NUMERIKUS EREDMÉNYEK AZ ÁLTALÁNOS LINEÁRIS KOMPLEMENTARITÁSI FELADATRA VONATKOZÓAN

NUMERICAL RESULTS FOR THE GENERAL LINEAR COMPLEMENTARITY PROBLEM

Darvay Zsolt,¹ Füstös Ágnes²

Babeş-Bolyai Tudományegyetem, Matematika és Informatika Kar, Kolozsvár, Románia

¹ darvay@cs.ubbcluj.ro

² fustosagi@yahoo.com

Abstract

In this article we discuss the general interior-point algorithm for linear complementarity problems (LCP) introduced by Tibor Illés, Marianna Nagy and Tamás Terlaky. Moreover, we present a various set of numerical results with the help of a code implemented in the C++ programming language. These results support the efficiency of the algorithm for both monotone and sufficient LCPs.

Keywords: interior-point algorithm, general linear complementarity problem, numerical results, sufficient matrix, object-oriented programming.

Összefoglalás

Az alábbi cikkben az Illés Tibor, Nagy Marianna és Terlaky Tamás által bevezetett általános lineáris komplementaritási feladatra vonatkozó belsőpontos algoritmust tárgyaljuk és C++ programozási nyelvben megvalósított kód segítségével különböző numerikus eredményeket mutatunk be. Ezen eredmények alátámasztják az algoritmus hatékonyságát mind monoton, mind elégséges lineáris komplementaritási feladatok esetén.

Kulcsszavak: belsőpontos algoritmus, általános lineáris komplementaritási feladat, numerikus eredmények, elégséges mátrix, objektumorientált programozás.

1. Bevezető

Lineáris komplementaritási feladatokkal (LCP) különböző gyakorlati problémák megoldása során találkozhatunk. A feladatot egy M mátrix határozza meg, amellyel egy lineáris összefüggést írunk le, de ezen kívül egy komplementaritási feltételnek is teljesülnie kell.

Az LCP $\mathbb{NP}$ -teljes feladat, ezért az M -re vonatkozó megszorítások nélkül nem találunk hatékony megoldást. Léteznek algoritmusok, amelyek polinomiális időn belül oldják meg az LCP-t, ha az M mátrix pozitív szemidefinit vagy $P_*(\kappa)$ tulajdonságú (habár ez esetben a bonyolultság függ κ -tól is), azonban a $P_*(\kappa)$ tulajdonság ellenőrzésére nem született polinomiális időben működő algoritmus.

Illés Tibor, Nagy Marianna és Terlaky Tamás [1, 2, 3] egy olyan, módszert adtak meg, melynek segítségével módosíthatóak a belsőpontos algo-

ritmusok úgy, hogy az általános lineáris komplementaritási feladat is megoldható polinomiális időben. Ez azt jelenti, hogy meghatározunk egy közelítő megoldást, vagy levonjuk a következtést, hogy nem teljesül a $P_*(\kappa)$ tulajdonság.

Az algoritmust C++ programozási nyelvben implementáltuk.

2. A feladat leírása

Az LCP esetén az $x, s \in \mathbb{R}^n$ vektorokat keressük, amelyekre teljesül:

$$\begin{cases} Mx + q = s, \\ xs = 0, \\ x \geq 0, s \geq 0, \end{cases} \quad (1)$$

ahol $q \in \mathbb{R}^n$, $M \in \mathbb{R}^{n \times n}$ és xs az x és s vektorok elemenkénti szorzatát jelöli.

3. A $P_*(\kappa)$ tulajdonság

A $P_*(\kappa)$ mátrioxosztályt legelőször Kojima és társai [4] vezették be a pozitív szemidefinit mátrixok egy általánosításaként.

Egy $M \in \mathbb{R}^{n \times n}$ mátrix $P_*(\kappa)$ tulajdonságú, ha bármely $x \in \mathbb{R}^n$ esetén teljesül az alábbi feltétel:

$$(1 + 4\kappa) \sum_{i \in \mathcal{I}_+(x)} x_i (Mx)_i + \sum_{i \in \mathcal{I}_-(x)} x_i (Mx)_i \geq 0. \quad (2)$$

Egy $M \in \mathbb{R}^{n \times n}$ mátrix P_* tulajdonságú, ha $P_*(\kappa)$ tulajdonságú valamely pozitív κ -ra vagyis:

$$P_* = \bigcup_{\kappa > 0} P_*(\kappa).$$

4. A P_0 tulajdonság

Egy $M \in \mathbb{R}^{n \times n}$ mátrix P_0 tulajdonságú, ha egyetlen főminorja sem negatív.

