

AZ ALGEBRAILAG EKVIVALENS ÁTALAKÍTÁS MÓDSZERE SÚLYOZOTT LINEÁRIS KOMPLEMENTARITÁSI FELADATOKRA

ALGEBRAICALLY EQUIVALENT TRANSFORMATION TECHNIQUE FOR WEIGHTED LINEAR COMPLEMENTARITY PROBLEMS

Darvay Zsolt,¹ Jakab Zsanett²

¹ Babeş-Bolyai Tudományegyetem, Matematika és Informatika Kar. Kolozsvár, Románia,
 darvay@cs.ubbcluj.ro; Erdélyi Múzeum-Egyesület, Matematikai és Informatikai Szakosztály

² Babeş-Bolyai Tudományegyetem, Matematika és Informatika Kar. Kolozsvár, Románia,
 zsanett28@yahoo.com

Abstract

We study the algebraically equivalent transformation technique using an application implemented in the Java programming language that solves weighted linear complementarity problems. In general, in the case of the algebraically equivalent transformation, we divide the nonlinear equation of the system that characterizes the central path, the so-called centrality equation, by the barrier parameter. However, in this case, we divide, component-wise, the centrality equation by the right-hand side vector, which depends on the barrier parameter. We apply the same continuously differentiable and invertible function to both sides of the obtained equation and generate different search directions using Newton's method. We analyze the numerical results provided by our application in the case of applying the algebraically equivalent transformation technique using the identical map and the square root function.

Keywords: *weighted linear complementarity problem, algebraically equivalent transformation, interior-point algorithm.*

Összefoglalás

Az algebrailag ekvivalens átalakítás módszerét tanulmányozzuk egy Java programozási nyelvben implementált súlyozott lineáris komplementaritási feladatokat megoldó alkalmazás által. Az algebrailag ekvivalens átalakítás módszere esetén a centrális utat meghatározó rendszer nem lineáris egyenletét, az úgynevezett centralitási egyenletet, általában elosztjuk a barrier-paraméterrel. Ebben az esetben viszont a centralitási egyenletet a barrier-paramétertől függő jobb oldali vektorral komponensenként osztjuk el. A kapott összefüggés mindkét oldalára alkalmazunk egy folytonosan differenciálható, invertálható függvényt, majd a Newton-módszer segítségével különböző keresési irányokat kapunk. Elemezzük az alkalmazás által szolgáltatott numerikus eredményeket abban az esetben, ha az algebrailag ekvivalens átalakítás módszerét az identikus függvény, illetve a négyzetgyökfüggvény segítségével alkalmazzuk.

Kulcsszavak: *súlyozott lineáris komplementaritási feladat, algebrailag ekvivalens átalakítás, belsőpontos algoritmus.*

1. A súlyozott lineáris komplementaritási feladat ismertetése

A súlyozott komplementaritási probléma (WLCP) jelentősen kiterjeszti a hagyományos lineáris komplementaritási feladat (LCP) fogalmát. A WLCP bevezetése Potra [1] által történt 2012-ben.

Legyen $\mathbb{R}_+^n$ az n -ed rendű nem negatív valós számokból álló vektorok halmaza. Feltételezzük, hogy $M \in \mathbb{R}^{n \times n}$ egy monoton mátrix, azaz $x^T M x \geq 0$ bármely $x \in \mathbb{R}^n$ vektorra.

A hagyományos komplementaritási feladathoz hasonlóan, az alábbi rendszer tekinthető [2]:

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

ahol $x, s, q \in \mathbb{R}^n$ és $w \in \mathbb{R}_+^n$.

Megfigyelhető, hogy a súlyozott feladat esetén az xs komponensenkénti szorzat a w , nem negatív valós számokból álló vektorral kell egyenlő legyen. Emellett teljesülnie kell a $-Mx + s = q$ összefüggésnek, valamint annak is, hogy az x és s vektorok komponensei nem negatív valós számok.

