Alkalmazott Matematikai Lapok **30** (2013), 1-21.

A KVADRATIKUS SZIMPLEX ALGORITMUS VÉGESSÉGE INDEXVÁLASZTÁSI SZABÁLYOK ALKALMAZÁSA ESETÉN

ILLÉS TIBOR, NAGY ADRIENN

Dolgozatunkban bebizonyítjuk a kvadratikus primál szimplex módszer végességét, a lineáris feltételes, konvex kvadratikus optimalizálási feladatra, ciklizálás elleni indexválasztási szabályok alkalmazásával. Az eredeti kvadratikus primál szimplex módszert Wolfe, illetve van de Panne és Whinston dolgozták ki, és több cikkben publikálták az 1960-as években. Az említett szerzők, a kvadratikus primál szimplex módszer végességét az ún. perturbációs eljárásra alapozva igazolták.

Megmutatjuk, hogy a kvadratikus szimplex módszer ciklizálásához szükséges, hogy a feladat degenerált legyen (degenerált feladat olyan, amelyben minden bázisbeli, a hányadosteszt részét képező primál változó értéke nulla), továbbá a feladathoz tartozó Karush–Kuhn–Tucker-rendszerben a transzformált oszlopokban a kvadratikus célfüggvénynek megfelelő komponensek nullák.

Gondolatmenetünkből következik, hogy a kvadratikus primál szimplex módszer véges mindazon indexválasztási szabályok esetén, melyek kizárólag a transzformált jobboldal és redukált költségek előjelére hagyatkoznak, és melyek alkalmazása esetén a lineáris programozási feladatra kidolgozott, hagyományos primál szimplex algoritmus véges.

1. Bevezető

Az 1950-es évek kezdetétől, többször az érdeklődés középpontjába került, a következő *lineáris feltételes, kvadratikus optimalizálási feladat (LKOF)*

$$\begin{aligned} \min \frac{1}{2}\,\mathbf{x}^T Q\mathbf{x} + \mathbf{c}^T\mathbf{x} \\ A\mathbf{x} \le \mathbf{b}, \quad \mathbf{x} \ge \mathbf{0}, \end{aligned}$$

ahol $Q \in \mathbb{R}^{n\times n}$ és $A \in \mathbb{R}^{m\times n}$ mátrixok, illetve $\mathbf{c} \in \mathbb{R}^n$ és $\mathbf{b} \in \mathbb{R}^m$ vektorok, $\mathbf{x} \in \mathbb{R}^n$ az ismeretlenek vektora. A *megenegedett megoldások halmaza*

$$\mathcal{P} = \{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} \le \mathbf{b},\ \mathbf{x} \ge \mathbf{0}\} \subset \mathbb{R}^n$$

egy konvex poliéder, és a célfüggvény $f : \mathcal{P} \to \mathbb{R}$ kvadratikus függvény, amelyet

$$f(\mathbf{x}) = \frac{1}{2}\,\mathbf{x}^T Q\mathbf{x} + \mathbf{c}^T\mathbf{x}$$

alakban adunk meg. Valamely $\mathbf{x}^* \in \mathcal{P}$ megengedett megoldást *optimális* megoldásnak nevezünk, ha

$$f(\mathbf{x}^*) \le f(\mathbf{x}) \quad \text{teljesül, bármely } \mathbf{x} \in \mathcal{P}$$

esetén. Most már bevezethetjük az *optimális megoldások halmaz*át az alábbi formában:

$$\mathcal{P}^* = \{\mathbf{x}^* \in \mathcal{P} : f(\mathbf{x}^*) \le f(\mathbf{x}) \quad \text{teljesül, bármely } \mathbf{x} \in \mathcal{P}\}.$$

A kutatások homlokterébe már az 1960-as években, a lineáris feltételes, kvadratikus optimalizálási feladat (hatékony) megoldhatóságának, illetve alkalmazási területének a vizsgálata került. A jól megoldható részosztályok beazonosítása és leírása gyorsan megtörtént, hiszen, ha Q pozitív szemidefinit mátrix, akkor a fenti feladat konvex programozási feladat.

A lineáris feltételes, kvadratikus optimalizálási feladat megoldására már az 1950-es évek végén, az 1960-as évek elején általánosították a szimplex módszert. A hagyományos kvadratikus szimplex algoritmus témakörében számos publikáció jelent meg [28, 29, 30, 31, 36][1].

A lineáris feltételes, kvadratikus optimalizálási feladatokból kiindulva könnyen felírhatók az ún. *általános lineáris komplementaritási feladatok*, amelyek igen széles alkalmazási területtel rendelkeznek, ezért a kezdetektől népszerűek voltak a kutatók körében. Lineáris komplementaritási feladatok megoldására is pivot algoritmusokat dolgoztak ki először. Ezek közül a legismertebb a *Lemke-* [26] és a *criss–cross algoritmus* [21]. Terlaky algoritmusa [32] nem igényli a pivot tábla megnagyobbítását.

A lineáris komplementaritási feladatok megoldhatóságának kérdése összefügg a feladat mátrixának tulajdonságával. Az egyik érdekes kutatási irány a lineáris komplementaritási feladatok esetén az volt, hogy a criss–cross algoritmus általánosításának segítségével milyen tulajdonsággal kell, hogy rendelkezzen a feladat mátrixa annak érdekében, hogy a feladat véges lépésben megoldható legyen. Ezen a területen az első eredményeket, az ún. *biszimmetrikus mátrix*okkal adott lineáris komplementaritási feladatok esetén érték el a kutatók, igazolva, hogy a criss-cross algoritmus megfelelő variánsa, ciklizálás ellenes indexválasztási szabályok alkalmazásával, véges [1, 21, 34].

Az igazi elméleti kérdés az volt, hogy milyen tulajdonságokkal kell rendelkeznie a mátrixnak ahhoz, hogy a lineáris komplementaritási feladat a konvex optimalizálási feladatok közé tartozzon és a pivot algoritmusok véges sok lépésben

[1] Ezeknek közös jellemzője, hogy az algoritmus végességének a bizonyítására az ún. *perturbációs módszer*t alkalmazzák.

megoldják a feladatot. Mint később kiderült, a Cottle és társszerzői [4] által bevezetett *elégséges mátrix*ok osztálya biztosítja ezt [12]. Természetes módon merült fel az a kérdés, hogy mi történik, ha a lineáris komplementaritási feladat mátrixának tulajdonságairól nincsen információnk. Erre az esetre dolgoztak ki Fukuda és társszerzői egy olyan minimál indexes criss-cross algoritmus változatot [10], amelyik elégséges mátrixok esetén ugyanúgy működik, mint a korábbi változat [12], míg nem elégséges mátrixok esetén vagy véges sok lépésben megoldja a feladatot, vagy bizonyítékot szolgáltat arra, hogy a mátrix nem elégséges mátrix. Csizmadia és Illés [6] megmutatták, hogy az ismert ciklizálás ellenes index választási szabályok mindegyike alkalmas arra, hogy a criss-cross algoritmus végességét biztosítsák, általános lineáris komplementaritási feladat megoldásakor, a Fukudáék által bevezetett értelemben. A közelmúltban Csizmadia és társszerzői egy egész, új ciklizálás ellenes index választási szabály osztályt definiáltak, az ún. s-monoton index választási szabályokat, és ezekre igazolták, hogy a legáltalánosabb criss-cross algoritmus is véges az összes s-monoton index választási szabály alkalmazása mellett [9]. A lineáris komplementaritási feladat és a criss-cross algoritmus kapcsolatáról szóló érdekes eredmények nagyrészét jól foglalja össze Csizmadia doktori (PhD) disszertációja [7].

