0;j<m;j++) {
  pom1=pom1+b[j]*y[j];
}
G << (fx-pom1);
G.put('\t');
if (pom1 <=fx) G << "Y";
else           G << "N";
G.put('\t');

start=clock();

//-----novy odhad a veci s tym spojene-----
for (int i=0;i<n;i++) x[i]=x[i]+ lambopt*s[i];

if (f % (StrToInt(Edit8->Text))==0) {            //zmena mi
  if (StrToInt(Edit13->Text)==1) mi=mi/(StrToFloat(Edit9->Text));
  else {
    mi=0;
    for (int j=0;j<m;j++) {
      pomm = - b[j];
      for (int i=0;i<n;i++) {
```

```
          pomm=pomm + A[i][j]*x[i];
        }
        mi = mi + 2* sqrtl(pomm) * y[j];
      }
      mi=mi / m;
    }
  }

  rozdiel=0;
  pomrozdiel=0;
  for (int i=0;i<n;i++) {
    pomrozdiel=x[i]-xopt[i];
    pomrozdiel=pomrozdiel*pomrozdiel;
    rozdiel=rozdiel+pomrozdiel;
  }
  rozdiel=sqrtl(rozdiel);



                                                    //stop kriteria, ak by sme dosiahli dobru...
                                                    //...aproximaciu pred pozadovanym poctom vnut....
                                                    //...iteracii
  if ( StrToInt(Edit10->Text)==2 ) {                //gradientne kriterium
    pom1=0;
    for (int i = 0;i<n;i++) pom1 = pom1 + powl(g[i],2);
    pom1=sqrtl(pom1);
    if (pom1 < StrToFloat(Edit11->Text)) f=(stopka-1);
  }
  if ( StrToInt(Edit10->Text)==3 ){                 //kriterium presnosti optimalnej funkcnej hodnoty
transformacnej funkcie
    if((Txmi(A,b,c,x,mi)-Txmi(A,b,c,xopt,mi)) < StrToFloat(Edit6->Text))  f=(stopka-1);
  }
  if ( StrToInt(Edit10->Text)==1 ){                 //kriterium kvality odhadu optimalneho bodu
    if (rozdiel < StrToFloat(Edit11->Text)) f=(stopka-1);
  }
   //kvoli zlrez inputu
  end=clock();//CPU
  dokopy = dokopy + end - start;//CPU

  pom1=0;
  pom2=0;
  for (int j=0;j<m;j++) {
    pom1=y[j]-yopt[j];
    pom1=pom1*pom1;
    pom2=pom2+pom1;
  }
  pom2=sqrtl(pom2);

  G << pom2;
  G.put('\t');
  G << rozdiel;
  G.put('\t');
  G.put('\n');
  if (f<(stopka-1)){
    G << mi;
    G.put('\t');
    G << zlrez;
    G.put('\t');
    G << f+1;
    G.put('\t');
```

```
   }
   else {
     G << "CPU Time:";
     G.put('\n');
     G << (dokopy/(double) CLK_TCK);
     G.put('\n');
   }
  }


  G.close();
  }
}
//---------------------------------------------------------------------------
```

//-----primarno - dualna metoda-----

```
void __fastcall TForm1::Button2Click(TObject *Sender)
{
if(OpenDialog1->Execute()) {
  for (int i=0;i<OpenDialog1->FileName.Length();i++) {
    fname[i]=OpenDialog1->FileName[i+1];
  }
  fname[OpenDialog1->FileName.Length()]=0;
  ofstream G(fname);

  srand(StrToInt(Edit3->Text));                    //seed random
  int        xnula  = StrToInt(Edit4->Text);       //seed random pre x0
  const      n      = StrToInt(Edit1->Text);       //stanovenie rozmerov ulohy
  const      m      = StrToInt(Edit2->Text);
  long double ro    = StrToFloat(Edit6->Text);     //vzdialenost start. bodu od opt. bodu
  int        zlrez;//  = StrToInt(Edit7->Text);   //stanovenie poctu iteracii zlateho rezu

  clock_t start, end, dokopy;

  //-----stanovenie rozmerov premennych-----

  xopt.set_length(n);
  x.set_length(n);
  yopt.set_length(m);
  y.set_length(m);
  zopt.set_length(m);
  z.set_length(m);
  A.set_length(n);
  ADDA.set_length(n);
  H.set_length(n);
  b.set_length(m);
  c.set_length(n);
  g.set_length(n);
  s.set_length(n);
  POM.set_length(m);
  POM2.set_length(m);
  D1.set_length(m);
  D2.set_length(m);
  D2inv.set_length(m);
  rb.set_length(m);
  rc.set_length(m);
  rmi.set_length(m);
  r1.set_length(m);
  r.set_length(n);
```

