Vybrané metódy riešenia nelineárnej úlohy o komplementarite

DIPLOMOVÁ PRÁCA

Ondrej Fekete

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY FYZIKY A INFORMATIKY

KATEDRA APLYKOVANEJ MATEMATIKY A ŠTATISTIKY

ŠTUDIJNÝ ODBOR:
EKONOMICKÁ A FINANčNÁ MATEMATIKA

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

BRATISLAVA 2008

Čestne prehlasujem, že som diplomovú prácu vypracoval samostatne a použil som len literatúru uvedenú v zozname.

....

Poďakovanie: V prvom rade chcem vysloviť vďaku svojmu vedúcemu diplomovej práce doc. RNDr. Milanovi Hamalovi CSc. S pomocou jeho odborného vedenia, cených rád a hlavne veľkej trpezlivosti som našiel vždy inšpiráciu pokračovať a nevzdávať sa.
Ďalej chcem poďakovať tým, ktorých práca ma sprevádzala počas môjho vysokoškolského štúdia, jednak učiteľom a zamestnancom FMFI či autorom učebníc a iných materiálov, z ktorých som počas štúdia čerpal vedomosti a inšpiráciu.
Nakoniec chcem vysloviť vďaku ľuďom, ktorí ma podporovali v ťažších chvíľach, hlavne rodičom a súrodencom, za ich zhovievavosť s mojimi chybami.

Obsah

Úvod

Znalosť spoľahlivých algoritmov, ktoré riešia úlohy o komplementarite má nezanedbateľný význam v oblasti matematického programovania. Komplementárne úlohy a osobitne úloha o nelineárnej komplementarite (angl. nonlinear complementarity problem - NCP) majú totiž široké uplatnenie pri riešení ekonomických a inžinierskych problémov. Cieľom tejto práce je preskúmanie určitých prístupov k numerickému riešeniu spomenutej NCP, čo zahŕňa naprogramovanie a otestovanie algoritmu podľa naštudovanej metódy.

Na začiatok je predsavená samotná NCP a niektoré iné komplementárne úlohy. Ukážeme ako je formulovaná úloha NCP a aké vlastnosti môže mať v závislosti od druhu zobrazenia F.

V druhej časti sú načrtnuté určité prístupy pri riešení nelineárnej úlohy o komlementarite. Popíšeme Newtonovu metódu, pomocou ktorej môžme riešiť hladký systém nelineárnych rovníc.

Tretia kapitola je zameraná na konkrétnu metódu - Chenov algoritmus. Popíšeme vyhladenie funkcie $(z)_+$, ktorá nie je diferencovateľná a následne jeho využitie v rámci algoritmu založenom na Newtonovej metóde.

Iný - zložitejší algoritmus sa vyskytuje v práci [5], ktorú podľa autora označíme ako metódu Fachinei.

V poslednej časti sa budeme venovať numerickému riešeniu NCP pomocou naprogramovaného algoritmu na základe Chenovej metódy v programe MATLAB. Pomocou neho budeme riešiť skupinu úloh náhodne vygenerovaných na základe zvoleného kľúča.

Kapitola 1

Klasifikácia úloh o komplementarite

Komplementarita sa dá popísať ako ortogonalita dvoch nezáporných vektorov.

Definícia 1.0.1. *Dva vektory x,$y \in R^n$ nazývame komplementárne práve vtedy, keď*

$$x \geq 0_n, \quad y \geq 0_n, \quad x^T y = 0 \tag{1.1}$$

Rovnicu $x^T y = 0$ označujeme ako podmienku komplementarity a môžme ju vyjadriť v zložkovom tvare

$$x_1.y_1 + x_2.y_2 + ... + x_i.y_i + ... + x_n.y_n = 0 \tag{1.2}$$

Keďže

$$x_i.y_i \geq 0, \quad i = 1, 2, ..., n$$

vidíme, že komplementarita dvoch nezáporných vektorov je vlastne komplementarita ich jednotlivých zložiek

$$x \geq 0_n, \quad y \geq 0_n, \quad x_i.y_i = 0, \quad i = 1, 2, ..., n \tag{1.3}$$

1.1 Nelineárna úloha o komplementarite a existencia jej riešenia

Popíšeme základnú úlohu ktorej sa venujeme v tejto práci. Ide o takzvanú nelineárnu úlohu o komplementarite (NCP-Nonlinear Complementarity Problem).

Formulácia úlohy NCP(F): Nech je dané zobrazenie $F : R^n \to R^n$ Nájdite $x \in R^n$ také, aby platilo

$$x \geq 0_n \quad F(x) \geq 0_n \quad x^T F(x) = 0 \tag{1.4}$$

Tak ako sme (1.1) vyjadrili v tvare (1.3), môžme (1.4) vyjadriť taktiž v zložkovom tvare

$$x \geq 0_n \quad F(x) \geq 0_n \quad x_i.F_i(x) = 0 \quad \forall i \tag{1.5}$$

Všimnime si, že podľa formulácie úlohy NCP(F) nie je zaručená existencia riešenia tejto úlohy, čo ukážeme v nasledujúcom príklade.

Príklad 1.1.1. *Majme NCP(F) kde*

$$F(x_1, x_2) = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

Potom $(\hat{x}_1, \hat{x}_2)^T$ *bude riešením NCP(F) práve vtedy, keď*

$$\begin{pmatrix} \hat{x}_1 \\ \hat{x}_2 \end{pmatrix} \geq \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \begin{pmatrix} \hat{x}_2 + 1 \\ \hat{x}_1 - 1 \end{pmatrix} \geq \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \begin{pmatrix} \hat{x}_1.(\hat{x}_2 + 1) \\ \hat{x}_1.(\hat{x}_2 + 1) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Z toho že $\hat{x}_2 \geq 0$ *a* $\hat{x}_1.(\hat{x}_2 + 1) = 0$ *máme* $\hat{x}_1 = 0$ *čo je v rozpore s* $\hat{x}_1 - 1 \geq 0$. *Uvedená úloha NCP(F) teda nemá riešenie.*

V nasledujúcom príklade ukážeme ďalšie možnosti, ktoré môžu nastať.

Príklad 1.1.2. *Nech funkcia $F : R \to R$, potom*

1. *ak $F(x) = -x - 1$, NCP(F) nemá riešenie, keďže $F(x) < 0$ pre všetky $x > 0$*
2. *ak $F(x) = x - 1$, NCP(F) má práve jedno riešenie, pretože $F(x) \geq 0$ len ak $x \geq 1$ a $F(x) = 0$ len ak $x = 1$. Tým pádom jediné riešenie je $\hat{x} = 1$*
3. *ak $F(x) \equiv 0$, potom riešením NCP(F) je každé $x \geq 0$. Všimnime si, že množina riešení je v tomto prípade kompaktná.*
4. *ak $F(x) = \sin(x)$, potom sú riešnia NCP(F) dané $\hat{x}^k = k.\pi$ pre $k = 0, 1, 2, \ldots.$*

Aby sme mohli formulovať tvrdenia ohladne riešení NCP(F) uvedieme niekoľko tried transformačných zobrazení.

Definícia 1.1.1. *Zobrazenie $F : R^n \to R^n$ nazývame*

1. *monotónne ak $(x-y)^T(F(x) - F(y)) \geq 0$, $\forall x, y \in R^n$*
2. *rýdzomonotónne ak $(x-y)^T(F(x) - F(y)) > 0$, $\forall x, y \in R^n, x \neq y$*
3. *silnomonotónne ak existuje konštanta $\mu > 0$ taká, že $(x-y)^T(F(x) - F(y)) \geq \mu\|x - y\|$, $\forall x, y \in R^n$*

Veta 1.1.1.

