sciendo

Műszaki Tudományos Közlemények vol. 17. (2022) 6–10.
DOI: Magyar: https://doi.org/10.33895/mtk-2022.17.02
Angol: https://doi.org/10.33894/mtk-2022.17.02

EME MTSZ MTK

SZEMIDEFINIT OPTIMALIZÁLÁSRA VONATKOZÓ ALGORITMUSOK IMPLEMENTÁLÁSA JAVÁBAN

IMPLEMENTATION IN JAVA OF ALGORITHMS FOR SEMIDEFINITE OPTIMIZATION

Darvay Zsolt,[1] Garfield Adrienne[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, adriennegarfield2000@gmail.com*

Abstract

We discuss the possibility of solving the semidefinite optimization problem using interior-point algorithms. We present the primal and dual semidefinite programming problems, and then determine the interior-point condition and the optimality criteria. We analyze the central path system and the modification of this, using the method of algebraically equivalent transformation. We use the Nesterov-Todd scaling technique to obtain the proper search directions. We give a modified version of the Nesterov-Todd step interior-point algorithm based on the implementation point of view. We present some numerical results based on a code implemented in the Java programming language. We compare the results obtained for the identity map and the square root function within the framework of the algebraically equivalent transformation technique.

Keywords: *semidefinite optimization, interior-point algorithm, central path, algebraically equivalent transformation.*

Összefoglalás

A szemidefinit optimalizálási feladat belsőpontos algoritmusokkal történő megoldási lehetőségét tárgyaljuk. Bevezetjük a primál, illetve duál szemidefinit programozásai feladatokat, majd ezt követően meghatározzuk a belsőpont-feltételt és az optimalitási kritériumot. Elemezzük a centrális útnak megfelelő rendszert, valamint ennek módosítását az algebrailag ekvivalens átalakítás módszerével. A Nesterov–Todd skálázási technikát használjuk a megfelelő keresési irányok meghatározása érdekében. Az implementáció szemszögéből módosított változatát adjuk meg a teljes Nesterov–Todd-lépéses belsőpontos algoritmusnak. Numerikus eredményeket mutatunk be egy Java programozási nyelvben fejlesztett kód segítségével. Az algebrailag ekvivalens átalakítás technikája keretében használt azonos függvényre, illetve négyzetgyökfüggvényre vonatkozó eredményeket hasonlítjuk össze.

Kulcsszavak: *szemidefinit optimalizálás, belsőpontos algoritmus, centrális út, algebrailag ekvivalens transzformáció.*

1. Elméleti alapok

1.1. A szemidefinit optimalizálási feladat

A szemidefinit optimalizálás a konvex optimalizálásnak egy olyan sajátos esete, amely számos alkalmazással rendelkezik például a kombinatorikus optimalizálás területén. Mérnöki szempontból a szerkezeti optimalizálás esetén, a mesterséges intelligenciában pedig az ellipszoidokkal végzett mintaelválasztás során találkozhatunk ilyen jellegű feladatokkal [1].

Jelölje $\mathbb{R}^n$, $\mathbb{R}^{n\times n}$, illetve S^n az n-ed rendű oszlopvektorok, mátrixok, illetve szimmetrikus mátrixok halmazát. Tetszőleges $X, S \in \mathbb{R}^{n\times n}$ esetén bevezetjük az alábbi algebrai műveletet:

$$X \circ S = \mathrm{Tr}(X^T S),$$

ahol X^T az X mátrix transzponáltját jelöli. Ha az $X \in S^n$ mátrix pozitív szemidefinit, illetve pozitív definit, akkor ezt az $X \succcurlyeq 0$, illetve $X \succ 0$ alakban írjuk fel.

A szemidefinit optimalizálási feladat az alábbi módon adható meg:

$$\begin{aligned} &min\ C \circ X, \\ &A_i \circ X = b_i,\ i = 1,2,\dots,m, \\ &X \succcurlyeq 0, \end{aligned} \tag{1}$$

ahol A_i, $C \in S^n$ és $b_i \in \mathbb{R}$ minden i = 1,2,...,m esetén. Ez azt jelenti, hogy olyan szimmetrikus és pozitív szemidefinit X mátrixot keresünk, amelyre fennállnak a korlátozó feltételek, és a $C \circ X$ kifejezés minimális. Az (1) feladatot primál feladatnak nevezzük.