Annak ellenére, hogy a jelen dolgozatnak nem tárgya a lineáris feltételes, kvadratikus optimalizálási feladat speciális osztályaira kifejlesztett belsőpontos megoldási módszerek tárgyalása, mégis úgy gondoljuk, hogy a teljesség igénye nélkül néhány érdekesebb belsőpontos algoritmust megemlítenénk. A belsőpontos módszerek 1980-as évek második felében való megjelenése óta időről-időre fellángol az a vita, hogy milyen szempontok szerint célszerű, egy-egy feladatosztály esetén, a pivot- és belsőpontos algoritmusokat összehasonlítani. Lineáris programozási feladatokra a több szempont szerinti összehasonlítást Illés és Terlaky [15] végezték el. Hasonló összefoglaló cikk, lineáris feltételes, kvadratikus optimalizálási feladatok megoldó algoritmusairól – legjobb tudomásunk szerint – még nem készült.

A belsőpontos módszerek között eléggé elterjedtek a primál-duál típusú algoritmusok. Primál-duál belsőpontos módszerekkel a lineáris feltételes, kvadratikus optimalizálási feladatok megoldását, az optimalitási feltételekből – általános lineáris komplementaritási feladatokból – nyerhető centrális út feladat sorozat iteratív megoldásával állítják elő. A centrális út feladatok, iterációról-iterációra, egyre kisebb centralitási paraméterhez tartoznak. A primál-duál belsőpontos módszerek megállási kritériuma az, hogy a dualitás rés egy előre megadott $\varepsilon > 0$ paraméter alá kerül. Ekkor azt mondjuk, hogy a belsőpontos algoritmus egy *ε-optimális* megoldást állított elő.

A centrális út létezése és egyértelműsége [25] alapvetően fontos a primál-duál belsőpontos algoritmusok működése szempontjából. Az operációkutatók egy jelentős részében él az a tévhit, hogy a belsőpontos algoritmusokkal nem lehet pontos megoldást előállítani. Ezt a tévhitet cáfolta meg elégséges lineáris komplementaritási feladatok esetén Illés és társszerzőinek a cikke [14].

A pivot algoritmusokkal összehasonlítva a primál-duál belsőpontos algoritmusokat, az az elvárásunk, hogy a legfontosabb mátrix osztályok (pozitív szemidefinit, illetve elégséges mátrixok) esetén a racionális mátrixokkal és vektorokkal adott lineáris komplementaritási feladatokat elméletileg hatékonyan oldják meg, azaz az algoritmus iterációinak számára polinomiális iterációszám korlát létezzen.

A pozitív szemidefinit mátrixszal adott lineáris feltételes, kvadratikus optimalizálási feladatból származtatott lineáris komplementaritási feladat megoldására Kojima és társszerzői, egy korábbi lineáris programozási feladatra megfogalmazott, kis lépéses, primál-duál belsőpontos algoritmusuk [23] általánosításával adtak meg olyan belsőpontos algoritmust, amelyre polinomiális iterációszám korlátot igazoltak [24]. A biszimmetrikus mátrixszal megfogalmazott lineáris komplementaritási feladat esetén, Kojimaék primál-duál belsőpontos algoritmusának a polinomiális iterációszámát, egy – a célfüggvényben szereplő mátrix pozitív szemidefinit tulajdonságából származtatott – egyenlőtlenség segítségével igazolták. Természetesen merült fel a kérdés, hogy milyen tulajdonságú mátrixok esetén lehet hasonló, a komplexitás bizonyítás szempontjából használható egyenlőtlenséget levezetni. Kojima és társszerzői [25] a P*(κ)-mátrixok osztályának bevezetésével adták meg a választ erre a kérdésre, ahol $\kappa \geq 0$ valós, meghatározott paraméter. A P*(κ)-mátrixok a pozitív szemidefinit mátrixok egy lehetséges általánosításai, amelyek rendelkeznek még azzal a fontos tulajdonsággal is, hogy a lineáris komplementaritási feladathoz tartozó centrális út feladatnak létezik egyértelmű megoldása bármely $\mu > 0$ centralitási paraméter esetén.

A P*(κ)-mátrixok és az elégséges mátrixok kapcsolatát teremti meg a P*-mátrixok osztályának definiálása, amely a P*(κ)-mátrixosztályok uniója, amikor a κ paraméter befutja a nemnegatív valós számok halmazát. Väliaho megmutatta, hogy a P*-mátrixok, elégséges mátrixok [35]. A másik irányú tartalmazás igazolása Cottle és Guu [11, 5], illetve Kojima és társszerzői [25] eredménye.

Az elégséges mátrixokkal adott lineáris komplementaritási feladatok témakörében még napjainkban is jelennek meg új belsőpontos algoritmusok, illetve régi algoritmusok új elemzései. Ezek közül mi két cikkre [13, 16] hívnánk fel a figyelmet, amelyek szerepet játszottak a lineáris komplementaritási feladatok megoldhatóságának kiterjesztésében. Az egyik érdekes kérdés az volt, hogy Fukudaék [10] eredményéhez hasonlóan, készíthető-e olyan belsőpontos algoritmus, amelyik elégséges mátrixok esetén ugyanúgy működik, mint a korábbi belsőpontos algoritmusok [13, 16], míg nem elégséges mátrixok esetén *polinomiális lépésben* megoldja a feladatot, vagy bizonyítékot szolgáltat arra, hogy a mátrix nem elégséges mátrix [18, 17]. Az ilyen típusú algoritmusok első, részletes és kimerítő tárgyalását Nagy Marianna adja meg doktori (PhD) disszertációjában [27].

Visszatérve a lineáris feltételes, kvadratikus optimalizálási feladatok pivot algoritmusokkal való megoldásának kérdésére, elmondhatjuk, hogy a későbbiekben megjelent pivotálási algoritmusok jellemzően (általános) lineáris komplementaritási feladatok megoldására felírt algoritmusok voltak.

Dolgozatunkban megmutatjuk, hogy a hagyományos, lineáris feltételes, kvadratikus programozási feladat, lineáris komplementaritási feladatára felírt kvadratikus primál szimplex algoritmus is véges, azon hagyományos, ciklizálás ellenes indexválasztási szabályok alkalmazása esetén, melyek kizárólag a redukált költségek és duál változók előjelén, illetve a változók indexein alapulnak. Azaz a lineáris feltételes, kvadratikus programozási feladat megoldására általánosított primál szimplex algoritmus végességének bizonyításhoz nincsen szükség a perturbációs eljárás használatára.

A ciklizálás ellenes indexválasztási szabályok a minimál index, a last-in-first-out, és a leggyakrabban választott változó elvét használó szabályok. A criss-cross algoritmus esetén lineáris feltételes, kvadratikus programozási feladatra, az algoritmus végességét a felsorolt indexválasztási szabályok használata mellett Illés és társszerzői igazolták [1]. A témakörben, pivot algoritmusok végességével kapcsolatban az eddigi legáltalánosabb eredményeket Csizmadia és társszerzői publikálták [9] az ún. **s**-*monoton szabály*ok esetében.

1.1. Jelölések

Cikkünkben a mátrixokat dőlt nagy betűkkel, a vektorokat vastag betűkkel jelöljük. A skalárok normál kisbetűk, az index halmazok pedig kaligrafikus nagybetűk. Egy mátrix oszlopát alsóindexszel, míg a sorokat felsőindexszel jelöljük a továbbiakban. Jelölje $A \in \mathbb{R}^{m\times n}$ a lineáris feltételes, kvadratikus optimalizálási feladat mátrixszát, $\mathbf{b} \in \mathbb{R}^m$ a jobboldalt; az általánosság korlátozása nélkül feltehetjük, hogy $rank(A) = m$. A célfüggvény lineáris részét $\mathbf{c} \in \mathbb{R}^n$, a kvadratikus részét $Q \in \mathbb{R}^{n\times n}$ jelöli. A lineáris feltételes, kvadratikus programozáshoz tartozó, lineáris komplementaritási feladat mátrixát $M \in \mathbb{R}^{K\times K}$ jelöli, ahol $K = n + m$. A következő részben bevezetett lineáris komplementáritási feladat jobboldal vektorát $\mathbf{q} \in \mathbb{R}^{n+m}$ jelöli.