Egy $M \in \mathbb{R}^{n \times n}$ mátrix akkor és csakis akkor P_0 tulajdonságú, ha az

$$M' = \begin{bmatrix} -M & I \\ S & X \end{bmatrix}$$

mátrix nem szinguláris, bármely $X, S \in \mathbb{R}^{n \times n}$ pozitív diagonális mátrixok esetén.

5. Az algoritmushoz használt egyéb összefüggések

Az algoritmus a centrális trajektóriát követi és minden egyes iterációban egy Newton-lépés segítségével határozzuk meg a következő pontot.

A Newton-lépést meghatározó egyenletrendszer a következő:

$$\begin{aligned} -M\Delta x + \Delta s &= 0, \\ s\Delta x + x\Delta s &= a. \end{aligned} \quad (3)$$

Ha az M egy P_0 mátrix, akkor az előbbi rendszernek egyértelmű megoldása van és az a hosszúságú Newton-lépést az

$$x(a) = x + a\Delta x, \quad s(a) = s + a\Delta s \quad (4)$$

összefüggések adják meg.

A centrális úttól való távolságot az alábbi δ_c -vel jelölt centralitási mérték segítségével határozzuk meg:

$$\delta_c(xs, \mu) = \left\| \sqrt{\frac{xs}{\mu}} - \sqrt{\frac{\mu}{xs}} \right\| \quad (5)$$

Ha a centralitási mérték túllép egy előre megadott határt ($\delta_c(xs, \mu) < \tau$), akkor a centrális út megfelelő közelségében vagyunk. Ez esetben csökkentjük a μ -t.

A lokális κ kiszámításához minden lépésben a következő közelítő függvényt használjuk:

$$\kappa(\Delta x) = -\frac{1}{4} \frac{x^T M x}{\sum_{i \in \mathcal{I}_+(x)} x_i (Mx)_i}. \quad (6)$$

Ha egy belső iteráció során a közelség csökkenése nem elégséges, vagyis

$$\delta_c^2(xs, \mu) - \delta_c^2(x(\bar{\alpha})s(\bar{\alpha}), \mu) < \frac{5}{3(1 + 4\kappa)}, \quad (7)$$

akkor az LCP mátrixa nem $P_*(\kappa)$ tulajdonságú a lokális κ -ra, ezért kiszámítjuk κ -t az új Δx -re [1].

6. Leállási feltételek

Az algoritmus során több különböző feltételt tanulmányozunk. Amennyiben ezek közül valamelyik teljesül, akkor különböző következtetések vonhatunk le. Az alábbi feltételeket vizsgáljuk:

1. Az x és s elemenkénti szorzatösszege (komplementaritási rés) elér egy előre definiált hibaküszöböt: $x^T s < \epsilon$.
2. Az új κ nem meghatározott, vagyis nincs egyetlen pozitív $\Delta x_i, \Delta s_i$ sem a (2) összefüggés kiszámításakor. Ebben az esetben a mátrix nem P_* tulajdonságú.
3. Az új κ túllépte az előre megadott $\tilde{\kappa}$ felső korlátot: $\kappa(\Delta x) > \tilde{\kappa}$. Ebben az esetben M nem $P_*(\tilde{\kappa})$ mátrix.

7. Az algoritmus

Az [1]-beli általános LCP-re vonatkozó algoritmust a következőképpen adhatjuk meg az implementáció szemszögéből nézve:

Bemenet: $\tilde{\kappa} > 0$ a κ felső határa; $\tau \geq 2$ közelségi mértékre vonatkozó paraméter; $\epsilon > 0$ pontossági paraméter; $0 < \sigma < 1$ léptetési (barrier) paraméter; (x_0, s_0) kezdőpont, $\mu_0 > 0$ ü.h. $\delta_c(x_0 s_0, \mu) < \tau$.