1. *Ak $F : R^n \to R^n$ je monotónne, množina všetkých riešení prislúchajúcich k úlohe NPC(F) je konvexná (môže byť prázdna).*
2. *Ak F je rýdzomonotónne, NPC(F) má najviac jedno riešenie.*
3. *Ak F je silnomonotónne, NPC(F) má práve jedno riešenie.*

Vzhľadom na vetu (1.1.1) je zabezpečená existencia riešenia len v prípade silnomonotónnej NCP(F). V konkrétnom prípade však nemusí byť potrebná ani monotónnosť F

1.2 Lineárna úloha o komplementarite

Špeciálnym prípadom NCP(F) je úloha s lineárnym zobrazením F. Vtedy hovoríme o lineárnej úlohe o komplementarite (LCP-Linear Complementarity Problem).

Formulácia úlohy LCP(A,b): Nech je daná n-rozmerná matica A a vektor $b \in R^n$. Nájdite $x \in R^n$ také, aby platilo

$$x \geq 0_n \quad Ax + b \geq 0_n \quad x^T(Ax + b) = 0 \tag{1.6}$$

1.3 Zmiešaná úloha o komplementarite

Iným typom komplementárnej úlohy je zovšeobecnenie NCP. Pôvodne sme uvažovali o zobrazení z kladného priestoru na kladný priestor. Vo zovšeobecnení obmedzujeme definičný obor zobrazenia na danú množinu B definovanú dolnou a hornou hranicou, pričom v prípade, že istá zložka obrazu nie je nulová požadujeme aby príslušná zložka zobrazovaného bodu ležala na dolnej hranici v prípade kladného zobrazenia, alebo hornej v prípade záporného zobrazenia. Toto zovšeobecnenie sa nazýva zmiešaná úloha o komplementarite(MCP - Mixed Complementarity Problem)

Formulácia úlohy MCP(F,l,u): Nech l,$u \in R^n$, také že $l_i \leq u_i \ \forall i$. Množina B je daná ako tie $x \in R^n$ pre ktoré platí $l_i \leq x_i \leq u_i \ \forall i$. Ďalej máme dané zobrazenie $F : R^n \rightarrow R^n$. Nájdite také $x \in B$ a w,$v \in R^n$ pre ktoré platí

$$w \geq 0_n, \quad v \geq 0_n, \quad x \in B \tag{1.7}$$

$$F(x) = w - v, \quad w^T(x - l) = 0, \quad v^T(u - x) = 0 \tag{1.8}$$

Podmienky (1.7) a (1.8) zapíšeme v tvare

$$w \geq 0_n, \quad v \geq 0_n, \quad x \in B \tag{1.9}$$

$$F_i(x) = w_i - v_i, \quad w_i.(x_i - l_i) = 0, \quad v_i.(u_i - x_i) = 0 \quad \forall i \tag{1.10}$$

Vektory w a v zohrávajú len vedľajšiu úlohu o čom vypovedá nasledujúca veta.

Veta 1.3.1. *Majme MCP(F,l,u) tak ako bola formulovaná. Ak existuje x ktoré spĺňa podmienky* (1.9) *a* (1.10) *pre nejaké w a v, potom budú tieto podmienky splnené aj pre nasledovné w a v:*

$$\begin{array}{lllll} F_i(x) = 0 & \Rightarrow & w_i = 0, & & v_i = 0 \\ F_i(x) > 0 & \Rightarrow & w_i = F_i(x), & & v_i = 0 \\ F_i(x) < 0 & \Rightarrow & w_i = 0, & & v_i = F_i(x) \end{array} \tag{1.11}$$

Dôkaz:

- *Ak $x \in B$ a $F_i(x) = 0$ potom sú* (1.9) *a* (1.10) *splnené pre $w = v = 0$.*
- *Ak $F_i(x) > 0 \Rightarrow w_i > 0 \Rightarrow x_i - l_i = 0$ a z toho sú* (1.9) *a* (1.10) *splnené pre $v_i = 0$.*
- *Ak $F_i(x) < 0 \Rightarrow v_i > 0 \Rightarrow u_i - x_i = 0$ a z toho sú* (1.9) *a* (1.10) *splnené pre $w_i = 0$.*

Z vety (1.3.1) máme inú definíciu MCP*, ktorá je však ekvivalentná s tou predchádzajúcou.

Formulácia úlohy MCP*(F,l,u): Nech l,$u \in R^n$, také že $l_i \leq u_i$ $\forall i$. Množina B je daná ako tie $x \in R^n$ pre ktoré platí $l_i \leq x_i \leq u_i$ $\forall i$. Ďalej máme zobrazenie $F : R^n \rightarrow R^n$.
Hľadáme také x, ktoré vyhovuje jednej z nasledujúcich podmienok

1. $F_i(x) > 0$ a $z_i = l_i$
2. $F_i(x) < 0$ a $z_i = u_i$
3. $F_i(x) = 0$ a $z_i \in B$

NCP(F) je špeciálnym prípadom MCP(F,l,u) a síce pre $l_i = 0$ a $u_i = +\infty$ $\forall i$

1.4 Úlohy o komplementarite a podmienky optimality v nelineárnom programovaní

Pri úlohách nelineárneho programovania(NP-nonlinear programming) sa stretávame s takzvanými Kuhn-Tuckerovými(KT) podmienkami, ktoré sú súčasťou nutných podmienok optimality niektorých diferencovateľných úloh NP. V nasledujúcom ukážeme, že pre úlohu NP s ohraničeniami v tvare nerovností na kladnom ortante sú KT podmienky ekvivalentné úlohe NCP(F).

Formulácia úlohy NP(Φ,g): Majme zobrazenia $\Phi : R^k \rightarrow R$ a $g : R^k \rightarrow R^l$. Nájdite také $\hat{z} \in R^k$ ktoré rieši:

$$\begin{gathered}\mathrm{MIN}_z \Phi(z) \\ g(z) \leq 0 \\ z \geq 0\end{gathered}$$

Ak sú Φ a g diferencovateľné, potom k tejto úlohe prislúchajú spomínané KT podmienky.

Definícia 1.4.1. *Nech*

$$z,\Phi,g \textit{ ako v NP,diferencovateľné a } \lambda \in R^k$$

potom $L(z,\lambda)$ *je vektor Lagrangeových multiplikátorov práve vtedy, keď*

$$L(z,\lambda) = \Phi(z) + \lambda^T g(z)$$

L *je Lagrangeova funkcia a* λ

Definícia 1.4.2. *Nech je L Lagrangeova funkcia. Potom $\hat{z}$ spĺňa Kuhn-Tuckerove podmienky, práve vtedy,keď existuje také $\hat{\lambda}$, že platí nasledovné:*

$$\frac{\partial L}{\partial \hat{z}} \geq 0, \qquad \frac{\partial L}{\partial \hat{\lambda}} \leq 0 \tag{1.12}$$

$$\frac{\partial L}{\partial \hat{z}}.\hat{z}_i = 0, \quad \frac{\partial L}{\partial \hat{\lambda}}.\hat{\lambda}_i = 0 \tag{1.13}$$

$$\hat{z} \geq 0, \qquad \hat{\lambda} \geq 0 \tag{1.14}$$

Použijeme označenie

$$n = k + l, \quad x = \begin{pmatrix} z \\ \lambda \end{pmatrix}, \quad F(x) = \begin{pmatrix} \frac{\partial L}{\partial z} \\ \\ -\frac{\partial L}{\partial \lambda} \end{pmatrix}$$

Ak $\hat{z}$ spĺňa KT podmienky, potom existuje $\hat{\lambda}$ také, že pre $\hat{x} = (\hat{z}^T, \hat{\lambda}^T)^T$ platí nasledovné:

$$\hat{x} \geq 0, \quad F(\hat{x}) \geq 0, \quad \hat{x}^T F(\hat{x}) = 0$$

Súvis s nutnými podmienkami optimality NP dáva nasledujúce tvrdenie.