1.2. Duál feladat

A primál feladathoz hozzárendelhető egy duál feladat, melyben egy lineáris célfüggvény maximumát keressük az A_i, illetve C mátrixokkal kifejezett korlátozó feltételek teljesítése mellet:

$$\begin{aligned} &max\ b^T y, \\ &\sum_{i=1}^{m} y_i A_i + S = C, \\ &S \succcurlyeq 0. \end{aligned}$$

A duál feladat esetén az $y \in \mathbb{R}^m$ oszlopvektort, illetve az S pozitív szemidefinit szimmetrikus mátrixot keressük úgy, hogy teljesüljenek a megengedettségi feltételek, és a $b^T y$ lineáris függvény maximális legyen.

1.3. Kezdeti pontra vonatkozó feltétel

A továbbiakban a primál-duál feladatpárral foglalkozunk, melyet egy útkövető belsőpontos algoritmussal oldunk meg. Fontos, hogy már az (X^0, y^0, S^0) kezdőpont esetén kell, hogy teljesüljön a belsőpont-feltétel (BPF), melyet az alábbi módon írhatunk le:

$$\begin{aligned} &A_i \circ X^0 = b_i, \quad X^0 \succ 0, \quad i = 1,2,..,m, \\ &\sum_{i=1}^{m} y_i^0 A_i + S^0 = C, \quad S^0 \succ 0. \end{aligned}$$

1.4. Optimalitási kritérium

Igazolható, hogy amennyiben a BPF fennáll, akkor az alábbi rendszer határozza meg az optimális megoldást [2]:

$$\begin{aligned} &A_i \circ X = b_i, \quad i = 1,2,\dots,m, \quad X \succcurlyeq 0, \\ &\sum_{i=1}^{m} y_i A_i + S = C, \quad S \succcurlyeq 0, \\ &XS = 0. \end{aligned}$$

Az optimalitási feltétel első két sorát megengedettségi feltételnek, a harmadikat pedig komplementaritási feltételnek nevezzük.

1.5. Centrális út

A centrális útnak megfelelő rendszert úgy kapjuk, hogy az optimalitási kritérium komplementaritási feltételében szereplő nullmátrixot a μI_n kifejezéssel helyettesítjük, ahol $\mu > 0$ egy pozitív valós szám, és I_n az n-ed rendű egységmátrix. A kapott egyenlőséget centralizációs összefüggésnek nevezzük. A centrális utat leíró rendszert így adhatjuk meg:

$$\begin{aligned} &A_i \circ X = b_i, \quad i = 1,2,\dots,m, \quad X \succcurlyeq 0, \\ &\sum_{i=1}^{m} y_i A_i + S = C, \quad S \succcurlyeq 0, \\ &XS = \mu I_n. \end{aligned} \tag{2}$$

Igazolható, hogy ha a BPF teljesül, akkor minden μ pozitív valós számra a fenti rendszernek egyetlen megoldása van [2]. A μ paraméter különböző értékeire kapott megoldások határozzák meg a centrális út pontjait. Ennek mentén fog haladni a belsőpontos algoritmus.

1.6. Keresési irányok

A keresési irányoknak fontos szerepe van a belsőpontos algoritmusok különböző változatainak a bevezetésében. A keresési irányoknak egy teljes osztályát a centrális utat meghatározó rendszer algebrailag ekvivalens átalakítása [3, 4] által adhatjuk meg, amely abban rejlik, hogy a (2) rendszer harmadik egyenlete által megadott centralizációs összefüggést előbb elosztjuk a μ paraméterrel, majd ezt követően az egyenlőség mindkét oldalára alkalmazunk egy folytonosan differenciálható és invertálható $\varphi : (0,\infty) \to \mathbb{R}$ függvényt. Így az alábbi rendszerhez jutunk [5]:

$$\begin{aligned} A_i \circ X &= b_i, \quad i = 1,2,\dots,m, \quad X \succcurlyeq 0, \\ \sum_{i=1}^{m} y_i A_i + S &= C, \quad S \succcurlyeq 0, \\ \varphi\left(\frac{XS}{\mu}\right) &= \varphi(I_n). \end{aligned}$$

