sciendo

Műszaki Tudományos Közlemények vol. 15. (2021) 15–18.
DOI: Magyar: https://doi.org/10.33895/mtk-2021.15.04
Angol: https://doi.org/10.33894/mtk-2021.15.04

EME MTSZ MTK

Súlyozott lineáris komplementaritási feladatok teljes Newton-lépéses algoritmusának implementációja

Implementation of the Full-Newton Step Algorithm for Weighted Linear Complementarity Problems

Darvay Zsolt,[1] Orbán Attila-Szabolcs[2]

[1] *Babeș–Bolyai Tudományegyetem, Matematika és Informatika Kar, Kolozsvár, Románia, darvay@cs.ubbcluj.ro; Erdélyi Múzeum Egyesület, Matematikai és Informatikai Szakosztály*

[2] *Babeș–Bolyai Tudományegyetem, Matematika és Informatika Kar, Kolozsvár, Románia, orban.attila@yahoo.com*

Abstract

We present a path-following interior-point algorithm for solving the weighted linear complementarity problem from the implementation point of view. We studied two variants, which differ only in the method of updating the parameter which characterizes the central path. The implementation was done in the C++ programming language and the obtained numerical results prove the efficiency of the proposed method.

Keywords: *weighted linear complementarity problem, path-following interior-point algorithm, full-Newton step method.*

Összefoglalás

A súlyozott lineáris komplementaritási feladatot megoldó útkövető belsőpontos algoritmust az implementáció szemszögéből nézve mutatjuk be. Két változatot vizsgáltunk, amelyek csak a centrális út pontjait jellemző paraméter változtatási módjában térnek el egymástól. A megvalósítás C++ programozási nyelvben történt, és a kapott numerikus eredmények igazolják az általunk javasolt módszerek hatékonyságát.

Kulcsszavak: *súlyozott lineáris komplementaritási feladat, útkövető belsőpontos algoritmus, teljes Newton-lépéses algoritmus.*

1. Bevezetés

Sok optimalizálási feladat adható meg lineáris komplementaritási problémaként (*linear complementarity problem*), amelyet az angol elnevezésnek megfelelően LCP-vel jelölhetünk [1, 2]. Bebizonyosodott az is, hogy az LCP különböző gyakorlati feladatok, például mérnöki vagy gazdasági területről származó kérdések megoldására alkalmas.

Az LCP-k egy olyan kiterjesztett változatával foglalkozunk, amely az eddigieken kívül további gyakorlati alkalmazásokkal rendelkezik [3–5]. Ez az úgynevezett súlyozott lineáris komplementaritási feladat (WLCP – *weighted linear complementarity problem*), amelyet 2012-ben Florian Potra vezetett be [6].

A továbbiakban bemutatjuk a feladatot.

2. A feladat leírása

A WLCP alakja:

$$\begin{cases} xs = w, \\ -Mx + s = q, \\ x \geq 0, s \geq 0, \end{cases}$$

ahol $q \in \mathbb{R}^n, M \in \mathbb{R}^{n\times n}$, és a súlyvektor pedig $w \in \mathbb{R}^n_+ = \{x \in \mathbb{R}^n | x \geq 0\}$. Továbbá feltételezzük, hogy M egy monoton mátrix, tehát $x^T Mx \geq 0$ bármely $x \in \mathbb{R}^n$ esetén. Az M mátrix és a q vektor adottak, valamint az x és az s ismeretlenek.

A belsőpontos algoritmusok (IPMs), főképp a primál-duál belsőpontos algoritmusok, igen hatékony módszernek bizonyultak az LCP-k megoldására. Ezen algoritmusok esetén feltételezzük, hogy a WLCP megengedett megoldásainak $\mathcal{F}$-el jelölt halmaza nem üres:

$\mathcal{F} := \{(x, s) \in \mathbb{R}^n_+ \times \mathbb{R}^n_+ : -Mx + s = q\}$.

A WLCP optimális megoldásainak halmaza olyan megengedett (x, s) párokból áll, amelyekre a komplementaritási feltétel is teljesül. Az így kapott halmazt $\mathcal{F}^*$-al jelöljük:

$\mathcal{F}^* := \{(x, s) \in \mathcal{F} : xs = w\}$.