A belsőpontos algoritmusok általában a centrális utat követik, a WLCP esetén viszont egy sajátos útvonalat vezetünk be. Feltételezve, hogy adottak az x_0 és s_0 kezdeti belső pontok és $\mu^0 = \frac{(x_0^T s_0)}{n}$ vezessük be a μ pozitív paramétertől függő alábbi vektort [1, 2]:

$$w(\mu) = \left(1 - \frac{\mu}{\mu_0}\right)w + \frac{\mu}{\mu_0}c, \text{ ahol } c = x_0 s_0.$$

Ezáltal a centrális út, amely kijelöli a haladási irányt, a következőképpen adható meg:

$$\begin{aligned} -Mx + s &= q, \\ xs &= w(\mu), \\ x &> 0, s > 0, \end{aligned} \quad (1)$$

ahol $\mu \in (0, \mu^0]$. Mivel az M mátrix monoton és létezik az (x^0, s^0) kezdeti belső pont, igazolható a centrális út létezése és egyértelműsége [3].

A továbbiakban a fenti rendszernek egy lehetséges módosításával foglalkozunk.

2. Az algebrailag ekvivalens átalakítás módszere

Legyen $e = [1 \ 1 \dots 1]^T$ az egyesekből álló n -dimenziós vektor. Az algebrailag ekvivalens átalakítás módszere [4, 5] esetén a hagyományos esetben

úgy járunk el, hogy az $xs = \mu e$ centralitási egyenletet elosztjuk a μ barrier-paraméterrel, ezt követően pedig egy folytonosan differenciálható és invertálható $\varphi: (0, \infty) \rightarrow \mathbb{R}$ függvényt alkalmazunk az egyenlet mindkét oldalára. Így a $\varphi\left(\frac{xs}{\mu}\right) = \varphi(e)$ egyenletet kapjuk. A WLCP esetén viszont a centralitási egyenlet az $xs = w(\mu)$ alakban van megadva, ezért ezt a jobb oldali $w(\mu)$ vektorral fogjuk komponensenként elosztani. Ezt követően alkalmazzuk a φ függvényt, így az (1) rendszer a következő alakot ölti:

$$\begin{aligned} -Mx + s &= q, \\ \varphi\left(\frac{xs}{w(\mu)}\right) &= \varphi(e), \\ x > 0, s > 0. \end{aligned}$$

A Newton-módszert alkalmazva megkapjuk azt a rendszert, amely egyértelműen meghatározza a Δx és Δs keresési irányokat [3]:

$$\begin{aligned} s\Delta x + x\Delta s &= w(\mu) \cdot \frac{\varphi(e) - \varphi\left(\frac{xs}{w(\mu)}\right)}{\varphi'\left(\frac{xs}{w(\mu)}\right)}, \\ -M\Delta x + \Delta s &= 0. \end{aligned} \quad (2)$$

Figyeljük meg, hogy a φ függvény bevezetése által a keresési irányokat meghatározó lineáris egyenletrendszer mátrixa mindig ugyanaz lesz, csak a jobb oldali vektor függ a φ függvénytől. Az implementáció esetén kétféle φ függvényt használunk, az identikus, illetve a négyzetgyök függvényt: $\varphi(t) = t$, $\varphi(t) = \sqrt{t}$.

Az algoritmus elemzése érdekében be kell vezetni a δ függvényt is, amely a centrális út adott pontjának a környezetét határozza meg. Ezáltal egy közelségi mértéket adunk meg, amely az x és s pontoknak a centrális úttól való távolságát fejezi ki:

$$\delta(x, s, \mu) = \left\| \frac{w(\mu)}{\mu} - v^2 \right\|, \text{ ahol } v = \sqrt{\frac{xs}{\mu}}, \mu > 0.$$

A továbbiakban az algoritmusnak az elméleti, illetve az implementáció szempontjából módosított változatát ismertetjük.