Megjegyezzük, hogy a φ függvény alkalmazása, tetszőleges $U \in S^n$ szimmetrikus és pozitív szemidefinit mátrix esetén, az alábbi módon történik. Felhasználjuk, hogy U felírható $U = Q^T \Lambda Q$ alakban, ahol

$\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$

a sajátértékekből alkotott diagonálmátrix és Q egy ortonormált mátrix, azaz $Q^T = Q^{-1}$. Ebben az esetben

$$\varphi(U) = Q^T \mathrm{diag}\big(\varphi(\lambda_1), \varphi(\lambda_2), \ldots, \varphi(\lambda_n)\big) Q.$$

A Wang és Bai [5] által megadott elemzés értelmében a keresési irányokat az

$$\begin{aligned} A_i \circ \Delta X &= 0, \quad i = 1,2,\ldots,m, \\ \sum_{i=1}^{m} \Delta y_i A_i + \Delta S &= 0, \\ \Delta X + X \Delta S S^{-1} &= \mu \left(\varphi' \left(\frac{XS}{\mu} \right) \right)^{-1} \left(\varphi(I_n) - \varphi \left(\frac{XS}{\mu} \right) \right) S^{-1}. \end{aligned} \tag{3}$$

rendszer segítségével adhatjuk meg.

1.7. Nesterov–Todd-skálázás

A keresési irányokat meghatározó (3) egyenletrendszernek nincs olyan egyértelmű $(\Delta X, \Delta y, \Delta S)$ megoldása, melyre ΔX szimmetrikus mátrix volna. Mivel a szimmetrikus mátrixok terében keresendő a megoldás, emiatt szükséges a rendszer skálázása. Többféle skálázási módszer létezik. A továbbiakban a Nesterov és Todd által bevezetett skálázást alkalmazzuk [6, 7].

A módszer értelmében egy P szimmetrikus mátrix bevezetése szükséges, mely által felírt egyenletrendszer közelítése lesz az előzőnek. A mátrix az alábbi két egymással egyenlő összefüggéssel adható meg.

$$P := X^{\frac{1}{2}} (X^{\frac{1}{2}} S X^{\frac{1}{2}})^{-\frac{1}{2}} X^{\frac{1}{2}} = S^{-\frac{1}{2}} (S^{\frac{1}{2}} X S^{\frac{1}{2}})^{\frac{1}{2}} S^{-\frac{1}{2}}.$$

Így az alábbi rendszert kapjuk:

$$\begin{aligned} A_i \circ \Delta X &= 0, \quad i = 1,2,\ldots,m, \\ \sum_{i=1}^{m} \Delta y_i A_i + \Delta S &= 0, \\ \Delta X + P \Delta S P^T &= \mu \left(\varphi' \left(\frac{XS}{\mu} \right) \right)^{-1} \left(\varphi(I_n) - \varphi \left(\frac{XS}{\mu} \right) \right) S^{-1}. \end{aligned} \tag{4}$$

A (4) egyenletrendszerben szereplő harmadik egyenlet bal oldalát egy egyszerű összeggé szeretnénk alakítani. Ennek érdekében bevezetjük a

$$D = P^{\frac{1}{2}}, \tag{5}$$

$$D_X = \frac{1}{\sqrt{\mu}} D^{-1} \Delta X D^{-1}, \tag{6}$$

$$D_S = \frac{1}{\sqrt{\mu}} D \Delta S D$$

mátrixokat. Figyeljük meg, hogy az (5) és (6) egyenlőségeket arra is használhatjuk, hogy a D_X, illetve D_S ismeretében meghatározzuk a ΔX és ΔS keresési irányokat.