Legyen $\mathcal{I} = \{1, 2, \ldots, 2K\}$ a változók indexhalmazai, és $\mathcal{I}_B$ jelölje a bázis indexhalmazait. $\mathcal{I}_B = \mathcal{I}_B^p \cup \mathcal{I}_B^d$, ahol $\mathcal{I}_B^p$ a primál bázis változók indexhalmaza, míg $\mathcal{I}_B^d$ a duál bázis változók indexhalmaza. Hasonlóan megadható az $\mathcal{I}$ és $\mathcal{I}_N$ indexhalmazok felbontása is, azaz $\mathcal{I} = \mathcal{I}^p \cup \mathcal{I}^d$ és $\mathcal{I}_N = \mathcal{I}_N^p \cup \mathcal{I}_N^d$.

Egy B bázishoz tartozó rövid pivot táblát $T := B^{-1}N$ jelöli, ahol $N \in \mathbb{R}^{K\times K}$ a $[-M, I] \in \mathbb{R}^{K\times 2K}$ a mátrix nem bázis részmátrixa. A transzformált jobboldalt pedig $\overline{\mathbf{q}} := B^{-1}\mathbf{q}$ képlettel számolhatjuk ki. Az egyes együtthatók a rövid pivot táblában legyenek a t_{ij}-együtthatókkal jelölve.

1.2. A lineáris feltételes kvadratikus programozási feladat

Ha Q pozitív szemidefinit mátrix, akkor az (LKOF) konvex programozási feladat [22].

A feladathoz rendelt Lagrange-függvény [22] a következő:

$$
\begin{aligned}
&L : \mathbb{R}_{\oplus}^{m+n} \to \mathbb{R} \\
&L(\mathbf{x}, \mathbf{y}, \mathbf{z}) = f(x) + \mathbf{y}^T(A\mathbf{x} - \mathbf{b}) - \mathbf{z}^T\mathbf{x} \qquad (1)
\end{aligned}
$$

A konvex Karush–Kuhn–Tucker-tételt alkalmazva $\mathbf{x}^*$ akkor és csak akkor primál optimális megoldás, ha $\exists \mathbf{y}^* \in \mathbb{R}_{\oplus}^m, \mathbf{z}^* \in \mathbb{R}_{\oplus}^n$ úgy, hogy $(\mathbf{y}^*, \mathbf{z}^*) \neq \mathbf{0}$ és $(\mathbf{x}^*, \mathbf{y}^*, \mathbf{z}^*)$ kielégíti a

$$
\begin{aligned}
Q\mathbf{x} + c + A^T\mathbf{y} - \mathbf{z} &= \mathbf{0} && (2)\\
A\mathbf{x} - \mathbf{b} &\leq \mathbf{0} && (3)\\
\mathbf{y}^T(A\mathbf{x} - \mathbf{b}) &= \mathbf{0} && (4)\\
\mathbf{z}^T\mathbf{x} &= 0 && (5)\\
\mathbf{x}, \mathbf{y}, \mathbf{z} &\geq \mathbf{0} && (6)
\end{aligned}
$$

rendszert.

Bevezetve az $\mathbf{s} \in \mathbb{R}_{\oplus}^m$ változót, a fenti rendszer a konvex (LKOF) optimalitási kritériumait adja meg:

$$
\begin{aligned}
-Q\mathbf{x} - A^T\mathbf{y} + \mathbf{z} &= \mathbf{c} && (7)\\
A\mathbf{x} + \mathbf{s} &= \mathbf{b} && (8)\\
\mathbf{y}^T\mathbf{s} &= 0 && (9)\\
\mathbf{z}^T\mathbf{x} &= 0 && (10)\\
\mathbf{x}, \mathbf{s}, \mathbf{y}, \mathbf{z} &\geq \mathbf{0} && (11)
\end{aligned}
$$

Ugyanez mártix alakban kifejezve, a biszimmetrikus, lineáris komplementaritási feladatra $(BLCP)$ vezet:

$$
\begin{pmatrix} \mathbf{s} \\ \mathbf{z} \end{pmatrix} - \begin{pmatrix} -A & 0 \\ Q & A^T \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ \mathbf{c} \end{pmatrix} \qquad (12)
$$

$$
\begin{aligned}
\mathbf{y}^T\mathbf{s} &= 0 && (13)\\
\mathbf{z}^T\mathbf{x} &= 0 && (14)\\
\mathbf{x}, \mathbf{s}, \mathbf{y}, \mathbf{z} &\geq 0 && (15)
\end{aligned}
$$

A lineáris feltételes, konvex kvadratikus programozási feladat tehát ekvivalens a következő lineáris komplementaritási feladattal: keressünk olyan vektorokat, amelyek kielégítik a

$$
\begin{aligned}
-M\mathbf{u} + \mathbf{v} &= \mathbf{q} && (16)\\
\mathbf{u}^T\mathbf{v} &= 0 && (17)\\
\mathbf{u}, \mathbf{v} &\geq 0 && (18)
\end{aligned}
$$

rendszert, ahol

$$M = \begin{pmatrix} Q & A^T \\ -A & 0 \end{pmatrix}, \quad \mathbf{q} = \begin{pmatrix} \mathbf{c} \\ \mathbf{b} \end{pmatrix} \quad \text{és} \quad \mathbf{v} = \begin{pmatrix} \mathbf{z} \\ \mathbf{s} \end{pmatrix}, \quad \text{illetve} \quad \mathbf{u} = \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix}.$$

Valamint (18) miatt (17) nyilvánvalóan $v_j u_j = 0$, (j = 1,. . . ,m+n). Az M mátrix definíciójában az egyszerűbb felírási forma kedvéért a sorokat az eredeti felíráshoz képest felcseréltük.

A lineáris feltételes konvex kvadratikus programozási feladatból származó lineáris komplementaritási feladat mátrixát biszimmetrikus mátrixnak nevezzük, melynek számos hasznos tulajdonsága ismert [33].

A lineáris feltételes konvex kvadratikus programozási feladat gyenge és erős dualitás tételeiről, optimalitási kritériumáról, megoldási módszereiről a de Klerk és szerzőtársai által írt jegyzetben [22] olvashatunk.

1.3. A kvadratikus primál szimplex algorimus

Dolgozatunkban a Wolfe-féle kvadratikus primál szimplex algoritmust [36] vizsgáljuk, melynek egy jó összefoglalását találjuk a [30] dolgozatban is.

A $[-M, I]$ mátrix bármely reguláris $K \times K$ részmátrixát *bázisnak* nevezzük. A lineáris komplementaritási (16)-(18) feladat egy bázisát *komplementáris*nak nevezzük, ha teljesülnek a komplementaritási feltételek, azaz $\mathbf{xz} = \mathbf{0}$ és $\mathbf{sy} = \mathbf{0}$. Az új változókkal $\mathbf{uv} = \mathbf{0}$ alakban is kifejezhetjük a komplementaritást.

A kvadratikus primál szimplex algoritmus egy komplementáris, primál megengedett bázisból indul. Ilyen bázis előállítható az eredeti primál megengedettségi feladat egy megengedett bázisát kiegészítve a lineáris komplementáris feladat bázisává a duál feltételek eltérésváltozóinak, a $\mathbf{z}$ vektornak bázishoz vételével. Az eredeti primál megengedettségi feladat egy megengedett bázisát előállíthatjuk az MBU-szimplex algoritmus, illetve a criss-cross algoritmus megfelelő variánsainak felhasználásával is [2, 7, 21].

1.1. Definíció. Legyen adott egy $(BLCP)$ feladat. A lineáris komplementaritási feladat egy bázisát *majdnem komplementáris*nak nevezzük, ha egyetlen indexpár kivételével teljesülnek a komplementaritási feltételek, vagyis létezik olyan $i \in \{1 \dots 2n\}$, hogy $u_j v_j = 0$ minden $j \in \{1 \dots 2n\} - \{i\}$ esetén.

Két komplementáris bázis között a kvadratikus primál szimplex algoritmus tetszőleges számú majdnem komplementáris bázist generálhat.

1.2. Definíció. Legyen adott egy $(BLCP)$ feladat. A kvadratikus primál szimplex algorimus által végzett, két komplementáris bázis közötti majdnem komplementáris báziscserék sorozatát *huroknak* fogjuk nevezni.