Veta 1.4.1. *Ak sú splnené podmienky regularity a $\hat{x}$ je riešením NP, potom $\hat{x}$ spĺňa KT podmienky.*

Analogický súvis medzi optimalizačnými a komplementérnymi úlohami môžeme nájsť v iných prípadoch. Konkrétne úloha kvadratického programovania(Quadratic Programming - QP) vedie k lineárnej úlohe o komplementarite.

Formulácia úlohy QP(Φ,g):

$$\begin{gathered} \mathrm{MIN}_z \tfrac{1}{2} z^T Q z + r^T \\ Sz + k \leq 0 \\ z \geq 0 \end{gathered}$$

Tvar KT podmienok

$$L(z, \lambda) = \frac{1}{2} z^T Q z + r^T + \lambda^T (Sz + k)$$

$$Qz + r + S^T \lambda \geq 0 \qquad Sz + k \leq 0 \tag{1.15}$$

$$z^T Q z + z^T r + z^T S^T \lambda = 0 \quad \lambda^T (Sz + k) = 0 \tag{1.16}$$

$$z \geq 0 \qquad \lambda \geq 0 \tag{1.17}$$

Preznačenie premenných

$$x = \begin{pmatrix} z \\ \lambda \end{pmatrix} \quad T = \begin{pmatrix} Q & S^T \\ -S & 0 \end{pmatrix} \quad u = \begin{pmatrix} r \\ -k \end{pmatrix}$$

KT podmienky po preznačení

$$x \geq 0 \tag{1.18}$$
$$Tx + u \geq 0 \tag{1.19}$$
$$x^T(Tx + u) = 0 \tag{1.20}$$

Kapitola 2

Prístupy k riešeniu nelineárnej úlohy o komplementarite

Napriek tomu, že riešenie nelineárnej úlohy o komplementarite je pomerne novou disciplínou, zohrávajú pri jej riešení doležitú úlohu metódy vyvinuté pre riešenie starších problémov,ako minimalizačná úloha s voľnými premennými, prípadne riešenie systému nelineárnych rovníc. Dôvodom pre tento jav je, že nelineárna úloha o komplementarite sa dá transformovať na inú, pre ktoré sú známe efektívne metódy riešenia.
Podstatnou otázkou sa stáva - ako previesť komplementárnu úlohu tak, aby sme mohli využiť nejaký známy efektívny algoritmus. V tejto kapitole si ukážeme zopár príkladov.

2.1 Projekčné metódy

V tomto prípade je cieľovou úlohou hľadanie pevného bodu zobrazenia. Pre NCP(F) zvoľme funkciu

$$f_i(x, F) = (x - \mu F(x))_+$$

kde μ je kladná konštanta a funkcia $(z)_+ \equiv \max(z, 0)$.

Veta 2.1.1. *Bod $\hat{x}$ je riešením NCP(F) práve vtedy, keď platí*

$$\hat{x}_i = (\hat{x}_i - \mu F_i(\hat{x}))_+ \quad \forall i \tag{2.1}$$

Dôkaz:"⇒" Nech $\hat{x}$ je riešenie NCP(F), potom

- *ak* $\hat{x}_i > 0 \Rightarrow F_i(\hat{x}) = 0 \Rightarrow (\hat{x}_i - \mu F_i(\hat{x}))_+ = (\hat{x}_i)_+ = \hat{x}_i$
- *ak* $\hat{x}_i = 0 \Rightarrow (\hat{x}_i - \mu F_i(\hat{x}))_+ = (-\mu F_i(\hat{x}))_+ = 0 = \hat{x}_i$

"$\Leftarrow$" Nech $\hat{x}$ spĺňa (2.1), z toho máme $\hat{x}_i \geq 0$ a ďalej $\mu F_i(\hat{x}) \geq 0$ $\forall i$. Naviac

- *ak $\hat{x} > 0 \Rightarrow 0 < (\hat{x}_i - \mu F_i(\hat{x}))_+ = \hat{x}_i - \mu F_i(\hat{x}) = \hat{x}_i \Rightarrow F_i(\hat{x}) = 0$*
- *ak $F_i(\hat{x}) > 0 \Rightarrow \hat{x}_i - \mu F_i(\hat{x}) < 0 \Rightarrow \hat{x}_i = (\hat{x}_i - \mu F_i(\hat{x}))_+ = 0 \Rightarrow \hat{x}_i = 0$*

Predpokladáme F(x) silne monotónne a Lipschitzovské. V tom prípade sú splnené predpoklady Banachovej vety o pevnom bode pre dostatočne malé μ a tým pádom algoritmus

$$x_i^{k+1} = (x_i^k - \mu F_i(x^k))_+$$

zaručuje globálnu konvergenciu.

Iné zobrazenie využíva takzvaná extragradientná metóda

$$x_i^{k+1} = (x_i^k - \mu F_i((x^k - \mu F_i(x^k))_+))_+$$

,ktorá požaduje slabší predpoklad monotónnosti namiesto silnej monotónnosti.

Hlavnou nevýhodou metódy je nanajvýš lineárna konvergencia, kvôli čomu je využívaný skôr v prípade nehladkých NCP, keďže nevyužíva pri výpočte derivácie.

2.2 Transformácia pomocou NCP-funkcie

Možnosťou riešena NCP je transformácia na systém nelineárnych rovníc (SNR), konkrétne využitím takzvaných NCP-funkcií.

Definícia 2.2.1. *Funkciu $c : R^2 \rightarrow R$ nazývame NCP-funkcia práve vtedy, keď*

$$c(a,b) = 0 \Leftrightarrow a \geq 0, b \geq 0, a.b = 0$$

Prirodzenou snahou je dosiahnuť hladkú NCP-funkciu. Jedna taká pochádza od Mangasariana z roku 1976

$$c(a,b) = (a-b)^2 - |a|.a - |b|.b$$

Iným príkladom NCP-funkcie je

$$c(a,b) = a - (a-b)_+$$

Využitiu tejto funkcie v rámci konkrétnej metódy sa venujeme v ďalšej kapitole.

Ako ďalší objekt definujeme e-oprátor pre konkrétnu NCP nasledovne. $E(x)$ je e-operátor pre NCP(F) práve vtedy, keď

$$E(x) = \begin{pmatrix} c(x_1, F_1(x)) \\ c(x_2, F_2(x)) \\ \vdots \\ c(x_n, F_n(x)) \end{pmatrix}$$

,kde $c(a, b)$ je NCP-funkcia. Z tohoto následne vyplýva, že ak $E(x)$ je e-operátor pre NCP(F) potom $\hat{x}$ je riešením NCP práve vtedy, keď rieši systém nelineárnych rovníc

$$E(\hat{x}) = 0 \tag{2.2}$$

2.3 Funkcia kvality

Iným prípadom je transformácia na minimalizačný problém s pomocou funkcie kvality. Funkcia kvality (angl. merit function), ako napovedá názov, má za úlohu ohodnotiť blýzkosť zvoleného bodu k riešeniu problému.