Alkalmazva a skálázást, a fenti jelölések bevezetésével az alábbi rendszert kapjuk:

$$\begin{aligned} \bar{A}_i \circ D_X &= 0, \quad i = 1,2,\ldots,m, \\ \sum_{i=1}^{m} \Delta y_i \bar{A}_i + D_S &= 0, \\ D_X + D_S &= P_V, \end{aligned}$$

ahol

$$\bar{A}_i := \frac{1}{\sqrt{\mu}} D A_i D, \quad i = 1,2,\ldots,m,$$

$$V = \frac{1}{\sqrt{\mu}} DSD,$$

$$P_V = \sqrt{\mu} D^{-1} (D \varphi'(V^2) D^{-1})^{-1} \cdot \cdot (\varphi(I_n) - D\varphi(V^2) D^{-1}) S^{-1} D^{-1}.$$

2. Implementációra vonatkozó sajátosságok

2.1. Hibavektorok és hibamátrixok

A belsőpontos algoritmusok implementációja esetén nem mindig élhetünk azzal a feltételezéssel, hogy a megengedettségi feltételek az algoritmus minden iterációjában teljesülni fognak. Ez azt jelenti, hogy a (4) rendszer első két egyenletének jobb oldalán nem feltétlenül a 0 vektor, illetve a zérusmátrix jelenik meg. Mivel az $A_i \circ X = b_i$ egyenlőség nem mindig áll fenn, emiatt szükséges az alábbi jelölés bevezetése:

$$r_i^b = A_i \circ X - b_i, \quad i = 1,2,\ldots,m.$$

Megjegyezzük, hogy ily módon lényegében az r^b hibavektort vezettük be, melynek komponensei a fenti különbségek. Hasonlóan kell eljárni a

$$\sum_{i=1}^{m} y_i A_i + S = C$$

egyenlettel, melyhez hozzárendeljük az R^C hibamátrixot:

$$R^C = \sum_{i=1}^{m} y_i A_i + S - C.$$

A keresési irányokat meghatározó (4) rendszer első két egyenlete ennek megfelelően módosulni fog:

$$A_i \circ \Delta X = -r_i^b, \quad i = 1,2,\dots,m,$$
$$\sum_{i=1}^{m} \Delta y_i A_i + \Delta S = -R^C,$$
$$\Delta X + P\Delta S P^T = \mu\left(\varphi'\left(\frac{XS}{\mu}\right)\right)^{-1}\left(\varphi(I_n) - \varphi\left(\frac{XS}{\mu}\right)\right)S^{-1}. \tag{7}$$

Bevezetve az

$$\bar{r}_i^b = \frac{r_i^b}{\mu}, \quad i = 1,2,\dots,m,$$
$$\bar{R}^C = \frac{1}{\sqrt{\mu}} D R^C D$$

jelöléseket, a (7) rendszer skálázott alakja így írható:

$$\bar{A}_i \circ D_X = -\bar{r}_i^b, \quad i = 1,2,\dots,m,$$
$$\sum_{i=1}^{m} \Delta y_i \bar{A}_i + D_S = -\bar{R}^C, \tag{8}$$
$$D_X + D_S = P_V.$$

2.2. Lépéshossz

A Wang és Bai [5] által bevezetetett elméleti algoritmus teljes Nesterov–Todd-lépéses, amely azt jelenti, hogy az új (X,y,S) hármast az eredetiből úgy kapjuk, hogy hozzáadjuk a $(\Delta X, \Delta y, \Delta S)$ által megadott irányokat. Ebben az esetben a μ paraméter csökkentése az $(1-\theta)$ szorzó segítségével történik, ahol $0<\theta<1$. A θ megválasztásából kiderül, hogy rövid lépéses algoritmusról van szó. Az implementáció szempontjából viszont hatékonyabb az X és S mátrixok alapján megválasztani a következő μ értéket az alábbi módon:

$$\mu = \sigma\frac{X \circ S}{n},$$

ahol $0<\sigma<1$ egy rögzített paraméter. Ugyanakkor a lépéshossz megválasztása segítségével érjük el azt, hogy a kapott X és S a pozitív szemidefinit mátrixok kúpjában maradjon. Ennek érdekében tetszőleges $M \in S^n$ esetén legyen $\lambda(M)$ az M mátrix sajátértékeiből alkotott oszlopvektor, és vezessük be a