A kvadratikus primál szimplex algoritmus egy ciklusa egy tetszőleges primál megengedett, komplementáris bázissal indul. Egy ilyen bázis esetén, ha minden duál változó is megengedett, úgy az algoritmus a Karush–Kuhn–Tucker-tétel értelmében megtalálta az eredeti feladat egy optimális megoldását. Amennyiben létezik nem megengedett duál változó, úgy a kvadratikus primál szimplex algorimtus választ egy tetszőleges nem megengedett, duál bázis változót. A kvadratikus primál szimplex algoritmus végességét ciklizálás ellenes indexválasztási szabályokkal biztosítjuk.

1.3. Definíció. A kvadratikus primál szimpex algoritmus során a komplementáris táblán választott duál változót *duál vezérváltozó*nak, vagy egyszerűen *vezérváltozó*nak nevezzük.

A duál vezérváltozó kiválasztása után az algoritmus egy hurok során addig végez báziscseréket, míg a választott duál változó ki nem kerül a bázisból, vagy az algoritmus egy végtelen javító irányt nem talál.

A duál vezérváltozó választása után – tehát komplementáris bázisból indulva – a belépő változó a vezérváltozó primál párja. Ezen változó egy javító irányt határoz meg [29]. A primál megengedettség fenntartása érdekében az algoritmus a transzformált oszlopon a primál bázis változókon hányadostesztet végez,

$$\theta_2 = min\left\{\frac{\bar{q}_s}{t_{sj}}|s \in \mathcal{I}_B,\ s \text{ primál változó melyre } t_{sj} > 0\right\}. \tag{19}$$

Ez az az érték, amellyel a választott primál változót növelni lehet, mielőtt a lépés egy korlátozó feltételbe ütközne.

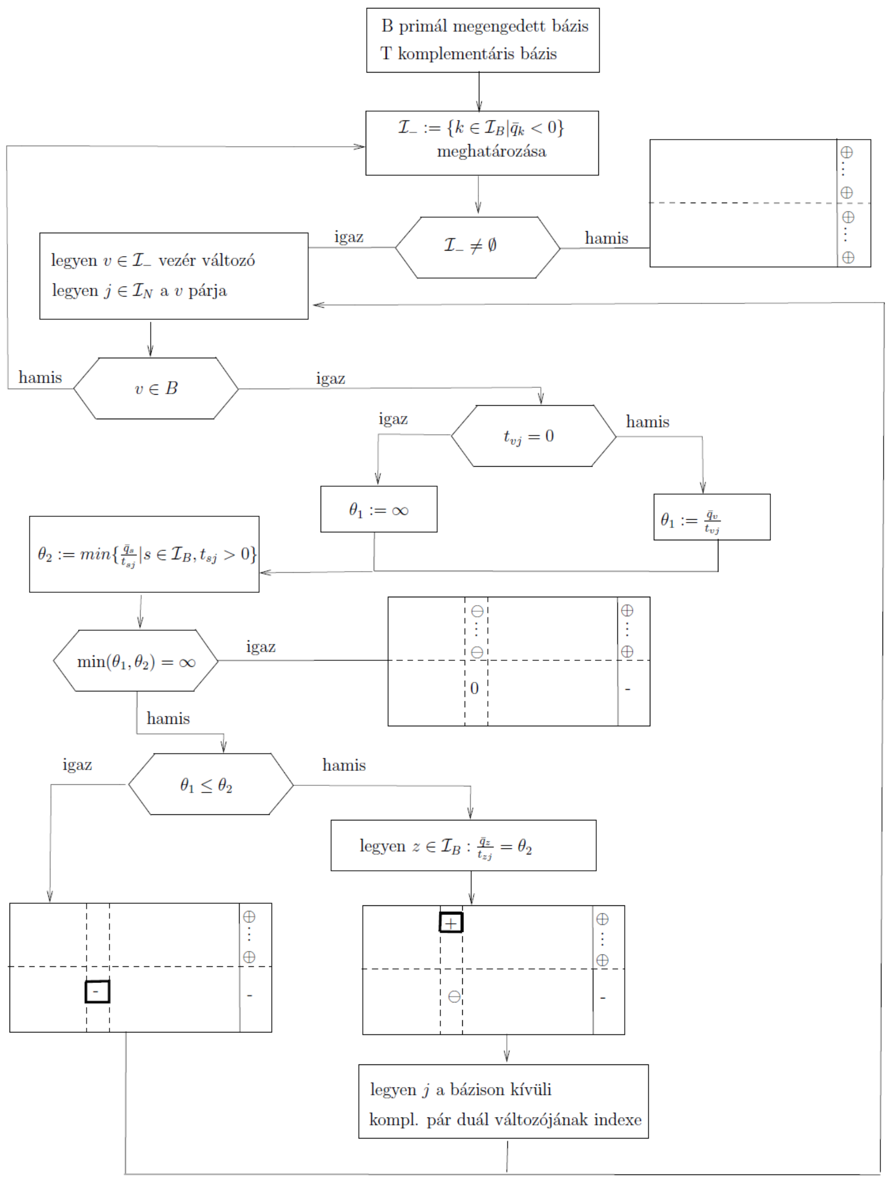
Az algoritmus kiszámít egy θ_1 hányadost is, ami a hányadosteszt értéke a vezérváltozó sorában, $\theta_1 := \frac{\bar{q}_v}{t_{vj}}$; illetve $\theta_1 = \infty$ akkor, ha a vezérváltozó és a hozzá tartozó primál változó találkozásánál nulla szerepel. A θ_1 érték képviseli azt a lépéshosszt, ahol a kvadratikus célfüggvény előjelet vált a belépő változó növelése során.

Abban az esetben, ha $\theta_1 = \theta_2 = +\infty$, akkor a feladat nem korlátos [29].

Amennyiben $\theta_1 \leq \theta_2$, úgy az algoritmus a duál vezérváltozó sorában végez báziscserét, a bázisban levő primál változók száma eggyel növekszik, az új bázis továbbra is megengedett, így az algoritmus egy 1 hosszú hurokkal lezárja a ciklust, a célfüggvény javul.

Amennyiben $\theta_2 < \theta_1$ úgy az algoritmus a primál hányadosteszt által minimálisnak talált hányados sorában végez báziscserét. Ha a hányadosteszt nem jelöli ki egyértelműen a pivot pozíciót, akkor alkalmazzunk ciklizálás ellenes indexválasztási szabályt. Az így keletkezett bázis majdnem komplementáris.

Egy majdnem megengedett bázis esetén az algoritmus bejövő változónak a bázison kívül levő nem komplementáris pár *duál* változóját választja, majd ugyanazon módon választ sort, mint a komplementáris bázis esetén: a primál részen végzett hányadostesztet hasonlítja a vezérváltozó sorának hányadosteszt értékével. Amennyiben a báziscsere a vezérváltozó sorában végezhető, úgy a kapott bázis ismét komplementáris lesz, és a hurok lezárul [29].



1. ábra. Az algoritmus folyamatábrája.

A $(BLCP)$ feladatra megfogalmazott kvadratikus, primál szimplex algoritmus folyamatábrája az 1. ábrán, míg pszeudókódja a 1.1. ábrán található.