3. Az algoritmus

Először a teljes lépéses belsőpontos algoritmust mutatjuk be, majd az implementáció szemszögéből módosított változatot vezetjük be, amely a súlyozott lineáris komplementaritási feladatok hatékony megoldására alkalmas. Az elméleti algoritmus az alábbi módon írható [2].

WLCP-re vonatkozó elméleti algoritmus

Adott az (x^0, s^0) pár úgy, hogy $-Mx_0 + s_0 = q$.

Feltételezzük, hogy $x^0 > 0$, $s^0 > 0$, valamint

$$\delta(x^0, s^0, \mu^0) \leq \tau, \text{ ahol } \tau \in (0,1) \text{ és } \mu^0 = \frac{(x^0)^T s^0}{n}.$$

$$x = x^0, s = s^0;$$

while $\|xs - w\| > \varepsilon$ **do**

$$\mu = (1 - \theta) \mu;$$

$(\Delta x, \Delta s)$ meghatározása a (2) rendszer alapján

$$x = x + \Delta x;$$

$$s = s + \Delta s;$$

end while

Az algoritmus bemenő adatai az induló belső pontok, melyeket az x^0 és s^0 vektorok jelölnek; a $0 < \theta < 1$ változó, amely a μ barrier-paraméter csökkenéséért felel; a $0 < \tau < 1$ paraméter, amely a centrális út környezetét szabályozza, és a leállási kritériumot meghatározó $\varepsilon > 0$ érték. Emellett természetesen szükséges megadni a feladat feltételei alapján felírt összefüggésnek megfelelő M mátrixot, illetve a q vektort. Az (x^0, s^0) kezdeti pontpárhoz hozzá kell rendelni egy kezdeti μ^0 értéket. Ebből a párból kiindulva, addig lehet haladni a ciklusban, amíg a megadott feltétel teljesül. A cikluson belül a μ értékét csökkenteni kell, erre pedig a legegyszerűbb módszer, hogy egy 1-nél kisebb számmal megszorozzuk, ez esetben $(1 - \theta)$ -val. Ugyanakkor a Δx és Δs irányokat az (2) rendszerből számoljuk ki. Ebben az esetben mindig teljes Newton-lépéssel határozzuk meg az új x és s pontokat.

Mivel a teljes Newton-lépéses belsőpontos algoritmusok megvalósítása általában nem elég hatékony, ezért számos módosítást végzünk annak érdekében, hogy kisebb futási idővel rendelkező módszereket kapjunk. Az általunk implementált algoritmus esetén, a μ^0 -ra vonatkozóan használunk egy $0 < \sigma \leq 1$ szorzótényezőt a csökkentés gyorsítása céljából. Ugyanakkor bemeneti adatként megjelenik még az aktuális, valamint maximális iteráció szám, és az ε , ρ , illetve σ paraméterek. Az implementált algoritmus hatékonyságának növelése érdekében nem teljes Newton lépést használunk, hanem kiszámítjuk az $\alpha(x)$ és $\alpha(s)$ maximális lépéshosszokat a határig hányadosztestet alkalmazva. Ezt követően az $\alpha(x)$ és $\alpha(s)$ minimumát tekintjük és a kapott α pozitív valószínűség adja meg a lépés hosszúságát, melyet megszorozunk egy 0 és 1 közötti ρ állandóval.

Az aktuális x és s pontok alapján a μ kiszámítása az [6] cikkben ismertetett módon történik, feltételezve, hogy $e^T c \neq e^T w$.

WLCP-re vonatkozó módosított algoritmus

Adottak az $x^0 > 0$, $s^0 > 0$ vektorok úgy, hogy

$$-Mx_0 + s_0 = q.$$

$$\text{Legyen } 0 < \sigma \leq 1 \text{ és } \mu^0 = \frac{(x^0)^T s^0}{n};$$

Feltételezzük, hogy $\delta(x^0, s^0, \mu^0) \leq \tau$, ahol $\tau \in (0,1)$.