$$\lambda_{min}(M) = min(\lambda(M)),$$

jelölést is, amely megadja az M mátrix legkisebb sajátértékét. Hasonlóan vezethetjük be a $\lambda_{max}(M)$ jelölést is, amely az M mátrix legnagyobb sajátértékét adja meg. A lépéshossz meghatározását a Tütüncu, Toh és Todd [8] által megadott módszer módosításával végezzük. Bevezetve az

$$\alpha(M) = \begin{cases} \dfrac{-1}{\lambda_{min}(M)}, & \lambda_{min}(M) \le -1 \\ 1, & \lambda_{min}(M) \in (-1, +\infty) \end{cases} \tag{9}$$

jelölést, az X mátrixra alkalmazott lépéshossz

$$\alpha_X = \alpha(X^{-1}\Delta X), \tag{10}$$

az y vektorra és az S mátrixra kiszámított lépéshossz pedig

$$\alpha_S = \alpha(S^{-1}\Delta S) \tag{11}$$

lesz. Megjegyezzük, hogy a (10) és (11) által kiszámolt értékeket egy $0<\rho<1$ konstans szorzóval is csökkenteni fogjuk, ezáltal biztosítva, hogy az X és S a pozitív szemidefinit mátrixok kúpjának belsejében marad.

2.3. Megállási feltétel

Az algoritmus megállási feltétele három részre bomlik. Az első arra vonatkozik, hogy a

$$\text{relgap} = \frac{X \circ S}{1 + |C \circ X| + |b^T y|} \tag{12}$$

kifejezéssel megadott relatív dualitási rés egy rögzített $\varepsilon>0$ valós számnál kisebb legyen. Vezessük be az

$$A \circ X = \begin{bmatrix} A_1 \circ X \\ A_2 \circ X \\ \dots \\ \dots \\ A_m \circ X \end{bmatrix}$$

jelölést. Mivel az algoritmus során megtörténhet, hogy egy adott iterációban a megengedettség nem teljesül, ezért a megállási feltétel részeként megadunk két erre vonatkozó mértéket is. A primál megengedettséget a

$$\text{pinfeas} = \frac{\|A \circ X - b\|}{1 + \|b\|} = \frac{\sqrt{\sum_{i=1}^{m}(A_i \circ X - b_i)^2}}{1 + \sqrt{\sum_{i=1}^{m} b_i^2}} \tag{13}$$

kifejezéssel, a duál megengedettséget pedig a

$$\text{dinfeas} = \frac{\|\sum_{i=1}^{m} y_i A_i + S - C\|}{1 + \|C\|} \tag{14}$$

segítségével fejezzük ki, ahol tetszőleges $M \in S^n$ mátrix esetén $\|M\| = \sqrt{\left(\lambda_{max}(M^T M)\right)} = |\lambda_{max}(M)|$.

Figyeljük meg, hogy a primál megengedett megoldás esetén a pinfeas értéke nulla. Hasonlóan belátható az is, hogy duál megengedett megoldás esetén a dinfeas értéke zérus. Ezért a megállási feltétel részeként azt fogjuk vizsgálni, hogy a pinfeas, illetve dinfeas értéke kisebb-e egy adott $\varepsilon>0$ valós számnál.

3. Az algoritmus

A fenti elméleti leírás alapján módosítjuk a Wang és Bai [5] által bevezetett belsőpontos algoritmust annak érdekében, hogy hatékony implementációt tudjunk megvalósítani.

Szemidefinit optimalizálásra vonatkozó belsőpontos algoritmus

Legyen $\varepsilon > 0$ *a pontossági paraméter,* $0 < \rho < 1$ *konstans szorzó, mely csökkenti a lépés méretét, és* $0 < \sigma < 1$ *konstans szorzó, mely csökkenti a* μ *értékét.*