Definícia 2.3.1. *Funkcia $M : R^n \to R_+$ je funkcia kvality pre NCP, keď spĺňa*

$$M(\hat{x}) = 0 \Leftrightarrow \hat{x} \textit{ je riešnie NCP}$$

Veta 2.3.1. *Bod $\hat{x}$ je iešenie NCP práve vtedy, keď rieši úlohu*

$$\begin{gathered} MIN_x M(x) \\ x \in R^n \end{gathered}$$

s funkčnou hodnotou v optimálnom riešení $\hat{x}$ rovnou $M(\hat{x}) = 0$.

Jednoduchou možnosťou je použiť pri tvorbe funkcie kvality e-operátor. Dostaneme funkciu

$$M(x) = \frac{1}{2} E(x)^T E(x)$$

ktorá sa nazýva prirodzená funkcia kvality.

V článku [5] sú spomenuté niektoré funkcie kvality aj s predpokladmi globálnej konvergencie algoritmu. V tom istom článku je vysvetlené využitie funkcií kvality v rámci iných metód, k čomu sa ešte dostaneme v ďalšej kapitole.

2.4 Newtonova metóda

Základným predpokladom fungovania Newtonovej metódy pre riešenie sústavy nelineárnych rovníc $E(x) = 0$ je, aby $E(x)$ bolo hladké zobrazenie. Na to je postačuje, aby zobrazenie F riešenej NPC bolo hladké a aby NPC-funkcia použitá pri transformácií bola tiež hladká. Namiesto diferencovateľnej NCP-funkcie je však možné použiť hladkú aproximáciu, ako ukážeme v nasledujúcej kapitole.

Samotný algoritmus začína v počiatočnom bode x^0, pričom sa postupným zlepšovaním prepracúva čo najblzžšie k riešeniu. Postup je daný lokálnou aproximáciov zobrazenia $E(x)$ prvým rádom Taylorovho rozvoja

$$E(x^{k+1}) \approx E(x^k) + \nabla E(x^k)^T(x^{k+1} - x^k) = 0$$

odtiaľ máme iteračný predpis

$$x^{k+1} = x^k + d^k \tag{2.3}$$

pričom d^k je riešením sústavy lineárnych rovníc pre vektor d

$$\nabla E(x^k)^T d = -E(x^k) \tag{2.4}$$

Aj keď je algoritmus (2.3) funkčný a konverguje v okolí riešenia, pokiaľ je sústava (2.4) riešiteľná, nie je vo všeobecnosti zaručená cesta k riešeniu.
Na zabezpečenie konvergencie sa využívajú viaceré techniky. Jednou z nich sa budeme zaoberať v ďalšej kapitole.

V nasledujúcej kapitole popíšeme algoritmus využívajúci Newtonuvu metódu pri riešení úlohy NCP(F).

Kapitola 3

Chenova metóda

Postup opísaný v tejto kapitole sa zakladá na transformácií úlohy NCP(F) na systém nelineárnych rovníc a jeho riešením pomocou Newtonovej metódy.

Už vieme, že NCP(F) sa dá transformovať na systém nelineárnych rovníc pomocou NCP-funkcií. Použitie Newtonovej metódy na riešenie tohoto systému je v zásade možné ak sú funkcia F a aj NCP-funkcia diferencovateľné. Na druhej strane môžeme namiesto nehladkej NPC-funkcie použiť pri transformácií jej diferencovateľnú aproximáciu, čo je popísané v práci [1] pre NCP-funkciu

$$c(a, b) = a - (a - b)_+ \tag{3.1}$$

A to nás privádza k potrebe nájsť hladkú aproximáciu pre funkciu $(z)_+$.

3.1 Aproximácia funkcie $(z)_+$

V prvom rade vidíme, že pre funkciu $(z)_+$ platí

$$(z)_+ = \int_{-\infty}^{z} \sigma(y) dy \tag{3.2}$$

kde funkcia $\sigma(y)$ nazývame schodkovitou funkciu danou ako

$$\sigma(y) = \begin{cases} 1 & \text{ak } x > 0 \\ 0 & \text{ak } x \leq 0 \end{cases} \tag{3.3}$$

Schodkovitá funkcia σ sa dá rovnako ako $(z)_+$ odvodiť z

$$\sigma(y) = \int_{-\infty}^{y} \delta(x) dx \tag{3.4}$$

z čoho máme, že funkcia $(z)_+$ nie je nič iné ako dva krát zintegrovaná $\delta(x)$, čo je Diracova delta funkcia.

$$(z)_+ = \int_{-\infty}^{z} \sigma(y) dy = \int_{-\infty}^{z} \left\{ \int_{-\infty}^{y} \delta(x) dx \right\} dy \tag{3.5}$$

Funkcia $\delta(x)$ má nasledovné vlastnosti

- $\delta(x) \geq 0$
- $\int_{-\infty}^{\infty} \delta(x)dx = 1$
- $\delta(x) \geq 0$ je limita pravdepodobnostných funkcií hustôt

Následne formulujeme postup pri hľadaní aproximácie funkcie $(z)_+$.

1. Zvolíme si nejakú funkciu hustoty d(x) takú, že
$$d(x) \geq 0 \quad \int_{-\infty}^{\infty} d(x)dx = 1 \tag{3.6}$$
2. Zavedieme parametrizáciu
$$\hat{t}(x,\beta) = \frac{1}{\beta} d(\frac{x}{\beta}) \tag{3.7}$$
vďaka čomu bude $\hat{t}(x,\beta)$ hustotou pre $\beta > 0$, pričom
$$\lim_{\beta \to 0} \hat{t}(x,\beta) = \delta(x)$$
3. Nakoniec dvojnásobnou integráciov $\hat{t}(x,\beta)$ dostaneme aproximáciu funkcie $(z)_+$
$$(z)_+ \approx \hat{p}(z,\beta) = \int_{-\infty}^{z} \left\{ \int_{-\infty}^{y} \hat{t}(x,\beta)dx \right\} dy$$

Tým pádom je otázka nájdenia aproximácie vyriešená. Zostáva overiť do akej miery sa bude funkcia $\hat{p}(z,\beta)$ blýžiť k $(z)_+$. Na to máme nasledujúce tvrdenia

Veta 3.1.1. *Nech funkcia hustoty $d(x)$ spĺňa* (3.6)*, pričom je po častiach spojitá a platí*
$$\int_{-\infty}^{\infty} |x|.d(x)dx < \infty$$
a nech $\hat{t}(x,\beta)$ je daná (3.7)*. Potom pre $\hat{p}(z,\beta)$ dané*
$$\hat{p}(z,\beta) = \int_{-\infty}^{z} \left\{ \int_{-\infty}^{y} \hat{t}(x,\beta)dx \right\} dy$$
platí nasledovné

1. *$\hat{p}(z,\beta)$ je spojitá. Navyše ak je $d(x)$ k-krát spojito diferencovateľná, $\hat{p}(z,\beta)$ je k+2-krát spojito diferencovateľná.*
2. *Platí nerovnosť*
$$-D_2\beta \leq \hat{p}(z,\beta) - (z)_+ \leq D_1\beta \tag{3.8}$$
kde
$$D_1 = \int_{-\infty}^{0} |x|.d(x)dx \tag{3.9}$$
$$D_2 = \max\left\{ \int_{-\infty}^{\infty} x.d(x)dx, 0 \right\} \tag{3.10}$$

3. $\hat{p}(0,\beta) = D_1\beta$

4. $\hat{p}(z,\beta)$ *je neklesajúca a konvexná podľa z*

5. $0 \leq \frac{\partial \hat{p}(z,\beta)}{\partial z} \leq 1$

Táto veta aj s dôkazom je uvedená v práci [1].