Kimenet: (x, s) az LCP feladat megoldása vagy egy üzenet arra vonatkozóan, hogy nem teljesül a $P_*(\kappa)$ tulajdonság.

begin

$x := x_0; s := s_0; \mu := \mu_0; \kappa := 0;$

while $x^T s > \epsilon$ **do begin**

$\mu = \sigma (x^T s) / n;$

while $\delta_c(xs, \mu) \geq \tau$ **do begin**

számítsuk ki a $(\Delta x, \Delta s)$ -t, ha $a = \mu e - xs;$

if M szinguláris **then**

return a mátrix nem P_0 tulajdonságú;

end if

```

 $\bar{\alpha} = \text{maximalisLepes}(x, s, \mu);$ 
if  $\delta_c^2(xs, \mu) - \delta_c^2(x(\bar{\alpha})s(\bar{\alpha}), \mu) < 5/(3(1+4\kappa))$  then
     $\kappa(\Delta x)$  kiszámítása;
    if  $\kappa(\Delta x)$  nem létezik then
        return a mátrix nem  $P_*$  tulajdonságú;
    end if
    if  $(\kappa(\Delta x) > \tilde{\kappa})$  then
        return a mátrix nem  $P_*(\tilde{\kappa})$  tulajdonságú;
    end if
     $\kappa = \kappa(\Delta x);$ 
end if
 $x = x(\bar{\alpha}); s = s(\bar{\alpha});$ 
end
end.

```

A maximális lépéshossz kiszámítására vonatkozó függvényt a következőképpen adjuk meg:

function maximalisLepes(x, s, μ):

$$\alpha_1 = \min \left\{ -\frac{x_i}{\Delta x_i} \mid \Delta x_i < 0 \right\};$$

$$\alpha_2 = \min \left\{ -\frac{s_i}{\Delta s_i} \mid \Delta s_i < 0 \right\};$$

$$\alpha = \min\{\alpha_1, \alpha_2\};$$

$\text{nr_it} = 0;$

while $(\delta_c(x(\alpha/2)s(\alpha/2), \mu) < \delta_c(x(\alpha)s(\alpha), \mu)$ **and**
 $\text{nr_it} < 10)$ **do begin**

$\alpha = \alpha/2;$

$\text{nr_it} = \text{nr_it} + 1;$

end

return $\alpha;$

end

Az algoritmus implementálását C++ programozási nyelvben valósítottuk meg, felhasználva a Visual Studio integrált fejlesztői környezetet, kiindulva az [5] publikációban bevezetett kódból.

8. Numerikus eredmények

8.1. A konvergencia „gyorsaságának” változása a σ függvényében

Az algoritmus implementációja során a μ kiszámítása minden iterációban a $\mu = \sigma(x^T s) / n$ összefüggés segítségével történik. Az 1. táblázat azt tükrözi, hogy a σ megválasztása hogyan befolyásolja az iterációs számot. A tanulmányozott feladatok a [6] oldalon közzétett elégséges LCP-kre vonatkoznak. A [7] publikációban a szerzők elsőként mutattak be numerikus eredményeket a fenti elégséges LCP-kre. A táblázatban a következő jelöléseket használtuk: M = mátrix, σ = szigma, K = külső iterációk száma, ΣB = belső iterációk összege, $B\bar{A}$ = belső iterációk átlaga.

1. táblázat. Eredmények elégséges LCP-re

M	σ	K	ΣB	B $\bar{A}$
HEF_10_01	0,15	7	24	3,429
	0,4	13	26	2
	0,6	23	46	2
	0,9	115	115	1
HEF_10_02	0,15	7	19	2,714
	0,4	13	26	2
	0,6	23	46	2
	0,9	115	115	1
HEF_20_01	0,15	7	23	3,286
	0,4	13	28	2,154
	0,6	23	46	2
	0,9	115	115	1
HEF_20_02	0,15	7	18	2,571
	0,4	13	27	2,077
	0,6	23	46	2
	0,9	115	115	1

2. táblázat. Eredmények monoton LCP-re

σ	K	ΣB	B $\bar{A}$
0,15	7	28	4
0,4	13	31	2,384
0,6	24	31	1,291
0,9	114	118	1,035

A továbbiakban a [8] cikkben található alábbi monoton LCP-re mutatunk be eredményeket:

$$M = \begin{pmatrix} 2 & 1 & 1 & 1 \\ 1 & 2 & 0 & 1 \\ 1 & 0 & 1 & 2 \\ -1 & -1 & -2 & 0 \end{pmatrix}, \quad q = \begin{bmatrix} -8 \\ -6 \\ -4 \\ 3 \end{bmatrix}$$

Az eredményeket a 2. táblázatban összegeztük. Megfigyelhetjük, hogy a σ paraméter csökkenésével a teljes, illetve a külső iterációs szám csökken, azonban a belső iterációk száma és átlaga növekszik.