1.1. Algoritmus. Az algoritmus pszeudókódja.

bemenő adatok:
A B primál megengedett bázishoz tartozó T komplementáris bázis

begin
$\mathcal{I}_- := \{k \in \mathcal{I}_B^d | \bar{q}_k < 0\}$ a negatív duál változók indexe a bázisban;
while $(\mathcal{I}_- \neq \emptyset)$ **do**
legyen $v \in \mathcal{I}_-$ tetszőleges vezérváltozó;
legyen $j \in \mathcal{I}_N^p$ a v komplementáris párja (primál, bázison kívüli);
while (v a bázisban van) **do**
if $(t_{vj} = 0)$
then $\theta_1 := \infty$
else $\theta_1 := \frac{\bar{q}_v}{t_{vj}}$
endif
$\theta_2 := min\{\frac{\bar{q}_s}{t_{sj}} | s \in \mathcal{I}_B^p,\ s$ primál változó melyre $t_{sj} > 0\}$
if $(\min(\theta_1, \theta_2) = \infty)$ **then** STOP: nem korlátos feladat **endif**
if $(\theta_1 \leq \theta_2)$
then pivotálás a t_{vj} elemen
else
legyen $z \in \mathcal{I}_B^p$ úgy hogy $\frac{\bar{q}_z}{t_{zj}} = \theta_2$
pivotálás a t_{zj} elemen
a bázison kívül pontosan egy komplementáris pár van
legyen j ezen pár duál változójának indexe, az új belépő változó
endif
endwhile
$\mathcal{I}_- := \{k \in \mathcal{I}_B^d | \bar{q}_k < 0\}$
endwhile
optimális megoldásnál vagyunk;
end

1.4. Egy példa

Az algoritmus összetettsége indokolttá tesz egy példát. A példa segítségével szeretnénk egy másik tulajdonságra is felhívni a figyelmet: nevezetesen, hogy legalábbis bázistáblán számolva egy feladatot, az első fázis technikailag nem egyszerű, hiszen elkerülendő a kiegészített bázistábla invertálás útján való kiszámítását, már a primál szimplex első fázisát is a KKT-rendszeren végezzük el.

Tekintsük tehát a következő egyszerű feladatot:

$$\begin{aligned} \min x_1^2 + x_3^2 - 2x_1x_3 \\ x_1 - x_2 + x_3 = 1 \\ x_1 + x_2 = 2 \end{aligned}$$

A KKT-rendszerhez vezessük be a következő váltózó párokat: x primál változó párja a z redukált költséget jelölő változó, mely egyben a duál sorok eltérésváltozója, s eltérésváltozó párja az y duál változók.

A kezdeti (rövid) pivot tábla az eltérésváltozókból álló bázisból indulva:

1.	x_1	x_2	x_3	y_1	y_2	
s_1^*	1	-1	**1**	0	0	1
s_2^*	1	1	0	0	0	2
z_1	-1	0	1	-1	-1	0
z_2	0	0	0	1	-1	0
z_3	1	0	-1	-1	0	0

Az első báziscserét a primál szimplex módszer első fázis célfügvényét követve az x_3 oszlopában végezzük. Ekkor a hányadostesztet csupán az első két sor szerint hajtjuk végre, azonban a tábla komplementaritását megőrzendő a komplementáris pozícióban is végrehajtunk egy báziscserét. Vegyük észre, hogy ezen „második” báziscserék a primál megengedettséget nem befolyásolják, hiszen mindaddig, amíg nem végzünk báziscserét a tábla duál soraiban és a primál változóinak metszeténél, addig duál oszlopok primál sor részében egy azonosan nulla mátrix áll. Az első két báziscsere tehát az x_3 és s_1^*, illetve az y_1 és z_3 párból áll.

2.	x_1	x_2	s_1^*	y_1	y_2	
x_3	1	-1	1	0	0	1
s_2^*	1	1	0	0	0	2
z_1	-2	1	-1	-1	-1	-1
z_2	0	0	0	1	-1	0
z_3	2	-1	1	$\mathbf{-1}$	0	1

3.	x_1	x_2	s_1^*	z_3	y_2	
x_3	1	-1	1	0	0	1
s_2^*	1	**1**	0	0	0	2
z_1	-4	2	-2	-1	-1	-2
z_2	2	-1	1	1	-1	1
y_1	-2	1	-1	-1	0	-1

Továbbra is az első fázist követve, a következő báziscsere pár az x_2 és s_2^* báziscsere, illetve az y_2 és z_2 báziscsere.

4.	x_1	s_2^*	s_1^*	y_1	y_2	
x_3	2	1	1	0	0	3
x_2	1	1	0	0	0	2
z_1	-6	-2	-2	-1	-1	-6
z_2	3	0	1	1	$\mathbf{-1}$	3
z_3	-3	-1	-1	-1	0	-3

5.	x_1	s_2^*	s_1^*	y_1	z_2	
x_3	2	1	1	0	0	3
x_2	1	1	0	0	0	2
z_1	$\mathbf{-9}$	-3	-3	-2	-1	-9
y_2	-3	-1	-1	-1	-1	-3
y_1	-3	-1	-1	-1	0	-3

Az 5. tábla már primál megengedett. Mindhárom duál sor duál nem megengedett, vagyis mindhárom bázison kívüli primál változó javító irányt határoz meg. Mivel azonban s_1^* és s_2^* mesterséges eltérésváltozók, így azok nem térhetnek vissza a bázisba (ezeket el lehetne hagyni a táblából). Így a bejövő változó az x_1. A kiegészített hányadosteszt alapján diagonális báziscserét végzünk z_1 sorában.

6.	z_1	s_2^*	s_1^*	y_1	z_2	
x_3	$\frac{2}{9}$	$\frac{1}{3}$	$\frac{1}{3}$	$-\frac{4}{9}$	$-\frac{2}{9}$	1
x_2	$\frac{1}{9}$	$\frac{2}{3}$	$-\frac{1}{3}$	$-\frac{2}{9}$	$-\frac{1}{9}$	1
x_1	$-\frac{1}{9}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{2}{9}$	$\frac{1}{9}$	1
y_2	$-\frac{1}{3}$	0	0	$-\frac{1}{3}$	$-\frac{2}{3}$	0
y_1	$-\frac{1}{3}$	0	0	$-\frac{1}{3}$	$\frac{1}{3}$	0

A tábla megengedett és optimális. Az optimális megoldás az azonosan 1, azaz $x = y = z = 1$.

2. A kvadratikus primál szimplex algoritmus végessége

Az 1. ábrán és 1.1. algoritmusban bemutatott kvadratikus szimplex algoritmusról számos cikk íródott az 1960-as évek elején [26, 28, 29, 30, 31, 36]. Az algoritmus végességét eredetileg a perturbációs módszerrel igazolták. Ebben a fejezetben a kvadratikus szimplex algoritmus új bizonyítását adjuk ciklizálás ellenes indexválasztási szabály segítségével.

Felidézzük az úgynevezett $\mathbf{s}$-monoton index választási szabályokat [2]:

2.1. Definíció. Legyen adott egy index választáson alapuló pivotálási szabály, egy $\mathbf{s} \in \mathbb{N}_{\oplus}^n$ vektor, amelynek a koordinátáit a feladat változóihoz rendeltük, és az algoritmus iterációi során a pivotálási szabálytól függően módosulhatnak. A pivotálási szabálytól függő $\mathbf{s}$ vektor sorozatra az alábbi elvárásokat fogalmazzuk meg:

1. Az $\mathbf{s}$ vektor értékei a báziscserék során nem csökkennek, illetve kizárólag a mozgó változók értéke változhat. Az index választási szabály, választási lehetőség esetén, az $\mathbf{s}$ vektor szerinti maximális értékű elemei közül választ.

2. A algoritmus során bármikor mozgó (vagyis bázisból kilépő vagy belépő) változókra megszorítva, van olyan B^* bázis, amikor az $\mathbf{s}$ vektor szerinti legkisebb értékű bázison kívüli változó egyértelmű. Legyen ez az x_l változó.
3. Ha a B^* bázis után az x_l változó belép a bázisba, akkor a legközelebbi belépés után, egészen addig, amíg esetleg az x_l változó újra távozik a bázisból, igaz, hogy azon változók $\mathbf{s}$ értéke, amelyek mozogtak az x_l változó bázisba belépése óta, nagyobbak, mint az x_l változó $\mathbf{s}$ vektor szerinti értéke.

Azokat a pivotálási szabályokat, amelyekhez tartozó $\mathbf{s}$ vektorokra az 1–3. feltételek teljesülnek $\mathbf{s}$-*monoton* pivotálási szabályoknak nevezzük.

Az $\mathbf{s}$-monoton indexválasztási szabályok alkalmazásra kerültek kvadratikus criss-cross algoritmus végességének igazolásakor [1, 7, 9].