V ďalšom budeme uvažovať už len nasledovné

$$d(x) = \frac{e^{-x}}{(1+e^{-x})} \tag{3.11}$$

Podľa (3.7) získame

$$t(x,\beta) = \frac{e^{-x}}{\beta(1+e^{-\frac{x}{\beta}})} \tag{3.12}$$

Odtiaľ máme aproximáciu $(z)_+$ $p(z,\beta)$

$$(z)_+ \approx p(z,\beta) = \int_{-\infty}^{z} \left\{ \int_{-\infty}^{y} t(x,\beta)dx \right\} dy = \int_{-\infty}^{z} \left\{ \int_{-\infty}^{y} \frac{e^{-x}}{\beta(1+e^{-\frac{x}{\beta}})} dx \right\} dy =$$

$$= \int_{-\infty}^{z} \frac{1}{1+e^{-\frac{y}{\beta}}} dy = z + \beta \ln(1+e^{-\frac{z}{\beta}})$$

Preznačíme parameter vyhladenia $\beta = \frac{1}{\alpha}$

$$(z)_+ \approx p(z,\alpha) = z + \frac{\ln(1+e^{-\alpha z})}{\alpha} \tag{3.13}$$

Pripomenieme, že apromácia $p(z,\alpha)$ je lepšia ak je parameter α väčší.
Pre apromáciu (3.13) sú v práci [1] ukázané nasledujúce vlastnosti

1. $p(z,\alpha)$ je ostro rastúca a ostro konvexná vzhľadom na z

2. $0 < \frac{\partial p(z,\beta)}{\partial z} < 1$

3. $p(z,\alpha) > x$

3.2 Transformácia NCP pomocou $p(z,\alpha)$

V druhej kapitole sme sa zaoberali transformáciov NCP na systém nelineárnych rovníc (2.2) pomocou NCP-funkcií. Teraz namiesto NCP-funkcie

$$c(a,b) = a - (a-b)_+$$

použijeme hladkú aproximáciu využívajucu (3.13) pre $z = a - b$

$$c(a,b) \approx c_\alpha(a,b) = a - (a-b) + \frac{\ln(1+e^{-\alpha(a-b)})}{\alpha} = b + \frac{\ln(1+e^{-\alpha(a-b)})}{\alpha}$$

Z toho dostaneme systém nelineárnych rovníc

$$E_\alpha(x) = \begin{pmatrix} c_\alpha(x_1, F_1(x)) \\ c_\alpha(x_2, F_2(x)) \\ \vdots \\ c_\alpha(x_n, F_n(x)) \end{pmatrix} = \begin{pmatrix} F_1(x) + \frac{\ln(1+e^{-\alpha(x_1 - F_1(x))})}{\alpha} \\ F_2(x) + \frac{\ln(1+e^{-\alpha(x_2 - F_2(x))})}{\alpha} \\ \vdots \\ F_n(x) + \frac{\ln(1+e^{-\alpha(x_n - F_n(x))})}{\alpha} \end{pmatrix} = 0_n \tag{3.14}$$

Tento systém budeme riešiť Newtonovou metódou. Keďže ide o numerické riešenie budeme odhadovať jeho presnosť pomocou funkcie kvality - konkrétne si zvolíme prirodzenú funkciu kvality

$$M(x) = \frac{1}{2} E_\alpha(x)^T E_\alpha(x) \tag{3.15}$$

Pretože riešenie systému $E_\alpha(x) = 0$ je zároveň bodom globálneho minima funkcie kvality budeme sledovať lokálnu linearizáciu - gradient funkcie kvality $\nabla M(x)$. Ak je norma gradientu $\|\nabla M(x)\|$ dosť malá, bod x je blýzko stacionárneho bodu $M(x)$ a teda aj blýzko riešenia $E_\alpha(x) = 0$.

V predchádzajúcej kapitole sme spomenuli techniky pre zabezpečenie globálnej konvergencie Newtonovej metódy. Jednou z nich je nasledujúca schéma. Pri určovaní kroku d_k budeme požadovať určité zlepšenie v hodnote funkcie kvality

$$M(x_k) - M(x_{k+1})$$

Konkrétne voľbou $x_{k+1} = x_k + \lambda_k . d_k$, kde d_k je riešenie

$$\nabla E(x^k)^T d = -E(x^k)$$

a λ_k je prvé z postupnosti 1,δ,δ^2,... ktoré spĺňa

$$M(x_k) - M(x_{k+1}) \geq \sigma \lambda_k |d_k^T \nabla M(x_k)|$$

3.3 Algoritmus Chenovej metódy

Majme úlohu NCP(F), pričom poznáme jakobiho maticu zobrazenia $F(x)$

$$\nabla F(x) = \begin{pmatrix} \frac{\partial F_1(x)}{\partial x_1} & \frac{\partial F_2(x)}{\partial x_1} & \cdots & \frac{\partial F_n(x)}{\partial x_1} \\ \frac{\partial F_1(x)}{\partial x_2} & \frac{\partial F_2(x)}{\partial x_2} & \cdots & \frac{\partial F_n(x)}{\partial x_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial F_1(x)}{\partial x_n} & \frac{\partial F_2(x)}{\partial x_n} & \cdots & \frac{\partial F_n(x)}{\partial x_n} \end{pmatrix} \tag{3.16}$$

Nech $E_\alpha(x)$ a $M_\alpha(x)$ sú definované podľa (3.14) a (3.15) potom ich jakobiho matice $\nabla E_\alpha(x)$ a $\nabla M_\alpha(x)$ sú nasledovné

$$\nabla E_\alpha(x) = \nabla F(x) + diag \begin{pmatrix} e^{-\alpha x_1 + \alpha F_1(x)} \\ e^{-\alpha x_2 + \alpha F_2(x)} \\ \vdots \\ e^{-\alpha x_n + \alpha F_n(x)} \end{pmatrix} \cdot (I - \nabla F(x)) \tag{3.17}$$

,kde $diag(v)$ je matica s vektorom v na diagonále, ostatné prvky má nulové a I je matica s jednotkami na diagonále, ostatné prvky má nulové.

$$\nabla Mx = (\nabla E_\alpha(x))^T E_\alpha(x) \tag{3.18}$$

Iteračný algoritmus

Zvolíme hodnotu $\alpha > 0$ - parametra vyhladenia, $\epsilon > 0$ - požadovaná gradientná presnosť, $0 < \delta < 1$ a $0 < \sigma < \frac{1}{2}$ - parametre voľby dĺžky kroku

Máme dané x_0, položme $k = 0$

1. Overenie gradientnej presnosti

   ak $\|\nabla M(x_k)\| < \epsilon$ stop, x_k je riešenie

   inak

2. Voľba smeru d_k. d_k je riešenie

$$\nabla E(x^k)^T d_k = -E(x^k) \tag{3.19}$$

3. Voľba dĺžky kroku λ_k

$$\lambda_k = \max(1, \delta, \delta^2, ...)$$

   také, že

$$M(x_k) - M(x_k + \lambda_k d_k) \geq \sigma \lambda_k |d_k^T \nabla M(x_k)|$$

4. Nový bod

$$x_{k+1} = x_k + \lambda_k . d_k$$
$$k = k + 1$$

   pokračuj 1.

Kapitola 4

Metóda Facchinei

Článok F. Facchineiho a J. Soaresa [5] nám ponúka nasledujúci postup pri riešení NCP(F). Najskôr si zadefinujme pojem nedegenerovaného riešenia NCP(F).

