

Neumann projektív geometrián alapuló, és Karmarkar projektív skálázású belsőpontos módszere

Összefoglalás: Neumann János a 20. század egyik legkiemelkedőbb matematikusa volt, aki összekapcsolta a tiszta és alkalmazott tudományokat. Jelentős szerepet játszott a matematika, fizika, közgazdaságtan és számítástechnika fejlődésében. Különösen fontos munkát végzett a lineáris programozás terén, amely alapvető jelentőségűvé vált a matematikai és gazdasági tervezésben. Neumann egyik úttörő eredménye a Paul Gordan homogén lineáris rendszerén alapuló új lineáris programozási módszer volt, amelyet később Karmarkar algoritmusá tett széles körben ismertté. Ez az algoritmus a lineáris programozás első belsőpontos módszere volt. A cikk ezeket az algoritmusokat ismerteti.

Kulcsszavak: Lineáris programozás; belsőpontos algoritmus; Karmarkar; Neumann János.

Abstract: John von Neumann was one of the most significant mathematicians of the 20th century, who built a bridge between pure and applied sciences. He had a tremendous impact on the development of mathematics, physics, economics, and computer science. He made outstanding contributions to the field of linear programming, which became fundamental in mathematical and economic planning. One of von Neumann's pioneering achievements was a method based on projective geometry, designed to solve the gravity center problem, which later became widely known through Karmarkar's algorithm. This algorithm was a precursor to the interior-point methods in linear programming. The article discusses these algorithms.

Keywords: Linear programming; interior point algorithm; Karmarkar; John Neumann.

* Dunaiújvárosi Egyetem, Informatikai Intézet, Matematika és Számítástudományi Tanszék
E-mail: pappz@uniduna.hu

[1] Iványi, A. (2005):
Informatikai algorit-
musok II, pp. 1–768.
 Budapest: ELTE
 Eötvös Kiadó.

Bevezetés

A lineáris programozás (LP) egy matematikai módszer az optimalizálásra, amelynek célja egy lineáris célfüggvény maximalizálása vagy minimalizálása adott lineáris korlátok mellett. Az LP-modellek széles körben alkalmazhatók, többek között az iparban, a közgazdaságtanban, a logisztikában és a pénzügyekben. Egy tipikus lineáris programozási probléma a következő formában fogalmazható meg:

$$\min_x c^T x, \text{ feltéve, hogy } Ax = b, x \geq 0,$$

ahol $x \in R^n$ a döntési változók vektora, $c \in R^n$ a célfüggvény együtthatóinak vektora, $A \in R^{m \times n}$ a korlátmátrix, $b \in R^m$ a korlátvektor. A lineáris programozás több mint egy évszázados múltra tekint vissza, és egyik korai kiemelkedő eredménye Farkas Gyula híres tétele, amelyet 1894-ben publikált. Az idők során a lineáris programozás fejlődése gyakran gyakorlati problémák megoldásából indult ki, amelyek új algoritmusok fejlesztésére és elméleti eredmények megfogalmazására ösztönözték a kutatókat. George Dantzig 1947-ben fejlesztette ki a lineáris feladat megoldására szolgáló szimplex-módszert. Khacijan 1979-ben mutatta be az ellipszoid módszert, amely abban az időben a legkorszerűbb, polinomiális időbeli komplexitású algoritmusnak számított, bár gyakorlati alkalmazhatósága elmaradt a szimplex-módszer mögött. 1984-ben Karmarkar közzétette saját projektív skálázású algoritmusát, amely jelentős előrelépést hozott a belsőpontos módszerek fejlődésében [1]. A belsőpontos algoritmusok olyan módszerek, amelyek a lineáris programozás megoldására egy belső, megvalósítható pontból kiindulva dolgoznak, és lépéseikkel a belső térben haladnak, szemben a hagyományos szimplex algoritmussal, amely a megvalósítható tartomány határán haladva jut el a megoldáshoz. Az első belsőpontos módszerek módszerek osztályába tartozik Neumann János algoritmus. Az általa javasolt eljárás egy konvexitási feltétellel rendelkező lineáris programozási feladat megoldására szolgált, és egyszerűsége, valamint gyors konvergenciája miatt figyelemre méltó. Mivel egy általános lineáris programozási feladat és annak duálisa egy ilyen típusú megvalósíthatósági problémává alakítható át, Neumann algoritmusát egy általános lineáris programozási módszernek is tekinthetjük. Neumann János a 20. század egyik legkiemelkedőbb matematikusa volt, aki összekapcsolta a tiszta és alkalmazott tudományokat, és jelentős szerepet játszott a matematika, fizika, közgazdaságtan és számítástechnika fejlődésében. A tanulmányban a szerző bemutatja és elemzi Neumann János, a gravitációs feladat megoldására szolgáló belsőpontos módszerét,