Dolgozatunk szempontjából a fő észrevétel, hogy az $\mathbf{s}$-monoton indexválasztási szabályok nem egyértelmű választás esetén egy, az adott pillanatban jól definiált preferencia vektor szerint választanak, nem használva a bázistábla elemeinek a konkrét értékeit.

Bizonyításunk az algoritmus és a lineáris feltételes konvex kvadratikus programozási feladat pivot táblájának tulajdonságainak vizsgálatán alapul. Általános esetben, sajnos a biszimmetrikus tulajdonság nem örződik meg közvetlen módon, de egy kis kiegészítéssel hasonló tulajdonság bizonyítható.

2.1. Lemma. *[33] Egy kvadratikus programozáshoz tartozó biszimmetrikus mátrix esetén tetszőleges bázis transzformációval nyert, komplementáris bázis esetén a bázistábla továbbra is biszimmetrikus azzal a kivétellel, hogy az eredeti primál feltételek és a duál változók metszetében levő nulla mátrix helyén egy pozitív szemidefinit mátrix áll.*

A fenti eredmény kivétel része elkerülhető, ha a kvadratikus feladat egy szimmetrikus felírását alkalmazzuk [21].

A végesség bizonyítását visszavezetéssel végezzük. Bizonyítjuk, hogy egy ciklizáló példa esetén a primál megoldás szükségszerűen nem változik, vagyis minden báziscsere primál degenerált. Szemléletes módon, ez azt jelenti hogy az adott megoldáshoz tartozó linearizált feladat változatlan marad. Megmutatjuk, hogy ilyen esetben az algoritmus által végzett báziscserék pontosan megfelelnek egy megfelelő lineáris programozási feladatra nézve a primál szimplex algoritmus báziscseréinek, mely indexválasztási szabály alkalmazása esetén véges: szükségképpen, a kvadratikus szimplex algoritmus is véges mindazon ciklizálás elleni indexválasztási szabályok esetén, melyre a primál szimplex az, amennyiben a ciklizálás elleni indexválasztási szabály kizárólag a redukált költségek előjelére és a változók indexével kapcsolatos választási preferenciákra hivatkozik [9]. A lexikografikus szabályra a bizonyításunk nem alkalmazható közvetlen módon. Legjobb tudomásunk szerint, a lexikografikus rendezés alkalmazásával a kvadratikus szimplex algoritmus végességét igazoló eredmény nem ismert.

Tegyük fel tehát, hogy az algoritmus nem véges, és tekintsünk egy ciklizáló ellenpéldát. Mivel a bizonyítás visszavezetésen alapszik, a ciklizáló ellenpélda méretének minimalitása nem szükséges, és nem is egyszerűsítené lényegesen a gondolatmenetet.

Először megmutatjuk, hogy egy ciklizáló ellenpéldán az algoritmus kizárólag egyfajta, mégpedig kettő hosszú hurkokat állít elő.

2.2. Lemma. *Tekintsük a konvex (LKOF) feladatot a hozzá tartozó* $(BLCP)$ *formában. A kvadratikus szimplex algoritmus végrehajtása során, egy ciklizáló példa esetén, legfeljebb véges sokszor fordulhat elő olyan báziscsere, amelyik egy hosszú hurkot végez.*

Bizonyítás. A kvadratikus szimplex algoritmus megfogalmazásából adódik, hogy egy 1 hosszú hurok egyetlen, a duál vezérváltozó sorában végzett báziscseréből áll, ez a $0 < \theta_1 \leq \theta_2$ esetnek felel meg. Mivel a duál vezérváltozót úgy választottuk, hogy a hozzátartozó jobboldal negatív, így ez a báziscsere nem degenerált. Ilyen esetben a belepő primál változó oszlopa egy javító irány és a célfüggvény értéke javul [29]. Mivel az egy hosszú hurkok esetén a célfüggvény javul, ezért a korábbi bázisok egyike sem térhet vissza, hiszen a kvadratikus szimplex algoritmus célfüggvénye monoton csökken. Figyelembe véve, hogy véges sok bázis van, egy hosszú hurok véges sokszor fordulhat elő. □

Most vizsgáljuk meg a kettőnél hosszabb hurkok lehetséges számát.

2.3. Lemma. *Tekintsük a konvex (LKOF) feladatot a hozzá tartozó* $(BLCP)$ *formában. A kvadratikus szimplex algoritmus végrehajtása során, egy ciklizáló példa esetén legfeljebb véges sok nem 2 hosszúságú hurok lehetséges.*

Bizonyítás. A 2.2. lemma alapján legfeljebb véges sok 1 hosszú hurok lehetséges. Figyeljük meg, hogy a bázisban levő primál változók száma legfeljebb az 1 hosszú hurkok esetén növekedhet. Nem 1 hosszú hurok esetén, az első báziscserét követően, egészen addig, amíg nem a duál vezérváltozó sorában végzünk báziscserét, addig a báziscserék során a bejövő duál változó egy primál változót cserél ki a bázisban. Vagyis amennyiben a hurok 3, vagy annál hosszabb, úgy a bázisban levő duál változók száma monoton növekedik. Mivel csökkenni csak 1 hosszú hurkok során tud, melyek száma véges, így szükségképpen a 3, vagy annál hosszabb hurkok száma is véges. □

Összefoglalva megállapíthatjuk, hogy egy ciklizáló példa esetén az algoritmus véges sok báziscsere után 2 hosszú hurkok végtelen sorozatát végzi.

Felvetődik a kérdés hogy nem lenne-e célszerű a bizonyítást a criss-cross algoritmus végességére visszavezetni, hiszen a 2 hosszú hurkok megfelelnek egy-egy felcserélős báziscserének [21, 1, 7]. A nehézséget az okozza, hogy a báziscsere sorának kiválasztása után az oszlopválasztás a kvadratikus szimplex algoritmus során kötött, így a criss-cross algoritmus második index választási lépése elmarad.

2.4. Lemma. *Tekintsük a konvex (LKOF) feladatot a hozzá tartozó ($BLCP$) formában. A kvadratikus szimplex algoritmus végrehajtása során, egy ciklizáló példa esetén legfeljebb véges sok báziscsere után a bázismegoldás primál része nem változik.*

Bizonyítás. Tételezzük fel, hogy az ciklizáló példa során már kizárólag 2 hosszú hurkokat végez az algoritmus a 2.2. és 2.3. lemmák alapján.

Egy 2 hosszú hurok első báziscseréje során egy nem degenerált báziscsere javítaná a célfüggény értékét [29].

A második báziscsere esetén ennél több is mondható. [33] alapján ilyen esetben a belépő duál változó és az előző iterációban a bázisból kilépett primál változó találkozásánál a bázistábla t_{ij} értéke nem-pozitív, és amennyiben szigorúan negatív, úgy a báziscsere javít a célfüggvényértéken – mely esetünkben azt jelenti, ez az eset csak véges sokszor fordulhat elő – illetve amennyiben nulla, úgy a bázistábla ezen oszlopában minden bázisban levő primál változóhoz tartozó érték nulla, vagyis a pivot tábla primál része már nem transzformálódik. □

A fenti bizonyításban szereplő [33] eredményének felhasználásával a következő erősebb lemmát is bizonyíthatjuk:

2.5. Lemma. *Tekintsük a konvex (LKOF) feladatot a hozzá tartozó ($BLCP$) formában. A kvadratikus szimplex algoritmus végrehajtása során, egy ciklizáló példa esetén legfeljebb véges sok báziscsere után az algoritmus csupa 2 hosszú hurkot végez, melyekre a következő igaz:*

- *A hurok első báziscseréje egy degenerált báziscsere, mely során egy primál változó belép, és egy primál változó kilép a bázisból.*
- *A hurok második (és utolsó) báziscseréje során a megelőző iterációban belépett primál változó duál párja kilép, míg a megelőző iterációban kilépett primál változó duál párja belép a bázisba.*
- *A hurok második báziscseréje során a belépő duál változó transformált oszlopában minden primál változóhoz tartozó sorban nulla érték szerepel.*