Végül, amennyiben x és s pozitív vektorok, szigorúan megengedett megoldáshalmazról beszélünk, amelyet $\mathcal{F}^0$-al jelölünk:

$\mathcal{F}^0 := \{(x, s) \in \mathcal{F} : x, s > 0\}$.

A továbbiakban feltételezzük, hogy adott egy szigorúan megengedett $(x^0, s^0) \in \mathcal{F}^0$ kiindulópont, melyhez hozzárendeljük a c vektort, illetve a μ^0 állandót:

$$c = x^0 s^0, \qquad \mu^0 = \frac{(x^0)^T s^0}{n}.$$

A μ^0-ból kiindulva az algoritmus rendre különböző μ értékeket határoz meg, amelyek egy ideális útvonalat, az úgynevezett centrális utat jellemzik. A hagyományos LCP-k esetén a centrális utat az alábbi rendszer segítségével határozzuk meg:

$$\begin{cases} xs = \mu e, \\ -Mx + s = q, \\ x > 0, s > 0, \end{cases}$$

ahol e az egyesekből álló n dimenziós vektor.

A WLCP esetén a nemlineáris összefüggés ($xs = \mu e$) jobb oldali vektorát az alábbi kifejezéssel helyettesítjük [6, 7]:

$$w(\mu) = \left(1 - \frac{\mu}{\mu^0}\right) w + \frac{\mu}{\mu^0} c, \qquad \mu \in [0, \mu^0]. \quad (1)$$

Amennyiben a $w = 0$ és $x^0 = s^0 = e$, akkor visszakapjuk a hagyományos esetet, tehát $w(\mu) = \mu e$ lesz. A $w(\mu)$ megválasztását az is indokolja, hogy $\mu = \mu^0$ esetén a c vektort, illetve $\mu = 0$ esetén a w súlyvektort kapjuk vissza.

A továbbiakban azt feltételezzük, hogy a $w(\mu)$ értékét az (1) összefüggés határozza meg. Ezért ebben az esetben a centrális utat a következő rendszer írja le:

$$\begin{cases} xs = w(\mu), \\ -Mx + s = q, \\ x > 0, s > 0. \end{cases}$$

A nemlineáris összefüggés miatt a centrális út pontjait nem fogjuk tudni pontosan meghatározni, de ennek a követését a Newton-módszer segítségével valósíthatjuk meg:

$$\begin{cases} s\Delta x + x\Delta s = w(\mu) - xs, \\ -M\Delta x + \Delta s = 0. \end{cases}$$

A Δx és Δs vektorok úgynevezett keresési irányokat adnak meg, amelyek lehetőséget teremtenek a következő iteráció x és s vektorainak a meghatározására. A teljes Newton-lépéses algoritmusok esetén az új pontokat az alábbi összefüggések adják meg:

$$x^+ = x + \Delta x, \qquad s^+ = s + \Delta s.$$

Egy adott pontnak a centrális úttól való távolságát, az alábbi közelségi mérték (*proximity measure*) segítségével becsülhetjük meg:

$$\delta(x, s; \mu) = \left\| \frac{w(\mu) - xs}{\mu} \right\|.$$

Fontos kiemelni, hogy az X és S az x, illetve s komponenseiből alkotott diagonálmátrix:

$X = diag(x)$, $S = diag(s)$.

A [7] cikkben bevezetett teljes Newton-lépéses algoritmust a továbbiakban az implementáció szemszögéből nézve változtatjuk meg.

3. Az algoritmus implementálása

Az implementáció szemszögéből nézve az algoritmusnak két módosított változatát vezettük be. Ezek a μ paraméter változtatási módjában térnek el egymástól, amely az alábbi algoritmus *updateOfMu()* műveletével lesz kifejezve.