valamint Karmarkar algoritmusát. Bár mindkét algoritmus a maga idejében úttörő jelentőségű volt, azóta számos hatékonyabb és a gyakorlatban jobban alkalmazható belsőpontos módszert fejlesztettek ki. Ennek következtében a két algoritmus ma már elsősorban történelmi szempontból bír jelentőséggel. A tanulmány szerkezete a következőképpen alakul: a második fejezet Neumann János belsőpontos algoritmusát tárgyalja, míg a harmadik fejezet Karmarkar úttörő belsőpontos módszerét mutatja be. A negyedik fejezetben a két algoritmus hatékonyságát különböző teszt-feladatok segítségével vizsgálja a szerző.

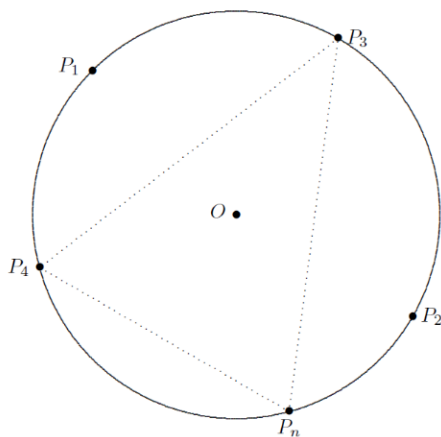
[2] Dantzig, G. B.–Thapa, M. N. (2006): *Linear Programming 2*. New York: Springer Science & Business Media.

Neumann János belsőpontos algoritmus

1948-ban Neumann János kapcsolatba lépett George B. Dantziggel, hogy megvitassák a következő gravitációs középpont feladatot. [2]

„Adott n darab $P_j \in R^m$ pont, amelyek egy egységnyi sugarú, m -dimenziós gömb S felületén helyezkednek el. A gömb középpontja az origóban van. Keressük meg a nemnegatív $x_j = x_j^*$ súlyokat, amelyeket a P_j pontokhoz rendelünk úgy, hogy a súlyozott tömegközéppontjuk az origó legyen, vagy bizonyítsuk be, hogy nincs ilyen súlyozás.”

1. ábra. Gravitációs középpont feladat



A gravitációs feladatban a cél meghatározni egy olyan x vektor n elemét, melyre érvényes:

$$x \geq 0, \sum_{j=1}^n P_j x_j = 0, \sum_{j=1}^n x_j = 1, ||P_j||_2 = 1, j = 1, \dots, n, \quad (1)$$

ahol $P_j \in \mathbb{R}^m$. Az (1) feladatot tekinthetjük egy LP megszorításainak. Az (1) feladat felírható mátrix alakban:

$$Px = 0, e^T x = 1, x \geq 0, \quad (2)$$

Ahol $P \in \mathbb{R}^{m \times n}$ oszlopai tartalmazzak a $P_j, j=1, 2, \dots, n$ pontok koordinátáit, $x \in \mathbb{R}^n$ a súlyok vektora és e egy n dimenziós vektor, mely összes eleme 1.

Neumann János bemutatta a gravitációs feladat megoldására szolgáló megoldási módszert Dantzignek, bár annak konvergenciáját nem bizonyította. Később, levelezésük során Dantzig igazolta a módszer konvergenciáját.

Neumann János algoritmus a gravitációs középpont-feladat megoldására (Goncalves 2004):

1. **Inicializáció:** Az algoritmus bármilyen, az origóhoz közeli közelítéssel inicializálható, vagyis $x^0 \in \mathbb{R}^n$ tetszőlegesen választható úgy, hogy $x^0 \geq 0$ és $e^T x^0 = 1$. Ekkor legyen $b^0 = Px^0$.
2. **Keresési irány kiszámítása:** Tegyük fel, hogy a k -adik iteráció kezdeténél, ahol $k \geq 1$, x^{k-1} is mert a $k-1$ -edik iterációból, melyre érvényes: $x^{k-1} \geq 0$ és $e^T x^{k-1} = 1$.

Legyen

$$b^{k-1} = Px^{k-1}, u_{k-1} = ||b^{k-1}||.$$

Az összes P_j vektor közül keresd meg azt a P_s vektort, amely a legnagyobb szöget zárja be a b^{k-1} vektorral, vagyis a keresett P vektor indexe.