Legyen $0 < \rho < 1$ és $\varepsilon > 0$.

$$x = x^0, s = s^0;$$

$$iter = 0, \text{ max_iter} = 3000;$$

do

$$\mu = \sigma \cdot \left| \frac{\mu_0(x^T s - e^T w)}{e^T c - e^T w} \right|;$$

$(\Delta x, \Delta s)$ meghatározása a (2) rendszer alapján;

$\alpha(x)$ és $\alpha(s)$ meghatározása;

$$\alpha = \min\{\alpha(x), \alpha(s)\};$$

$$x = x + \rho \cdot \alpha \cdot \Delta x;$$

$$s = s + \rho \cdot \alpha \cdot \Delta s;$$

$$gap = \left| \frac{x^T s - e^T w}{e^T c - e^T w} \right|;$$

$$infeasibility = \frac{\|Mx - s + q\|}{1 + \|q\|};$$

$$iter = iter + 1;$$

while $((gap \geq \varepsilon \text{ or } infeasibility \geq \varepsilon) \text{ and } iter < \text{max_iter});$

4. Numerikus eredmények

A továbbiakban különböző súlyozott komplementaritási feladatokra mutatunk be numerikus eredményeket.

1. feladat

Az első feladat megegyezik az Asadi és Mansouri [7] által tanulmányozott 1. feladattal, azzal a különbséggel, hogy $w = 0$ helyett most egy nullától különböző súlyvektort tekintünk. Legyenek

$$n = 4,$$

$$q = \begin{pmatrix} 4.0 \\ 4.0 \\ -2.0 \\ -1.0 \end{pmatrix}, w = \begin{pmatrix} 15.0 \\ 10.0 \\ 12.0 \\ 23.0 \end{pmatrix},$$

$$M = \begin{pmatrix} 1.0 & -2.0 & 1.0 & -1.0 \\ 2.0 & 0.0 & -2.0 & 1.0 \\ 1.0 & 2.0 & 0.0 & -3.0 \\ 2.0 & -1.0 & 3.0 & 3.0 \end{pmatrix}.$$

2. feladat

A második feladat a Mansouri és Pirhaji [8] által vizsgált 5.2. feladat módosított változata. Ebben az esetben is egy $w \neq 0$ súlyvektorral dolgozunk:

$$n = 7, q = \begin{pmatrix} -1.0 \\ -3.0 \\ 1.0 \\ -1.0 \\ 5.0 \\ 4.0 \\ -1.5 \end{pmatrix}, w = \begin{pmatrix} 10.0 \\ 12.0 \\ 15.0 \\ 35.0 \\ 15.5 \\ 12.0 \\ 20.2 \end{pmatrix},$$

$$M = \begin{pmatrix} 1.0 & 0.0 & -0.5 & 0.0 & 1.0 & 3.0 & 0.0 \\ 0.0 & 0.5 & 0.0 & 0.0 & 2.0 & 1.0 & -1.0 \\ -0.5 & 0.0 & 1.0 & 0.5 & 1.0 & 2.0 & -4.0 \\ 0.0 & 0.0 & 0.5 & 0.5 & 1.0 & -1.0 & 0.0 \\ -1.0 & -2.0 & -1.0 & -1.0 & 0.0 & 0.0 & 0.0 \\ -3.0 & -1.0 & -2.0 & 1.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 4.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{pmatrix}.$$