Bemeneti paraméterek:

- hibaküszöb (*threshold*) $\tau \in (0, 1)$;
- kívánt pontosság (*accuracy*) $\epsilon > 0$;
- a μ értékét módosító paraméter $\theta \in (0, 1)$ az első változat, illetve $\sigma \in (0, 1]$ a második változat esetén;
- a lépéshosszt csökkentő paraméter $\rho \in (0, 1)$;
- az M mátrix és a q vektor;
- a w súlyvektor;

Kimenet: x, s

Legyen $(x^0, s^0) \in \mathcal{F}^0$ és $\mu^0 = \frac{(x^0)^T s^0}{n}$ úgy, hogy $\delta(x^0, s^0; \mu^0) \le \tau$;

$x := x^0$; $s := s^0$; $\mu = \mu^0$;

while $\left(\frac{\|xs - w\|}{1 + \|c\|} > \epsilon\right)$ **or** $\left(\frac{\|Mx + q - s\|}{1 + \|q\|} > \epsilon\right)$ **do**

updateOfMu();

$w(\mu) = \left(1 - \frac{\mu}{\mu^0}\right) w + \frac{\mu}{\mu^0} c$;

$r_q = Mx + q - s$;

$rhs = w(\mu) - xs$;

$\Delta x = (M + X^{-1}S)^{-1}(X^{-1} rhs - r_q)$;

$\Delta s = X^{-1}(rhs - S\Delta x)$;

$\alpha_x = min\left\{-\frac{x_i}{\Delta x_i} | 1 \le i \le n, \Delta x_i < 0\right\}$;

$\alpha_s = min\left\{-\frac{s_i}{\Delta s_i} | 1 \le i \le n, \Delta s_i < 0\right\}$;

$\alpha = min\{\alpha_x, \alpha_s, 1\}$;

end while

Az első változatban az elméleti algoritmushoz hasonlóan a μ paraméter értékét mindig egy állandó szorzóval csökkentjük, amely a θ paraméter által lesz meghatározva.

procedure updateOfMu()
begin
$\mu := (1 - \theta)\,\mu$;
end

A belsőpontos algoritmusok implementálása esetén viszont általában úgy szoktunk eljárni, hogy az aktuális x és s vektorok alapján határozzuk meg a μ következő értékét. Ennek érdekében egy olyan μ értéket keresünk, amelyre $xs = w(\mu)$ fennáll. Kifejezve a μ értékét az alábbi egyenlőséget kapjuk:

$$\mu = \frac{\mu_0(x^T s - e^T w)}{e^T c - e^T w}. \qquad (2)$$

Figyeljük meg, hogy ebben az esetben feltételeznünk kell azt, hogy $e^T c \neq e^T w$. Abban az esetben, ha ez a feltétel nem teljesül, más x^0 és s^0 kezdeti pontokat kell választanunk.

Megjegyezzük, hogy a $w = 0$ esetben a (2) összefüggés a $\mu = \dfrac{x^T s}{n}$ alakra hozható, amely számos nem súlyozott LCP megoldása esetén használható.

Az *updateOfMu*() eljárás második változatában a fenti módon kiszámolt μ értéket egy $\sigma \in (0, 1]$ szorzóval csökkentjük:

procedure updateOfMu()
begin
$\mu := \sigma \dfrac{\mu_0(x^T s - e^T w)}{e^T c - e^T w}$;
end

Az implementálás C++ programozási nyelvben történt, Visual Studio fejlesztői környezetben, a [8] dolgozatban bevezetett kódra alapozva.

4. Numerikus eredmények

Az algoritmust pozitív szemidefinit bemeneti mátrixokra teszteltük, amelyeket mi állítottuk elő a következő módszer segítségével:
$M = LL^T$,
ahol L egy alsó háromszögmátrix. Az első esetben legyen:

$$L = \begin{pmatrix} 5 & 0 & 0 & 0 \\ 1 & 3 & 0 & 0 \\ 9 & -4 & 1 & 0 \\ -2 & 1 & 7 & 3 \end{pmatrix}.$$

Ennek a 4×4-es bemenetnek megfelelő M mátrix, q vektor és w súlyvektor:

$$M = \begin{pmatrix} 25 & 5 & 45 & -10 \\ 5 & 10 & -3 & 1 \\ 45 & -3 & 98 & -15 \\ -10 & 1 & -15 & 63 \end{pmatrix},$$

$q = -Me + e$,
$w = [0.5 \quad 1 \quad 15 \quad 0.3]^T$.