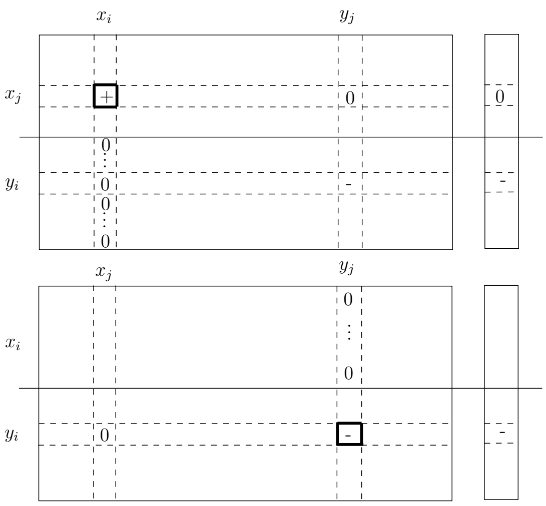
Bizonyítás. Következik a 2.1. – 2.5. lemmákból, a 2.5. lemma bizonyításához hasonló módon [33] eredményéből. □

Hasonló tulajdonság mondható a hurkok első báziscseréjének duál részére is.

2.6. Lemma. *Tekintsük a konvex (LKOF) feladatot a hozzá tartozó ($BLCP$) formában. A kvadratikus szimplex algoritmus végrehajtása során, egy ciklizáló példa esetén legfeljebb véges sok bázis csere után a komplementáris bázisból induló báziscserék esetén a belépő primál változó transzformált oszlopában a kiindulási bázistáblán a duál változókhoz tartozó sorokban nulla értékek szerepelnek.*

Bizonyítás. Felhasználva a 2.1. lemmát, mivel a választott duál vezérváltozó sorának és a hozzá tartozó belépő primál változó ennek a szemidefinit mátrixnak egy diagonális eleme, mely nulla, így szükséges, hogy ennek a szemidefinit mátrixnak ezen oszlopa (és sora) azonosan nulla legyen, hiszen ellenkező esetben nem volna pozitív szemidefinit, hiszen egy tetszőleges nemnulla érték és a diagonális pozíció által alkotott 2×2-es átló menti részmátrix determinánsa negatív lenne. □

A korábbi lemmákkal már bizonyítottuk, hogy a ciklizáló ellenpélda esetén a mozgó változók transzformált oszlopaiban nulla értékek szerepelnek a 2 hosszú hurkok első báziscseréje esetén a duál változók soraiban, míg a második báziscserék esetén a primál változók soraiban. A két bázistábla szerkezetét a 2. ábra mutatja.



2. ábra. Egy ciklizáló példa esetén a ciklizálás beállta után a 2 hosszú hurkok első, illetve második báziscseréjéhez tartozó bázistábla szerkezete.

Tekintsünk egy ciklizáló példát, és tegyük fel a 2.2. és 2.3. lemmák alapján, hogy az algoritmus már csupa 2 hosszú, degenerált hurkokat végez. Egy tetszőleges komplementáris bázis esetén, $\mathcal{I}_B^p$ és $\mathcal{I}_B^d$ rendre jelölje a bázisban levő primál-, illetve duál változók index halmazát, míg az $\mathcal{I}_N^p$ és $\mathcal{I}_N^d$ pedig a nem bázis változók megfelelő indexhalmazait, ahogyan azt korábban bevezettük.

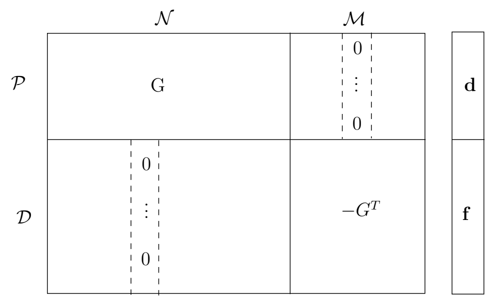
Legyen $G = \bar{M}_{\mathcal{I}_B^p \mathcal{I}_N^p}$ az $\mathcal{I}_B^p$ és $\mathcal{I}_N^p$ indexhalmazok által meghatározott részmátrix, $\mathbf{d} = \bar{\mathbf{q}}_{\mathcal{I}_B^p}$ a $\mathcal{I}_B^p$ halmaz elemeihez tartozó jobboldal, míg $\mathbf{f} = \bar{\mathbf{q}}_{\mathcal{I}_B^d}$ a $\mathcal{I}_B^d$ halmaz elemeihez tartozó jobboldal.

Ekkor a 2.1. lemma alapján $G = \bar{M}_{\mathcal{I}_B^p \mathcal{I}_N^p} = -\bar{M}_{\mathcal{I}_N^d \mathcal{I}_B^d}$. Látható, hogy az $\bar{M}$ transzformált bázis tábla megegyezik a

$$\begin{aligned} &\min \mathbf{f}^T \mathbf{x} \\ &G\mathbf{x} \leq \mathbf{d} \end{aligned} \qquad (LP_s)$$

peciális lineáris programozási feladatra (LP_s) felírt Karush–Kuhn–Tucker-feltételekkel, amennyiben a feladat azon részétől, mely nem játszik szerepet a ciklizálásban, eltekintünk.

Felhasználva a 2.6., 2.5. és 2.1. lemmákat, látható, hogy a bázistábla ugyanolyan módon transzformálódik a 2 hosszú hurkok során, mint ahogy az előző lineáris programozási feladat bázistáblája. Továbbá a duál vezérváltozó választása megfelel a célfüggvény sorában levő oszlopválasztásnak, majd a primál változók feletti hányadosteszt megfelel a lineáris programozási feladatra megfogalmazott primál szimplex módszer hányadostesztjének a kisebb méretű lineáris programozási feladat esetén.



3. ábra. A kvadratikus programozási feladat bázis táblájának szerkezete a ciklizáló változókra nézve.

Tehát a $(BLCP)$ feladatra megfogalmazott kvadratikus primál szimplex módszer pontosan akkor ciklizálhat, ha a lineáris programozási feladtra megfogalmazott primál szimplex algoritmus ciklizál az (LP_s) lineáris programozási feladaton.

Figyelembe véve, hogy a lineáris programozási feladatra megfogalmazott primál szimplex algoritmus nem ciklizálhat, ha olyan index választási szabályt használunk a végesség biztosítására, amelyik az ún. s-monoton indexválasztási szabályok (pl. minimál index szabály, LIFO- vagy a leggyakrabban választott változó szabálya) közé tartozik [9].

A $(BLCP)$ feladatra megfogalmazott kvadratikus primál szimplex módszert el kell látnunk ciklizálás ellenes indexválasztási szabállyal, amely biztosítja az algoritmus végességét. Összefoglalva, kimondhatjuk a következő tételt:

2.1. Tétel. *A $(BLCP)$ feladatra megfogalmazott kvadratikus primál szimplex módszer, s-monoton indexválasztási szabályok használata esetén véges.*

Megmutattuk, hogy a kvadratikus primál szimplex algoritmus véges az s-monoton indexválasztási szabályok alkalmazása esetén. Eredményünk közvetlen átültethető a kvadratikus duál szimplex algoritmusra is.

Köszönetnyilvánítás. A kutatást a TÁMOP-4.2.2./B-10/1-2010-0009 pályázattal a Nemzeti Innovációs Hivatal jogelődje, a Nemzeti Kutatási és Technológiai Hivatal támogatta.

Illés Tibor kutatásait a Strathclyde University, Glasgow a *John Anderson Research Leadership Program* keretében támogatta.

Hivatkozások

[1] Akkeles, A. A., Balogh L. és Illés T.: *A véges criss-cross módszer új variánsai biszimmetrikus lineáris komplementaritási feladatra.* Alkalmazott Matematikai Lapok **21**, 1–25, (2003).

[2] Bilen, F., Csizmadia, Z., Illés, T.: *Anstreicher-Terlaky típusú monoton szimplex algoritmusok megengedettségi feladatokra.* Alkalmazott Matematikai Lapok **24 (1-2)**, 163–185, (2007).