Definícia 4.0.1. *Bod $\hat{x}$ je nedegenerovaným riešením úlohy NCP(F) práve vtedy, keď $\hat{x}$ je riešením NCP(F) a naviac platí*

$$F_i(\hat{x}) = 0 \Leftrightarrow \hat{x}_i > 0 \tag{4.1}$$

4.1 Motivácia pre metódu Fachinei

Vezmime si nasledujúcu situáciu. Riešime úlohu NCP(F), ktorá má aspoň jedno nedegenerované riešenie $\hat{x}$. Potom existujú indexové množiny A a N, kde

$$A = \{i|\hat{x}_i = 0\}, \qquad N = \{i|\hat{x}_i > 0\}$$

Ak by sme tieto indexové množiny poznali, na vyriešenie NCP(F) by nám stačilo nájsť riešenie systému nelineárnych rovníc

$$F_i(\bar{x}) = 0, \qquad \forall i \in N$$

kde

$$\bar{x}_i = \begin{cases} x_i & \text{ak } i \in N \\ 0 & \text{ak } i \in A \end{cases} \tag{4.2}$$

V prípade, že je funkcia $F(x)$ diferencovateľná a poznáme jej Jakobiho maticu, môžme spomínaný systém riešiť nasledovne

Nech

$$x_N = (x_{N1}, x_{N2}, ...)^T$$

$$\bar{x} = (x_N, 0_A)$$

$$F_N(\bar{x}) = \begin{pmatrix} F_{N1}(\bar{x}) \\ F_{N2}(\bar{x}) \\ \vdots \end{pmatrix}$$

$$\nabla F_{NN}(\bar{x}) = \begin{pmatrix} \frac{\partial F_{N1}(\bar{x})}{\partial x_{N1}} & \frac{\partial F_{N2}(\bar{x})}{\partial x_{N1}} & \cdots \\ \frac{\partial F_{N1}(\bar{x})}{\partial x_{N2}} & \frac{\partial F_{N2}(\bar{x})}{\partial x_{N2}} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix} \tag{4.3}$$

kde x_i sú prvky vektora x, pričom platí $Ni, Nj \in N$ a ak $i < j$ potom $Ni < Nj$.

Zvolíme si štarovací bod x_N^0 respektíve $\bar{x}^0$ a za predpokladu, že $\nabla F_{NN}(\bar{x})$ je regulárna volíme iterácie

$$x_N^{k+1} = x_N^k + d_N^k$$

kde d_N^k je riešenie lineárneho systému

$$\nabla F_{NN}(\bar{x}^0)^T d_N^k = -F_N(\bar{x}^0)$$

čo je vlastne Newtonova metóda pre riešenie systému nelineárnych rovníc.

4.2 Fachineiho algoritmus

V predchádzajúcom sme predpokladali existenciu nedegenerovaného riešenia a znalosť indexových množín A a N pre toto riešenie, čo vo všeobecnosti nie je splnené. Z toho dôvodu pristupuje Facchinei k odhadovaniu množín A a N v každom iteračnom kroku. Konkrétne volíme

$$A^k = \{i | x_i^k \leq \varepsilon F_i(x^k)\}, \qquad N^k = \{i | x_i^k > \varepsilon F_i(x^k)\}$$

z čoho vyplýva nasledujúca stratégia. Zvolme štartovací bod x^0. Ďalšie body určujeme iterčne

$$x^{k+1} = x^k + d^k$$

kde

$$d_{A^k}^k = -x_{A^k}^k$$

kým $d_{N^k}^k$ je riešením nasledujúceho systému lineárnych rovníc

$$(\nabla F_{N^k N^K}(x^k))^T d_{N^k}^k = -F_{N^k}(x^k) + (\nabla F_{A^k N^k}(x^k))^T x_{A^k}^k$$

K algoritmu potrbujeme overenie či sme už dosť blýzko riešeniu úlohy. V predchádzajúcom sme spomínali NCP-funkcie. Jednou z nich je aj

$$\phi(a,b) = \sqrt{a^2+b^2} - (a+b)$$

Ako už bolo spomenuté pomocou NCP-funkcie a rovnicového operátora $E(x)$ môžme vytvoriť funkciu kvality

$$M(x) = \frac{1}{2}E(x)^T E(x)$$

pre ktorú platí, že má minimum s funkčnou hodnotou v bode minima

$$M(\hat{x}) = 0$$

, práve vtedy, keď je $\hat{x}$ riešením NCP(F).

Preto ako mieru blýzkosti k riešeniu volíme hodnotu spomýnanej funkcie kvality dostatočne blýzku 0.

Iteračný algoritmus

Zvolíme požadovanú presnosť riešenia ϵ, $0 < \delta < 1$ a $0 < \sigma < 1/2$ - parametre voľby dĺžky kroku, ε - parameter pri určovaní indexových množín A^k a N^k

Máme dané x^0, položme k=0

1. Overenie presnosti

   Ak $M(x^k) < \epsilon$ stop, x^k je riešenie

   inak

2. Voľba indexových množín

$$A^k = \{i | x_i^k \leq \varepsilon F_i(x^k)\}$$

$$N^k = \{i | x_i^k > \varepsilon F_i(x^k)\}$$

3. Voľba smeru

$$d_{A^k}^k = -x_{A^k}^k$$

   $d_{N^k}^k$ je riešenie lineárneho systému

$$(\nabla F_{N^k N^K}(x^k))^T d_{N^k}^k = -F_{N^k}(x^k) + (\nabla F_{A^k N^k}(x^k))^T x_{A^k}^k$$

4. Voľba dĺžky kroku λ_k

$$\lambda_k = \max(1, \delta, \delta^2, ...)$$

   také, že

$$M(x_k) - M(x_k + \lambda_k d_k) \geq \sigma \lambda_k |d_k^T \nabla M(x_k)|$$

5. Nový bod

$$x_{k+1} = x_k + \lambda_k.d_k$$
$$k = k + 1$$

pokračuj 1.

Kapitola 5

Numerický experiment

V tejto kapitole sa venujeme relizácií použitia Chenovej metódy a Facchineiho metódy v prípade konkrétneho zobrazenia. Zaujímala nás predovšetkým rýchlosť konvergencie - počet iterácií algoritmu v závislosti od rozmeru úlohy.

5.1 Vytváranie úloh

Funkciu F sme zvolili nasledovne

$$F_i(x) = x^T G_i x + h_i^T x + \gamma_i$$

kde

$$G_i = \frac{1}{n^2}\{A^T A + D\}$$

A-matica náhodných čísiel normálneho rozdelenia $N(0,1)$, D - diagonálna matica s prvkami z rovnomerného rozdelenia na $(1,2)$.
vektor h_i má ako zložky náhodné čísla z rovnomerného rozdelenia na $(0,1)$.
Prvky k_i dopočítame podľa zvoleného riešenia $\hat{x}$ nasledovne

$$\begin{array}{lllll} \text{Ak} & \hat{x} = 0 & \text{potom} & \gamma_i & = & 1 - \frac{1}{2}\hat{x}^T G_i \hat{x} - h_i^T \hat{x} \\ \text{Ak} & \hat{x} > 0 & \text{potom} & \gamma_i & = & -\frac{1}{2}\hat{x}^T G_i \hat{x} - h_i^T \hat{x} \end{array}$$

Riešenie sme volili ako vektor $\hat{x}$, ktorého prvky sú z alternatívneho rozdelenia - s 20% pravdepodobnosťou 0 a 80% pravdepodobnosťou 1. Štarcovací bod x_0 bol určený náhodne vo vzdialenosti θ od vygenerovaného riešenia.

$$x_0 = \theta \cdot \frac{\hat{x} + y}{\|\hat{x} + y\|}$$

kde y je vektor n náhodných čísiel z normálneho rozdelenia N(0,1).