A továbbiakban a κ paraméterre vonatkozó alsó korlátot vizsgáljuk.

8.2. A κ paraméter alsó korlátjának exponenciális növekedése

Az alábbi mátrix tanulmányozását Csizmadia Zsolt javasolta és a [9] cikkben a szerzők igazolták, hogy a mátrix elégséges, de a neki megfelelő κ paraméter a méret függvényében exponenciálisan megnövekedhet.

$$M = \begin{pmatrix} 1 & 0 & 0 & \dots & \dots & 0 \\ -1 & 1 & 0 & \dots & \dots & 0 \\ -1 & -1 & 1 & \dots & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & & \vdots \\ -1 & -1 & -1 & \dots & -1 & 1 \end{pmatrix}$$

Egy olyan LCP-t tanulmányoztunk, amelyre $q = [0 \ 1 \ \dots \ n-1]^T$.

Az elégséges mátrixok esetén általában nem tudjuk pontosan meghatározni a κ paraméter értékét, de alsó korlátot tudunk biztosítani erre vonatkozóan. Az algoritmus során a (6) képlet alapján számoljuk ki az adott lépésben a lokális κ értéket. Ezeknek a maximuma határoz meg egy alsó korlátot a κ -ra.

A kapott eredményeket a 3. táblázat mutatja be.

3. táblázat. A κ változása a méret függvényében

n	σ	κ
5	0,8	2,70815
10	0,8	347,535
20	0,8	1261800

9. Következtetések

Az Illés, Nagy és Terlaky által bevezetett általános LCP-re mutattunk be numerikus eredményeket. Az implementáció során a maximális lépéshossz meghatározására vonatkozóan egy sajátos módszert adtunk meg. Az algoritmus hatékonyan működött elégséges, illetve monoton LCP-re is. A Csizmadia Zsolt által bevezetett mátrix esetén vizsgáltuk a κ paraméter változását a méret függvényében.

Köszönetnyilvánítás

A szerzők köszönetüket fejezik ki az Erdélyi Múzeum-Egyesületnek a kutatási munkához nyújtott támogatásért.

Szakirodalmi hivatkozások

- [1] Illés T., Nagy M., Terlaky T.: A polynomial path-following interior point algorithm for general linear complementarity problems. *Journal of Global Optimization* 47/3. (2010) 329–342.
<https://doi.org/10.1007/s10898-008-9348-0>
- [2] Illés T., Nagy M., Terlaky T.: EP Theorem for Dual Linear Complementarity Problems. *Journal of Optimization Theory and Applications* 140/2. (2009) 233–238.
<https://doi.org/10.1007/s10957-008-9440-0>
- [3] Illés T., Nagy M., Terlaky T.: Polynomial Interior Point Algorithms for General Linear Complementarity Problems. *Algorithmic Operations Research*, 5. (2010) 1–12.
- [4] Kojima M., Megiddo N., Noma T., Yoshise A.: A Unified Approach to Interior Point Algorithms for Linear Complementarity Problems. *Lecture Notes in Computer Science*, 538, Springer Verlag, Berlin, Germany. (1991)
- [5] Darvay Zs., Takó I.: Computational comparison of primal-dual algorithms based on a new software. unpublished manuscript. (2012)
- [6] Morapitiye S.: Sufficient Matrices. (letöltve 2019. február 12.).
<http://math.bme.hu/~sunil/su-matrices/>
- [7] Darvay Zs., Illés T., Povh J., Rigó P. R.: Predictor-corrector interior-point algorithm for sufficient linear complementarity problems based on a new search direction. manuscript. (2019)
- [8] Hock W., Shittkowski K.: Test Examples for Non-linear Programming Codes. *Lecture Notes in Economics and Mathematical Systems* 187. Springer, Berlin (1981)
<https://doi.org/10.1007/978-3-642-48320-2>
- [9] de Klerk E., E.-Nagy M.: On the complexity of computing the handicap of a sufficient matrix. *Mathematical Programming Serie A* 129. (2011) 383–402.
<https://doi.org/10.1007/s10107-011-0465-z>