```
deltax.set_length(n);
deltay.set_length(m);
deltaz.set_length(m);
for (int i=0;i<n;i++) A[i].set_length(m);
for (int i=0;i<n;i++) H[i].set_length(n);
for (int j=0;j<m;j++) D1[j].set_length(m);
for (int j=0;j<m;j++) D2[j].set_length(m);
for (int j=0;j<m;j++) POM[j].set_length(n);
for (int j=0;j<m;j++) POM2[j].set_length(n);
for (int j=0;j<m;j++) D2inv[j].set_length(m);
for (int i=0;i<n;i++) ADDA[i].set_length(n);
long double pom1;

int       stopka = StrToInt(Edit16->Text)+1;                    //stanovenie poctu vnutornych iteracii

Generator(n,m,ro,A,H,xopt,x,y,yopt,z,zopt,b,c,g,s,xnula);     //vygenerovanie ulohy

dokopy=0;//CPU
start=clock();//CPU

long double mi=0;                                               //vypocet barieroveho parametra
if (StrToInt(Edit14->Text)==1) {
  mi = StrToFloat(Edit5->Text);
}
else {
  for (int j=0;j<m;j++) {
  mi = mi + 2*y[j]*sqrtl(z[j]);
  }
  mi=mi/(m*1000);
}

end=clock();//CPU
dokopy = dokopy + end - start;//CPU

//-----vstupy-----
G << "VSTUPY:";
G.put('\n');

G << "xopt:";
G.put('\n');
for (int i=0;i<n;i++) {
  G << xopt[i];
  if(i!=(n-1)) G.put('\t');
}
G.put('\n');
G.put('\n');

G << "yopt:";
G.put('\n');
for (int j=0;j<m;j++) {
  G << yopt[j];
  if(j!=(m-1)) G.put('\t');
}
G.put('\n');
G.put('\n');

G << "zopt:";
G.put('\n');
for (int j=0;j<m;j++) {
  G << zopt[j];
```

```
  if(j!=(m-1)) G.put('\t');
}
G.put('\n');
G.put('\n');

G << "x0:";
G.put('\n');
for(int i=0;i<n;i++) {
  G << x[i];
  if(i!=(n-1)) G.put('\t');
}
G.put('\n');
G.put('\n');

G << "y0:";
G.put('\n');
for (int j=0;j<m;j++) {
  G << y[j];
  if(j!=(m-1)) G.put('\t');
}
G.put('\n');
G.put('\n');

G << "z0:";
G.put('\n');
for (int j=0;j<m;j++) {
  G << z[j];
  if(j!=(m-1)) G.put('\t');
}
G.put('\n');
G.put('\n');

G << "A:";
for(int j=0;j<m;j++) {
  G.put('\n');
  for(int i=0;i<n;i++)  {
    G << A[i][j];
    if(i!=(n-1)) G.put('\t');
  }
}
G.put('\n');
G.put('\n');

G << "b:";
G.put('\n');
for(int i=0;i<m;i++) {
  G << b[i];
  if(i!=(m-1)) G.put('\t');
}
G.put('\n');
G.put('\n');

G << "c:";
G.put('\n');
for(int i=0;i<n;i++) {
  G << c[i];
  if(i!=(n-1)) G.put('\t');
}
G.put('\n');
G.put('\n');
```

```
G << "Optimalna Txzmi:" ;
G.put('\n');
G << Txmi2(zopt,c,xopt,mi);
G.put('\n');
G.put('\n');

fxopt=0;
for(int i=0;i<n;i++) {
  fxopt = fxopt + c[i]*xopt[i];
}
G << "cxopt:";
G.put('\t');
G << fxopt;
G.put('\n');
G.put('\n');

G << "skutocne ro:";
G.put('\n');
pom1=0;
for(int i=0;i<n;i++) pom1=pom1+(x[i]-xopt[i])*(x[i]-xopt[i]);
for (int j=0;j<m;j++) {
  pom1=pom1+(y[j]-yopt[j])*(y[j]-yopt[j])+(z[j]-zopt[j])*(z[j]-zopt[j]);
}
pom1=sqrtl(pom1);
G << pom1;
G.put('\n');
G.put('\n');

G << "ro:";
G.put('\n');
G << ro;
G.put('\n');
G.put('\n');

G << "rozmer n:";
G.put('\t');
G << n;
G.put('\n');

G << "rozmer m:" ;
G.put('\t');
G << m;
G.put('\n');


//-----VYSTUP-----
G << "TELO:";
G.put('\n');
G.put('\n');

G << "mi";
G.put('\t');
G <<"iZR";
G.put('\t');
G << "iteracia";
G.put('\t');
G << "c*x";
G.put('\t');
G << "b*y";
```