5.2 Výsledky pre Chenovu metódu

Pomoucou algoritmu sme riešili série po 100 úloh s konštantnými parametrami. Výsledky v tabulkách zodpovedajú aritmetickému priemeru danej série úloh.

Zvolili sme nasledovné parametre pre Chenovu metódu

$$\epsilon = 10^{-8} \quad \alpha = 10 \quad \theta = 0,5 \quad \delta = 0,6 \quad \sigma = 0,4$$

Najskôr nás zaujímal počet iterácií podľa rozmeru úlohy.

rozmer úlohy - n	priemerný počet iterácií
5	4,9
10	5,25
20	6,5
30	8,25

Predchádzajúca štatistika sa zakladá na 100 úspešne vyriešených úlohách. Ide o to, že pri vyšších rozmeroch algoritmus zlyhával, napríklad z dôvodu zle podmienenej sústavy (3.19). Preto uvedieme koľko úloh s daným rozmerom nebolo započítaných do štatistiky z dôvodu zlyhania.

rozmer úlohy - n	priemerný počet iterácií	počet zlyhaní
5	4,9	3
10	5,25	3
15	5,25	13
20	6,5	20
30	8,25	52

Newtonova metóda konverguje v okolí riešenia. Budeme skúmať množstvo nevypočítaných úloh v závislosti od rozmeru a počiatočnej vzdialenosti od riešenia.

rozmer úlohy - n	počet zlyhaní $\theta = 0,4$	počet zlyhaní $\theta = 0,3$
5	0	0
10	0	0
15	1	0
20	3	1
30	24	0
50	88	13

Nakoniec sme testovali úlohy vačšieho rozmeru so štartom v blýzkosti riešenia $\theta = 0,1$. Okrem priemerného počtu iterácií a počtu nevypočítaných úloh sme sledovali vzdialenosť vypočítaného riešenia od určeného riešenia. Okrem priemernej vzdialenosti sme hľadali najhorší prípad, teda ten kde je táto vzdialenosť najväčšia.

Zvolili sme nasledovné parametre

$$\epsilon = 10^{-8} \quad \alpha = 10 \quad \theta = 0,1 \quad \delta = 0,6 \quad \sigma = 0,4$$

rozmer úl. - n	priem. počet iter.	počet zlyh.	primerná odch.	max. odch.
50	3,01	0	$2,4 * 10^{-4}$	$1,8 * 10^{-3}$
70	3,07	0	$2,4 * 10^{-3}$	0,2138
100	3,15	0	$4,3 * 10^{-3}$	0,3647
150	3,31	1	$1,1 * 10^{-3}$	$2,2 * 10^{-2}$
200	3,64	1	$1,8 * 10^{-3}$	$5,3 * 10^{-2}$

Zistili sme, že vzdialenosť $\theta = 0,1$ bola v tomto prípade bezpečná a nevypočítané úlohy neboli žiadne, respektíve ich počet bol zanedbateľný aj pri väčšćh rozmeroch riešených úloh.
V priemere konvergoval algoritmus k zvolenému riešeniu, našli sa však prípady kedy bola vzdialenosť značne veľká. Treba však pripomenúť, že riešená úloha mohla mať viac ako jedno riešenie, tým pádom algoritmus skonvergoval k inému riešeniu.

5.3 Výsledky pre Facchineiho metódu

Pomoucou algoritmu sme riešili série po 100 úloh s konštantnými parametrami. Výsledky v tabuľkách zodpovedajú aritmetickému priemeru danej série úloh.

Zvolili sme nasledovné parametre pre Facchineiho metódu

$$\epsilon = 10^{-4} \quad \varepsilon = 10^{-2} \quad \theta = 0,5 \quad \delta = 0,6 \quad \sigma = 0,4$$

Najskôr nás zaujímal počet iterácií podľa rozmeru úlohy a počet zlyhaní.

rozmer úlohy - n	priemerný počet iterácií	počet zlyhaní
5	13,2	12
10	20,3	24
20	78,6	48

V porovnaní s Chenovou metódou sme teda dosiahli o dosť horšie výsledky.

5.4 Degenerované úlohy

V predchádzajúcich experimentoch sme volili úlohy tak, že vždy existovalo nedegenerované riešenie. Preto sme urobili zmenu pri tvorbe riešenej NCP(F) úlohy a to nasledovne

$$\begin{array}{llll} \text{Ak} \quad \hat{x} = 0 & \text{potom} \quad \gamma_i & = & -\frac{1}{2}\hat{x}^T G_i \hat{x} - h_i^T \hat{x} \\ \text{Ak} \quad \hat{x} > 0 & \text{potom} \quad \gamma_i & = & -\frac{1}{2}\hat{x}^T G_i \hat{x} - h_i^T \hat{x} \end{array}$$

z čoho vyplýva, že takto vytvorené úlohy budú mať degenerované riešenia, pokiaľ bude optimálne riešenie nadobúdať v nejakej zložke nulovú hodnotu.

Zvolili sme nasledujúce parametre

rozmer úlohy n=10

Pre Chenovu metódu

$$\epsilon = 10^{-8} \quad \alpha = 10 \quad \theta = 1 \quad \delta = 0,6 \quad \sigma = 0,4$$

Pre Facchineiho metódu

$$\epsilon = 10^{-4} \quad \varepsilon = 10^{-2} \quad \theta = 0,5 \quad \delta = 0,6 \quad \sigma = 0,4$$

Za degenerované riešenie sme považovali také, kde $\hat{x}_i$ aj $F_i(\hat{x})$ nadobúda hodnotu menšiu ako 10^{-4}

úlohy	Chenova meóda	Facchineiho metóda
vypočitane	68	86
degenerované	0	82

Chenova metóda teda riešila aj takto vytvorené úlohy, ale nikdy sa nedopracovala k degenerovanému riešeniu. Na druhej strane Fachineiho metóda sa takmer vždy dopracovala k riešeniu, ktoré bolo aj degenerované.

Záver

V teoretickej časti práce sme uviedli klasifikáciu úloh o komplementarite a základné tvrdenia ohladne existencie a jednoznačnosti riešenia nelineárnej úlohy o komplementarite. Popísali sme postup transformácie tejto úlohy na problém hľadania pevného bodu, minimalizačnú úlohu a riešenie systému nelineárnych rovníc. Ďalej bola predstavená Chenova metóda s algoritmom pre riešenie diferencovateľných NCP(F).

Algoritmus Chenovej metódy sme naprogramovali v programe MATLAB. Na zvolenom zobrazení sme demonštrovali konvergenciu v závislosti od rozmeru úlohy. Ukázalo sa, že metóda v našom prípade zlyhávala a to v závislosti od rozmeru úlohy ako aj vzdialenosti štartovacieho bodu od riešenia.

Pri vačších rozmeroch sme sledovali konvergenciu k zvolenému riešeniu. Ojedinelé prípady, kedy algoritmus skonveroval k vzdialenému bodu odôvodňujeme existenciov viacerých riešení.

Rovnako sme naprogramovali aj Facchineiho metódu, ktorá však nepreukázala podobnú úspešnosť pri riešení ako Chenova metóda. Na druhej strane sme preukázali, že táto metóda je schopná odhalovať degenerované riešenia, čo sa pri Chenovej metóde nepodarilo.