3. feladat

Az utolsó feladat az Achache [9] által tanulmányozott 3. feladat, de ebben az esetben is a $w \neq 0$ feltételezéssel élünk. Legyenek

$$n=500, q = \begin{pmatrix} -1.0 \\ -1.0 \\ \dots \\ \dots \\ -1.0 \\ -1.0 \end{pmatrix}, w = \begin{pmatrix} v_0 \\ v_1 \\ \dots \\ \dots \\ v_{498} \\ v_{499} \end{pmatrix},$$

$$M = \begin{pmatrix} 1.0 & 2.0 & \dots & 2.0 & \dots & 2.0 \\ 0.0 & \ddots & 2.0 & \dots & 2.0 & \dots \\ \dots & 0.0 & 1.0 & 2.0 & \dots & 2.0 \\ 0.0 & \dots & 0.0 & 1.0 & 2.0 & \dots \\ \dots & 0.0 & \dots & 0.0 & \ddots & 2.0 \\ 0.0 & \dots & 0.0 & \dots & 0.0 & 1.0 \end{pmatrix},$$

ahol $v_i, 1 \leq i \leq n$ véletlenszerűen generált pozitív valós számok.

Az irányok meghatározásához először a $\varphi(t) = t$ identikus függvényt, majd a $\varphi(t) = \sqrt{t}$ négyzetgyökfüggvényt alkalmazzuk. Továbbá az első esetben legyen $\epsilon = 10^{-6}, \rho = 0,95$ és $\sigma = 0,3$, melyekre az 1. táblázatban leírt módon alakulnak a kapott értékek a WLCP-feladatok esetén.

Megfigyelhető, hogy a 4×4 -es mátrixot tartalmazó feladat esetén a négyzetgyökfüggvény alkalmazásakor több iteráció alatt jutottunk el a megoldáshoz, mint az identikus függvény esetén. A másik két feladatra vonatkozóan ennek ellenkezője mondható el (1. táblázat).

A 2. táblázatra is hasonló megfigyelések mondhatóak el, mint az 1. táblázatra, viszont az ϵ értékének csökkentésével megnőnek az iterációs számok, hiszen e módosítás során nagyobb pontossággal szeretnénk megkapni az x és s értékeit.

A ρ értékének növelése esetén az első feladatra az identikus függvényes módszer bizonyult a leghatékonyabbnak. A második feladatot a négyzetgyökfüggvényes módszer oldotta meg kevesebb iteráció alatt, míg a harmadik esetén az iterációs számok megegyeztek (3. táblázat).

A 4. táblázat alapján megállapítható, hogy több iteráció alatt jutott el a program a megoldásokhoz, mint az előző esetben, viszont ez várható volt, mivel a σ paraméter növelésével a μ változó csökkenését lelassítottuk.

1. táblázat. Eredmények az $\epsilon = 10^{-6}, \rho = 0,95, \sigma = 0,3$ értékekre.

Feladat	φ függvény	Mátrix mérete	Iterációs szám	Futási idő (ms)
1	$\varphi(t) = t$	$n = 4$	6	1,382
1	$\varphi(t) = \sqrt{t}$	$n = 4$	14	1,947
2	$\varphi(t) = t$	$n = 7$	15	3,918
2	$\varphi(t) = \sqrt{t}$	$n = 7$	10	2,977
3	$\varphi(t) = t$	$n = 500$	19	7117,815
3	$\varphi(t) = \sqrt{t}$	$n = 500$	18	7474,826

2. táblázat. Eredmények az $\epsilon = 10^{-8}, \rho = 0,95, \sigma = 0,3$ értékekre.

Feladat	φ függvény	Mátrix mérete	Iterációs szám	Futási idő (ms)
1	$\varphi(t) = t$	$n = 4$	10	1,614
1	$\varphi(t) = \sqrt{t}$	$n = 4$	19	2,362
2	$\varphi(t) = t$	$n = 7$	20	3,276
2	$\varphi(t) = \sqrt{t}$	$n = 7$	19	3,756
3	$\varphi(t) = t$	$n = 500$	23	9283,661
3	$\varphi(t) = \sqrt{t}$	$n = 500$	22	8764,180

3. táblázat. Eredmények az $\epsilon = 10^{-6}, \rho = 0,98, \sigma = 0,3$ értékekre.