```
G.put('\t');
G << "Txzmi";
G.put('\t');
G << "rb";
G.put('\t');
G << "rc";
G.put('\t');
G << "rmi";
G.put('\t');
G << "Spadovost";
G.put('\t');
G << "lambda";
G.put('\t');
G << "deltax";
G.put('\t');
G << "deltax*lambda";
G.put('\t');
G << "deltay";
G.put('\t');
G << "deltay*lambda";
G.put('\t');
G << "deltaz";
G.put('\t');
G << "deltaz*lambda";
G.put('\t');
G << "x-xopt";
G.put('\t');
G << "y-yopt";
G.put('\t');
G << "z-zopt";
G.put('\n');

long double spadovost, pom, pom3;

start=clock();//CPU

//-----vnutorne iteracie-----
for (int f=1;f<stopka;f++) {
  Djedna(y,z,D1);                                   //vypocet diagonalnej matice D1
  Ddva(z,D2);                                       //vypocet diagonalnej matice D2
  InverznaMaticaD2(D2,D2inv);                       //vypocet inverznej matice k diagonalnej matici D2
  for (int i=0;i<m;i++) {                           //zaciatok vypoctu ADDA
    for (int j=0;j<n;j++) {
      POM[i][j]=0;
      for (int k=0;k<m;k++) {
        POM[i][j]=POM[i][j]+A[j][k]*D2inv[i][k];    //vypocet AD2inv
      }
    }
  }
  for (int i=0;i<m;i++) {
    for (int j=0;j<n;j++) {
      POM2[i][j]=0;
      for (int k=0;k<m;k++) {
        POM2[i][j]=POM2[i][j]+POM[k][j]*D1[i][k];
      }
    }
  }
  for (int i=0;i<n;i++) {
    for (int j=0;j<n;j++) {
      ADDA[i][j]=0;
```

```
   for (int k=0;k<m;k++) {
     ADDA[i][j]=ADDA[i][j]+POM2[k][j]*A[i][k];          //koniec vypoctu ADDA
    }
  }
}
for (int j=0;j<m;j++) {                                 //vypocet rezidua rb
  rb[j]=z[j]+b[j];
  for (int i=0;i<n;i++) {
    rb[j]=rb[j]-A[i][j]*x[i];
  }
}
for (int i=0;i<n;i++) {                                 //vypocet rezidua rc
  rc[i]=c[i];
  for (int j=0;j<m;j++) {
    rc[i]=rc[i]-A[i][j]*y[j];
  }
}
for (int j=0;j<m;j++) {                                 //vypocet rezidua rmi
  rmi[j]=mi-2*y[j]*sqrtl(z[j]);
}
for (int j=0;j<m;j++) {                                 //zaciatok vypoctu celkoveho rezidua r
  r1[j]=0;
  for (int i=0;i<m;i++) {
    r1[j]=r1[j]+D1[i][j]*rb[j];                         //vypocet D1rb
  }
  r1[j]=r1[j]+rmi[j];                                   //vypocet rmi+D1rb
}
for (int i=0;i<n;i++) {
  r[i]=0;
  for (int j=0;j<m;j++) {
    r[i]=r[i]+POM[j][i]*r1[j];                          //vypocet AtD2inv(rmi+D1rb)
  }
  r[i]=r[i]-rc[i];                                      //vypocet AtD2inv(rmi+D1rb)-rc
                                                        //koniec vypoctu celkoveho rezidua r
}
HansLaws(deltax,r,ADDA);                                //vypocet delta x
for (int i=0;i<n;i++) {
  deltax[i]=-deltax[i];
}
for (int j=0;j<m;j++) {                                 //vypocet delta z
  deltaz[j]=-rb[j];
  for (int i=0;i<n;i++) {
    deltaz[j]=deltaz[j]+A[i][j]*deltax[i];
  }
}
for (int j=0;j<m;j++) {                                 //vypocet delta y
  deltay[j]=D2inv[j][j]*(rmi[j]-D1[j][j]*deltaz[j]);
}
end=clock();//CPU
dokopy = dokopy + end - start;//CPU

spadovost=0;                                            //je smer spadovy?
for (int i=0;i<n;i++) {
  spadovost=spadovost + c[i]*deltax[i];
}
for (int j=0;j<m;j++) {
  spadovost=spadovost - (mi*deltaz[j])/(2*sqrtl(z[j]));
}

start=clock();//CPU
```

```
lambopt = Lambda2(x,deltax,y,deltay,z,deltaz,c,b,mi,zlrez); //vypocet optimalnej dlzky kroku
if(StrToInt(Edit12->Text)==1) {                          //pri zapnuti podmienky, v pripade ze optimalna...
                                                         //...dlzka kroku je vacsia ako nula, ohranici sa...
  if (lambopt>1) lambopt=1;                               //...jednotkou
}

end=clock();//CPU
dokopy = dokopy + end - start;//CPU

//-----vystupy-----

G << mi;                                                 //velkost barieroveho parametra
G.put('\t');