A fent leírt feladaton kívül még három esetet vizsgáltunk, amelyek megtalálhatóak a [9] weboldalon és 50×50, 100×100, illetve 700×700 méretű mátrixokra vonatkoznak. Ezen mátrixokat és a feladatokhoz rendelt súlyvektorokat véletlenszerűen generáltuk és a q értékét mindig a $q = -Me + e$ képlet alapján határozzuk meg.

Az eredményeinket az 1. és 2. táblázat tartalmazza az algoritmus két változatának megfelelően. A táblázatok fejlécében a szám a mátrix méretét jelöli. Minden esetben a $\rho = 0.95$ és $\epsilon = 10^{-5}$ értékekkel dolgoztunk.

Megállapítható, hogy a θ növelése, illetve a σ csökkentése minden esetben kevesebb iterációt eredményez.

1. táblázat. *Eredmények a θ-ra vonatkozóan*

θ	4	50	100	700
0.1	124	128	129	129
0.2	59	61	61	61
0.3	37	38	38	38
0.4	26	27	27	27
0.5	19	20	20	20
0.6	15	15	15	15
0.7	11	12	12	12
0.8	9	9	9	9
0.9	6	7	8	8

2. táblázat. *Eredmények a σ-ra vonatkozóan*

σ	4	50	100	700
0.1	8	8	9	10
0.2	10	11	12	12
0.3	13	13	14	15
0.4	17	17	18	18
0.5	21	22	23	23
0.6	28	29	30	30
0.7	40	40	41	41
0.8	62	64	64	65
0.9	131	135	135	136

5. Következtetések

A cikkben a WLCP-t vizsgáltuk meg az implementáció szemszögéből nézve. Ennek érdekében az algoritmus két változatát határoztuk meg.

A C++ programozási nyelvben implementált kód segítségével igazoltuk az algoritmus hatékonyságát. Különböző pozitív szemidefinit bemeneti mátrixok esetén vizsgáltuk az iterációszám változását a θ, illetve a σ paraméter függvényében.

Köszönetnyilvánítás

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

Szakirodalmi hivatkozások

[1] Cottle R. W., Pang J.-S., Stone R. E.: *The Linear Complementarity Problem.* Computer Science and Scientific Computing. Academic Press, Boston, 1992.

[2] Kojima M., Megiddo N., Noma T., Yoshise A.: *A Unified Approach to Interior Point Algorithms for Linear Complementarity Problems.* Springer Verlag, Berlin, Germany, 1991.

[3] Ye Y.: *A Path to the Arrow-Debreu Competitive Market Equilibrium*. Mathematical Programming volume 111/1–2. (2008) 315–348. https://doi.org/10.1007/s10107-006-0065-5

[4] Anstreicher K. M.: *Interior-Point Algorithms for a Generalization of Linear Programming and Weighted Centring, Optimization Methods and Software*. Optimization Methods and Software, 27/4–5. (2012) 605–612. https://doi.org/10.1080/10556788.2011.644791

[5] Jian Z.: *A Smoothing Newton Algorithm for Weighted Linear Complementarity Problem*. Optim Letters, 10. (2016) 499–509. https://doi.org/10.1007/s11590-015-0877-4

[6] Potra F. A.: *Weighted Complementarity Problems – a new paradigm for computing equilibria.* SIAM Journal on Optimization, 22/4. (2012) 1634–1654. https://doi.org/10.1137/110837310

[7] Asadi S., Darvay Zs., Lesaja G., Mahdavi-Amiri N., Potra F. A.: *A Full-Newton Step Interior-Point Method for Monotone Weighted Linear Complementarity Problems*. Journal of Optimization Theory and Applications, 186/3. (2020) 864–878. https://doi.org/10.1007/s10957-020-01728-4

[8] Darvay Zs., Takó I.: *Computational comparison of primal-dual algorithms based on a new software*. unpublished manuscript, 2012.

[9] Darvay Zs., Orbán A. Sz.: *Pozitív szemidefinit mátrixok és súlyvektorok generálása* (letöltve: 2021.03.06). http://cs.ubbcluj.ro/~darvay/semidefinite/