Feladat	φ függvény	Mátrix mérete	Iterációs szám	Futási idő (ms)
1	$\varphi(t) = t$	$n = 4$	11	1,646
1	$\varphi(t) = \sqrt{t}$	$n = 4$	14	2,134
2	$\varphi(t) = t$	$n = 7$	15	2,772
2	$\varphi(t) = \sqrt{t}$	$n = 7$	14	3,013
3	$\varphi(t) = t$	$n = 500$	18	7102,162
3	$\varphi(t) = \sqrt{t}$	$n = 500$	18	7119,939

4. táblázat. Eredmények az $\epsilon = 10^{-6}, \rho = 0,95, \sigma = 0,9$ értékekre.

Feladat	φ függvény	Mátrix mérete	Iterációs szám	Futási idő (ms)
1	$\varphi(t) = t$	$n = 4$	138	15,872
1	$\varphi(t) = \sqrt{t}$	$n = 4$	140	21,080
2	$\varphi(t) = t$	$n = 7$	139	18,307
2	$\varphi(t) = \sqrt{t}$	$n = 7$	140	19,608
3	$\varphi(t) = t$	$n = 500$	143	100614,3
3	$\varphi(t) = \sqrt{t}$	$n = 500$	143	105317,7

5. Következtetések

A WLCP megoldására vonatkozóan bevezettük az algebrailag ekvivalens átalakítás módszerét. Egy, a Java programozási nyelvben megírt alkalmazás segítségével megállapítottuk, hogy az irány meghatározását leíró függvény befolyásolhatja a WLCP-megoldó által szolgáltatott eredményeket. Az algebrailag ekvivalens átalakítás módszerét az implementáció szemszögéből nézve az identikus függvény és a négyzetgyökfüggvény esetére vizsgáltuk. Megállapítható, hogy az iterációs szám függ az ε pontossági paramétertől, valamint a barrier-paramétert csökkentő σ , illetve a lépés hosszúságát szabályozó ρ paraméterek értékeitől is.

Köszönetnyilvánítás

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

Szakirodalmi hivatkozások

- [1] 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>
- [2] Asadi S., Darvay Zs., Lesaja G., Mahdavi-Amiri N.: *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>
- [3] Potra F. A.: *Sufficient Weighted Complementarity Problems*. Computational Optimization and Applications, 64/2. (2016) 467–488. <https://doi.org/10.1007/s10589-015-9811-z>
- [4] Darvay Zs.: *A new Algorithm for Solving Self-Dual Linear Optimization Problems*. Studia Universitatis Babeş-Bolyai, Series Informatica, 47/1 (2002) 15–26.
- [5] Darvay Zs.: *New Interior Point Algorithms in Linear Programming*. Advanced Modeling and Optimization, 5/1. (2003) 51–92.
- [6] Darvay Zs., Orbán A.-Sz.: *Implementation of the Full-Newton Step Algorithm for Weighted Linear Complementarity Problems*. Műszaki Tudományos Közlemények, 15/1. (2021) 15–18. <https://doi.org/10.33894/mtk-2021.15.04>
- [7] Asadi S., Mansouri H.: *Polynomial Interior-Point Algorithm for $P_*(\kappa)$ Horizontal Linear Complementarity Problems*. Numerical Algorithms, 63/2. (2013) 385–398. <https://doi.org/10.1007/s11075-012-9628-0>
- [8] Mansouri H., Pirhaji M.: *A Polynomial Interior-Point Algorithm for Monotone Linear Complementarity Problems*. Journal of Optimization Theory and Applications, 157/2. (2013) 451–461. <https://doi.org/10.1007/s10957-012-0195-2>
- [9] Achache M.: *Complexity Analysis and Numerical Implementation of a Short-Step Primal-Dual Algorithm for Linear Complementarity Problems*. Applied Mathematics and Computation, 216/7. (2010) 1889–1895. <https://doi.org/10.1016/j.amc.2010.03.015>