Feltételezzük, hogy az (X^0, y^0, S^0) *hármasra teljesül a BPF, és legyen* $\mu_0 = \dfrac{X^0 \circ S^0}{n}$ *linear;*

begin
$X := X^0$; $y := y^0$; $S := S^0$;
$\mu := \mu_0$;
relgap := 1; pinfeas := 1; dinfeas := 1;
while relgap $\geq \varepsilon$ **or** pinfeas $\geq \varepsilon$ **or** dinfeas $\geq \varepsilon$ **do**
begin
meghatározzuk a $(D_X, \Delta y, D_S)$ *hármast (8) alapján;*
meghatározzuk a ΔX *mátrixot (5) alapján;*
meghatározzuk a ΔS *mátrixot (6) alapján;*
kiszámítjuk az α_X *lépéshossz értékét (10) alapján;*
kiszámítjuk az α_S *lépéshossz értékét (11) alapján;*
$X := X + \rho\, \alpha_X \Delta X$;
$y := y + \rho\, \alpha_S \Delta y$; $S := S + \rho\, \alpha_S \Delta S$;
$\mu = \sigma\, (X \circ S)/n$;
meghatározzuk a relgap *értékét (12) alapján;*
meghatározzuk a pinfeas *értékét (13) alapján;*
meghatározzuk a dinfeas *értékét (14) alapján;*
end
end.

4. Numerikus eredmények

A szemidefinit optimalizálásra vonatkozó belsőpontos algoritmust a [9] weboldalon található példákra teszteltük. Az első változatban az algebrailag ekvivalens átalakítás módszerét az identikus függvénnyel, míg a másodikban a négyzetgyökfüggvénnyel alkalmaztuk. Az **1. táblázat** mutatja be a kapott eredményeket.

Megállapítható, hogy az iterációszámok nem térnek el jelentősen egymástól, de bizonyos esetekben a négyzetgyökfüggvényre alapozott módszer hatékonyabban működik az identikus függvényesnél.

1. táblázat. *Eredmények az* $\varepsilon = 10^{-7}$, $\rho = 0{,}95$, $\sigma = 0{,}3$ *értékekre*

Feladat	φ függvény	Mátrix mérete	Iterációszám
sdp01	$\varphi(t) = t$	n=20	21
sdp01	$\varphi(t) = \sqrt{t}$	n = 20	18
sdp02	$\varphi(t) = t$	n = 30	19
sdp02	$\varphi(t) = \sqrt{t}$	n = 30	18
sdp03	$\varphi(t) = t$	n = 40	17
sdp03	$\varphi(t) = \sqrt{t}$	n=40	15

Köszönetnyilvánítás

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

Szakirodalmi hivatkozások

[1] Vandenberghe L., Boyd S.: *Semidefinite Programming.* SIAM Review. 38/1. (1996) 49–95. https://doi.org/10.1137/1038003

[2] Klerk E. de: *Aspects of Semidefinite Programming. Interior Point Algorithms and Selected Applications.* Kluwer Academic Publishers, Dordrecht, 2002.

[3] Darvay Zs.: *A new Algorithm for Solving Self-Dual Dinear Optimization Problems.* Studia Universitatis Babeş-Bolyai, Series Informatica, 47/1. (2002) 15–26.

[4] Darvay Zs.: *New Interior Point Algorithms in Linear Programming.* Advanced Modeling and Optimization, 5/1. (2003) 51–92.

[5] Wang G. Q., Bai Y. Q.: *A New Primal-Dual Path-Following Interior-Point Algorithm for Semidefinite Optimization.* Journal of Mathematical Analysis and Applications, 353/1. (2009) 339–349. https://doi.org/10.1016/j.jmaa.2008.12.016

[6] Nesterov Yu. E., Todd M. J.: *Primal-Dual Interior-Point Methods for Self-Scaled Cones.* SIAM Journal on Optimization, 8/2. (1998) 324–364. https://doi.org/10.1137/S1052623495290209

[7] Nesterov Yu. E., Todd M. J.: *Self-Scaled Barriers and Interior-Point Methods for Convex Programming.* Mathematics of Operations Research, 22/1. (1997) 1–42. https://doi.org/10.1287/moor.22.1.1

[8] Tütüncü R. H., Toh K. C., Todd M. J.: *Solving Semidefinite-Quadratic-Linear Programs Using SDTP3.* Mathematical Programming, Ser. B 95/2. (2003) 189–217. https://doi.org/10.1007/s10107-002-0347-5

[9] Darvay Zs., Garfield A.: *Examples of Semidefinite Optimization Problems.* 2022. https://www.cs.ubbcluj.ro/~darvay/sdp/ (letöltve: 2022. november 17.)