G << zlrez;                                              //pocet iteracii zlateho rezu
G.put('\t');

G << f;                                                  //poradove cislo vnutornej iteracie
G.put('\t');

pom=0;
for (int i=0;i<n;i++) pom=pom+c[i]*x[i];                 //hodnota c * x
G << pom;
G.put('\t');

pom=0;
for (int j=0;j<m;j++) pom=pom + b[j]*y[j];               //hodnota b * y
G << pom;
G.put('\t');

G << Txmi2(z,c,x,mi);                                    //funkcna hodnota transformacnej funkcie
G.put('\t');

pom=0;
for (int j=0;j<m;j++) pom=pom+rb[j]*rb[j];               //velkost rezidua F(rb)
G << sqrtl(pom);
G.put('\t');

pom=0;
for (int i=0;i<n;i++) pom=pom+rc[i]*rc[i];               //velkost rezidua F(rc)
G << sqrtl(pom);
G.put('\t');

pom=0;
for (int j=0;j<m;j++) pom = pom + rmi[j]*rmi[j] ;        //velkost rezidua F(rmi)
G << sqrtl(pom);
G.put('\t');

if (spadovost < 0) G << "<0";                            //plati spadovost?
else G << ">=0";
G.put('\t');
                                                         //optimalna dlzka kroku
G << lambopt;
G.put('\t');

pom=0;
for(int i=0;i<n;i++) pom=pom+deltax[i]*deltax[i];        //velkost delty x
G << sqrtl(pom);
G.put('\t');
```

```
G << lambopt*pom;                                   //velkost kroku v x
G.put('\t');

pom=0;
for (int j=0;j<m;j++) pom=pom+deltay[j]*deltay[j];  //velkost delty y
G << sqrtl(pom);
G.put('\t');

G << lambopt*pom;                                   //velkost kroku v y
G.put('\t');

pom=0;                                              //velkost delty z
for (int j=0;j<m;j++) pom=pom+deltaz[j]*deltaz[j];
G << sqrtl(pom);
G.put('\t');

G << lambopt*pom;                                   //velkost kroku v z
G.put('\t');

start=clock();//CPU

for (int i=0;i<n;i++) x[i]=x[i]+lambopt*deltax[i];  //nova aproximacia
for (int j=0;j<m;j++) y[j]=y[j]+lambopt*deltay[j];
for (int j=0;j<m;j++) z[j]=z[j]+lambopt*deltaz[j];

pom3=0;
for (int i=0;i<n;i++) pom3=pom3+(x[i]-xopt[i])*(x[i]-xopt[i]);  //odchylka aproximacie od optimalneho x
G << sqrtl(pom3);
G.put('\t');

pom=0;
for (int j=0;j<m;j++) pom=pom+(y[j]-yopt[j])*(y[j]-yopt[j]);  //odchylka aproximacie od optimalneho y
G << sqrtl(pom);
G.put('\t');

pom=0;
for (int j=0;j<m;j++) pom=pom+(z[j]-zopt[j])*(z[j]-zopt[j]);  //odchylka aproximacie od optimalneho z
G << sqrtl(pom);
G.put('\t');
G.put('\n');

if (f % (StrToInt(Edit8->Text))==0) {               //zmena barieroveho parametra...
  if (StrToInt(Edit13->Text)==1) {
    mi=mi/(StrToFloat(Edit9->Text));                //...ako prirastok ro (0<ro<1)
  }
  else {
    mi=0;                                           //...pomocou MNS
    for (int j=0;j<m;j++) {
      mi = mi + 2*y[j]*sqrtl(z[j]);
    }
    mi=mi/m;
  }
}

                                                    //stop kriteria, ak by sme dosiahli dobru...
                                                    //...aproximaciu prd pozadovanym poctom vnut....
                                                    //...iteracii
if ( StrToInt(Edit10->Text)==3 ){                   //kriterium presnosti optimalnej funkcnej hodnoty..
                                                    //... transformacnej funkcie
```

```
      if((Txmi2(z,c,x,mi)-Txmi2(z,c,xopt,mi)) < StrToFloat(Edit6->Text))  f=(stopka-1);
    }
    if ( StrToInt(Edit10->Text)==1 ){                    //kriterium kvality odhadu optimalneho bodu
      if (sqrtl(pom3) < StrToFloat(Edit11->Text)) f=(stopka-1);
    }

    end=clock();//CPU
    dokopy = dokopy + end - start;//CPU
    if (f==(stopka-1)) {
      G << "CPU Timer:";
      G.put('\n');
      G << (dokopy/(double) CLK_TCK);
      G.put('\n');
    }
   }

  G.close();
  }
}
//---------------------------------------------------------------------------
```