Literatúra

[1] Chunhui Chen:*Smoothing Methods in Mathematical Programming*,Disertačná práca - PhD,University of Wisconsin - Madison,1995

[2] Stephen C. Billups:*Algorithms for Complementarity Problems and Generalized Equations*,Disertačná práca - PhD,University of Wisconsin - Madison,1995

[3] Michael C. Ferris,Christian Kanzow, *Complementarity and Related Problems:Survey*,Mathematical Programming Technical Report 98-17, November 1998,1998

[4] Boris Balko,*Metódy riešenia nelineárnej úlohy o komplementarite*,Diplomová práca,FMFI UK Bratislava,2005

[5] Francisco Facchinei,João Soares,*A New Merit Function for Nonlinear Complementarity Problems and Related Algorithm*,1997

[6] Stephen J. Wright,Christina Oberlin,*An Accelerated Newton Method for Equations with Semismooth Jacobians and Nonlinear Complementarity Problems*,Optimization Technical Report 06-02, April 2006

[7] Francisco Facchinei, Jong Shi Pang, *Finite-Dimensional Variatonal Inequalities and Complementarity Problems*,Springer-Verlag New York, Inc.,2003

Kapitola 6

Príloha

chen.m

```
    function hod=chen()

    n=5; alfa=10; for i=1:n
G(:,:,i)=matG(n);
h(:,i)=vekh(n);
end xr=xries(n);

    Fx=F(xr,n,G,h,xr);
operator=E(xr,n,G,h,xr,alfa);
df=dF(xr,n,G,h,xr);
de=dE(xr,n,G,h,xr,alfa);
dM(xr,n,G,h,xr,alfa);

    y=randn(n,1);
xit=xr+0.1*(xr-y)/(norm(xr-y));
k=0;
zl=0;
ries=0;
for i=1:50
k=k+1;
vlc=abs(eig(dE(xit,n,G,h,xr,alfa)));
if (min(vlc)/max(vlc))¡0.000001
disp('zlyhanie')
zl=1;
break
else
dit=dE(xit,n,G,h,xr,alfa)E(xit,n,G,h,xr,alfa);
if dit¡inf
for i=1:30
lam=0.6ˆ(i-1);
xnit=xit+lam*dit;
if M(xit,n,G,h,xr,alfa)-M(xnit,n,G,h,xr,alfa)¿0.4*lam*abs(dit'*dM(xit,n,G,h,xr,alfa))
break
end
end
```

```
else
disp('neznama chyba')
zl=1;
break
end
xit=xnit;
if rem(i,10)==0
eps=norm(dM(xit,n,G,h,xr,alfa))
end
end
if norm(dM(xit,n,G,h,xr,alfa))¡10ˆ(-8)
disp('riesenie')
ries=1;
break
end
end

if norm(dM(xit,n,G,h,xr,alfa))¿=10ˆ(-8)
disp('zacyklenie')
zl=1;
end eps=norm(dM(xit,n,G,h,xr,alfa));

Fx=F(xit,n,G,h,xr);
for i=1:n kompl(i)=xit(i)*Fx(i);
end

chyba1=norm(xit-xr);
chyba2=max(abs(xit.*F(xit,n,G,h,xr)));
chyba3=min(xit);
chyba4=min(F(xit,n,G,h,xr));
hod=[k;chyba1;chyba2;chyba3;chyba4;zl;ries];
```

F.m

```
function hod=F(x,n,G,h,xr)

for i=1:n
if xr(i)==0
k(i)=1-(0.5*xr'*G(:,:,i)*xr)-h(:,i)'*xr;
else
k(i)=-(0.5*xr'*G(:,:,i)*xr)-h(:,i)'*xr;
end
F(i)=k(i)+(0.5*x'*G(:,:,i)*x)+h(:,i)'*x;
end

hod=F';
```

E.m

```
function hod=E(x,n,G,h,xr,alfa)

f=F(x,n,G,h,xr); for i=1:n
```

```
z=x(i)-f(i);
hod(i)=x(i)-p(z,alfa);
end
hod=hod';
```

M.m

```
function hod=M(x,n,G,h,xr,alfa)

hod=0.5*E(x,n,G,h,xr,alfa)'*E(x,n,G,h,xr,alfa);
```

matG.m

```
function hod=matG(n)

A=randn(n); D=diag(rand(n,1))+eye(n); hod=(nˆ2)*(A'*A+D);
```

vekh.m

```
function hod=vekh(n)

hod=rand(n,1);
```

dF.m

```
function hod=dF(x,n,G,h,xr)

for i=1:n
hod(:,i)=G(:,:,i)*x+h(:,i);
end

hod=hod';
```

dE.m

```
function hod=dE(x,n,G,h,xr,alfa)

f=F(x,n,G,h,xr); df=dF(x,n,G,h,xr);

for i=1:n
ex=exp(-alfa*x(i)+alfa*f(i));
if ex¡inf
ee(i)=ex/(1+ex);
else
ee(i)=1;
end
end die=diag(ee);
hod=df+die*(eye(n)-df);
```

dM.m

```
function hod=dM(x,n,G,h,xr,alfa)

hod=dE(x,n,G,h,xr,alfa)'*E(x,n,G,h,xr,alfa);
```

p.m

```
function hod=p(z,alfa)
hod=z+log(1+exp(-alfa*z))/alfa;
```

facch.m

```
function hod=chen()

n=10
for i=1:n
G(:,:,i)=matG(n);
h(:,i)=vekh(n);
end

xr=xries(n);

Fx=F(xr,n,G,h,xr);
df=dF(xr,n,G,h,xr);

y=randn(n,1);
xit=xr+1*(xr-y)/(norm(xr-y));
k=0;

zl=0;
ries=0;
ep=0.01;

for i=1:1000
k=k+1;
m=0;
xA=0;
FN=0;
dFNN=0;
dFAN=0;
for i=1:n
if xit(i)¡ep*F(xit,n,G,h,xr)
m=m+1;
A(m)=i;
else
N(i-m)=i;
end
end

if m¿0
```

```
for i=1:m
d(A(i))=-xit(A(i));
end

for i=1:m
xA(i,1)=-xit(A(i));
end

end

fF=F(xit,n,G,h,xr);
if n-m¿0
for i=1:n-m
FN(i,1)=fF(N(i));
end

fdF=dF(xit,n,G,h,xr);
for i=1:n-m
for j=1:n-m
dFNN(i,j)=fdF(N(i),N(j));
end
end
end

if m¿0 & n-m¿0
for i=1:n-m
for j=1:m
dFAN(i,j)=fdF(N(i),A(j));
end
end
end

if n-m¿0 vlc=abs(eig(dFNN)); if (min(vlc)/max(vlc))¡0.000001
disp('zlyhanie podm')
zl=1;
break
else
if m¿0
dN=dFNN (-FN+dFAN*xA);

else

dN=dFNNFN;
end

for i=1:n-m
d(N(i))=dN(i);
end

for i=1:10
lam=0.6ˆ(i-1);
xnit=xit+lam*d';
```

```
if
    M(xit,n,G,h,xr)-M(xnit,n,G,h,xr)¿0.4*lam*abs(d'*M(xit,n,G,h,xr))
break
end

    end

    xit=xnit;

    end
end
if M(xit,n,G,h,xr)¡10ˆ(-6)
disp('riesenie')
ries=1;
break
end
end

    if M(xit,n,G,h,xr)¿=10ˆ(-6)
disp('zacyklenie')
zl=1;
end

    Fx=F(xit,n,G,h,xr); for i=1:n kompl(i)=xit(i)*Fx(i);
end

    chyba1=norm(xit-xr);
chyba2=max(abs(xit.*F(xit,n,G,h,xr)));
chyba3=min(xit);
chyba4=min(F(xit,n,G,h,xr));
hod=[k;chyba1;chyba2;chyba3;chyba4;zl;ries];
```