[3] R. W. Cottle, G. B. Dantzig: *Complementarity Pivot Theory of Mathematical Programming.* Linear Algebra and its Applications **1**, 103–125, (1968).

[4] R. W. Cottle, J.-S. Pang, V. Venkateswaran: *Sufficient matrices and the linear complementarity problem.* Linear Algebra and its Applications **114-115**, 231–249, (1989).

[5] S.-M. Guu, R. W. Cottle: *On a subclass of P^*.* Linear Algebra and its Applications **223/224**, 325–335, (1995).

[6] Z. Csizmadia, T. Illés: *New criss-cross type algorithms for linear complementarity problems with sufficient matrices.* Optimization Methods and Software Vol. **21** No. **2**, 247–266, (2006).

[7] Z. Csizmadia: *New pivot based methods in linear optimization, and an application in petroleum industry.* PhD Thesis, Eötvös Loránd University of Sciences, Budapest, 2007.

[8] Z. Csizmadia, T. Illés, A. Nagy: *The s-monotone index selection rules for pivot algorithms of linear programming.* European Journal of Operation Research **221**, 491–500, (2012).

[9] Z. Csizmadia, T. Illés, A. Nagy: *The s-Monotone Index Selection Rule for Criss-Cross Algorithms of Linear Complementarity Problems.* Acta Universitatis Sapientiae - Informatica Vol. **5** No. **1**, 103–139, (2013).

[10] K. Fukuda, M. Namiki, A. Tamura: *EP theorems and linear complementarity problems.* Discrete Applied Mathematics Vol. **84** No. **1–3**, 107–119, (1998).

[11] R. W. Cottle, S.-M. Guu: *Two characterizations of sufficient matrices.* Linear Algebra and its Applications **170**, 65–74, (1992).

[12] D. den Hertog, C. Roos, T. Terlaky: *The linear complimentarity problem, sufficient matrices, and the criss-cross method.* Linear Algebra and its Applications **187**, 1–14, (1993).

[13] T. Illés, C. Roos, T. Terlaky: *Polynomial affine-scaling algorithms for $P^*(\kappa)$ linear complementary problems.* Lecture Notes in Economics and Mathematical Systems **452**, pp. 119–137, (1997).

[14] T. Illés, JM. Peng, C. Roos, T. Terlaky: *A strongly polynomial rounding procedure yielding a maximally complementary solution for $P^*(\kappa)$ linear complementarity problems.* SIAM Journal on Optimization **11**, 320–340, (2000).

[15] T. Illés, T. Terlaky: *Pivot versus interior point methods: Pros and cons.* European Journal of Operation Research **140**, 170–190, (2002).

[16] Illés T., és Nagy M.: *Mizuno–Todd–Ye típusú prediktor–korrektor algoritmus elégséges mátrixú lineáris komplementaritási feladatokra.* Alkalmazott Matematikai Lapok **22**, 41–46, (2005).

[17] T. Illés, M. Nagy, T. Terlaky: *A polynomial path-following interior point algorithm for general linear complementarity problems.* Journal of Global Optimization **47**, 329–342, (2010).

[18] T. Illés, M. Nagy, T. Terlaky: *Polynomial Interior Point Algorithms for General Linear Complementarity Problems.* Algorithmic Operations Research **5**, 1–12, (2010).

[19] Illés T.: *Lineáris optimalizálás elmélete és pivot algoritmusai.* Operations Research Reports, 2013-03. http://www.cs.elte.hu/opres/orr/download/IT-LP-pivot-jegyzet-20131031.pdf

[20] Klafszky E., Terlaky T.: *A pivot technika szerepe a lineáris algebra néhány alapvető tételének bizonyításában.* Alkalmazott Matematikai Lapok **14**, 425–448, (1989).

[21] E. Klafszky, T. Terlaky: *Some Generalizations of the Criss-Cross Method for Quadratic Programming.* Math. Oper. und Stat. ser. Optimization **24**, 127–139, (1990).

[22] E. de Klerk, C. Roos and T. Terlaky: *Nemlineáris Optimalizálás.* Operációkutatás No. **5**., Budapesti Közgazdaságtudományi és Államigazgatási Egyetem, Operációkutatás Tanszék, Budapest (2004).

[23] M. Kojima, S. Mizuno, A. Yoshise: *A primal-dual interior point algorithm for linear programming.* in: N. Megiddo, ed., *Progress in Mathematical Programming, Interior-Point and Related Methods*, Springer, New York, pp. 29–48, (1988).

[24] M Kojima, S. Mizuno, A. Yoshise: *A polynomial-time algorithm for a class of linear complementarity problems.* Mathematical Programming **44**, 1–26, (1989).

[25] M. Kojima, N. Megiddo, T. Noma, A. Yoshise: *A unified approach to interior point algorithms for linear complementarity problems.* Lecture Notes in Computer Science **538**, Springer-Verlag, (1991).

[26] C. E. Lemke, J. T. Howson, Jr.: *On complementary pivot theory.* Mathematics of decision sciences, Volume Part **1**, 95–114.

[27] M. Nagy: *Interior point algorithms for general linear complementarity problems.* PhD Thesis, Eötvös Loránd University of Sciences, Budapest, (2009).

[28] C. van de Panne, A. Whinston: *A Parametric Simplicial Formulation of Houthakker's Capacity Method.* Econometrica, Vol.**34**, No. **2**, 354–380, (1966).

[29] C. van de Panne, A. Whinston: *Simplicial methods for quadratic programming.* Naval Research Logistics, Vol. **11**, 273–302, (1964).

[30] C. van de Panne, A. Whinston: *The Simplex and the Dual Method for Quadratic Programming.* Operational Research Quarterly, Vol. **15**, 355–388, (1964).

[31] C. van de Panne, A. Whinston: *The Symmetric Formulation of the Simplex Method for Quadratic Programming.* Econometrica, Vol. **37**, No. **3**, 507–527, (1969).

[32] T. Terlaky: *A new algorithm for quadratic programming.* European Journal of Operational Research, Vol. **32**, **2**: 294–301, (1987).

[33] A. W. Tucker: *Principal pivotal transformations of square matrices.* SIAM Review, pp. 305, (1963).

[34] H. Väliaho: *A new proof for the criss-cross method for quadratic programming.* Optimization, Vol. **25**, No. **4**, 391–400, (1992).

[35] H. Väliaho: *P*-Matrices Are Just Sufficient.* Linear Algebra and its Applications **239**, 103–108, (1996).

[36] P. Wolfe: *The Simplex Method for Quadratic Programming.* Econometrica Vol. **27**, No. **3**, 382–398, (1959).

(Beérkezett: 2013. december 7.)

ILLÉS TIBOR
BME Differenciálegyenletek Tanszék
1111 Budapest, Egry József utca 1, H ép. IV. em.
illes@math.bme.hu

NAGY ADRIENN
FICO
B37 7GN, Birmingham, Starley Way, United Kingdom
adriennagy@gmail.com

FINITENESS OF THE QUADRATIC SIMPLEX METHOD WITH THE APPLICATION OF INDEX SELECTION RULES

Tibor Illés, Adrienn Nagy

We provide a new proof for the finiteness of the primal simplex method for linearly constrained convex quadratic programming problems when using index selection rules. The original

quadratic simplex algorithm was developed by Wolfe and van de Panne and Whintson, and have been published in a series of papers in the 1960s, using perturbation techniques to ensure finiteness.

We show that for the method to cycle, the pivots and the problem needs to be degenerate; i.e. the value of all the variables -in the corresponding pivot tableau of the Karush-Kuhn-Tucker system- taking part of the primal ratio test needs to be zero, but moreover, the the value of the entries in the transformed pivot columns that correspond to the quadratic objective must be zero.

It follows that the quadratic primal simplex method is finite for any index selection rule that only relies on the sign structure of the transformed right hand side and of the reduced costs, and for which the corresponding traditional primal simplex method is finite for linear programming problems.