3. Megoldhatatlanság ellenőrzése: Legyen

$$s = \arg \min_{j=1,2,\dots,n} P_j^T b^{k-1}.$$

a (2) feladatnak nincs megoldása, mivel az összes P_j pont a b^{k-1} vektorra merőleges és az origón áthaladó hipersík egy oldalán található. Ekkor nem létezik a P_j pontok olyan konvex kombinációja, melyek gravitációs középpontja az origó lesz. Tehát, a feladat nem megoldható.

$$v_{k-1} = P_s^T b^{k-1}. \text{ Ha } v_{k-1} > 0, \text{ STOP,}$$

4. Az új approximáció kiszámítása: Az új b^k approximáció a b^{k-1}

és P_s -t összekötő szakasz olyan pontja lesz, amely legközelebb van az origóhoz. Lásd a 2. ábrát. A konvex kombináció λ súlyozási tényezőjét, a b^k következő approximációt, a b^k kapproximáció u_k k távolságát az origótól és a gravitációs feladat súlyainak approximációját a következő képletekkel határozzuk meg:

$$\lambda = \frac{1 - v_{k-1}}{u_{k-1}^2 - 2v_{k-1} + 1}$$

$$b^k = \lambda b^{k-1} + (1 - \lambda)P_s$$

$$u_k^2 = \lambda v_{k-1} + (1 - \lambda)$$

$$x^k = \lambda x^{k-1} + (1 - \lambda)e_s,$$

ahol e_s egységvektor, melynek az s -edik eleme 1.

Legyen $k := k+1$ és menj a 2. lépésre.

[3] Dantzig, G. B. (1991): *Converting a Converging Algorithm into a Polynomial Bounded Algorithm*, pp. 1–9.

[4] Dantzig, G. B. (1992): An epsilon-Precise Feasible Solution to a Linear Program with a Convexity Constraint In: *1/epsilon² Iterations Independent of Problem Size*, pp. 1–18.

[5] Epelman, M.–Freund, R. M. (2000): Condition number complexity of an elementary algorithm for computing a reliable solution of a conic linear system. *Mathematical Programming*, 88., (3.), pp. 451–485. <https://doi.org/10.1007/s101070000136>

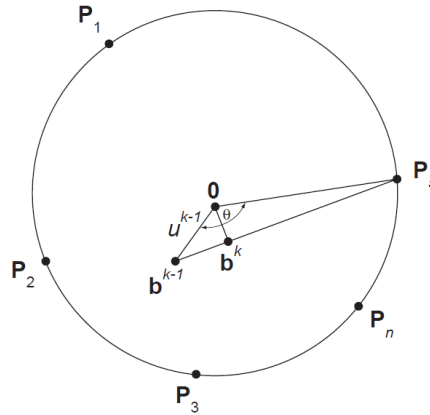
Ha a gravitációs feladat megoldható, akkor $v_{k-1} = P_s^T b^{k-1} \leq 0$.

Innen a 4. lépésben $0 < 1 - v_k - 1 < u_{k-1}^2 - v_{k-1} + 1 - v_{k-1} < 1$. 1 2. ábrán látható az Ob^k derékszögű háromszögből, hogy a következő b^k approximáció közelebb lesz az origóhoz az előző b^{k-1} approximációtól, vagyis $u_k < u_{k-1}$. Ez azért van így, mert Ob^k a derékszögű háromszög befogója, még a Ob^{k-1} a derékszögű háromszög átfogója.

A Neumann János belső pontos algoritmusában a legköltségesebb művelet a mátrixvektor szorzás, amely az algoritmus 2. lépésében felmerülő P_s pont meghatározásához szükséges.

Ennek a műveletnek a számítási komplexitása $O(mn)$. Az algoritmus konvergencia sebességét Dantzig [3, 4], illetve Epelman és Freund [5] tanulmányozta.

2. ábra. Új iteráció kiszámítása



Lemma 1. [4]: A Neumann-algoritmus által generált b^k approximáció felső korlátja.

$$\|b^k\| = \|Px^k\| \leq \frac{1}{\sqrt{k}}$$

Definíció 2.: A (2) feladat ε -megoldásának nevezzük a súlyok olyan x^k approximációját, melyre érvényes, hogy

$$x^k \geq 0, e^T x^k = 1 \text{ és } u_k = |(b^k)| = |(P x^k)| \leq \varepsilon.$$

Dantzig a Neumann-algoritmus által generált súly-approximációk sorozatára a következő konvergencia sebességet bizonyította be:

Tétel 3. [4]: Ha a (2) feladat megengedett, akkor minden $\varepsilon > 0$ -ra a Neumann-algoritmus

Legfeljebb $\left\lceil \frac{1}{\varepsilon^2} \right\rceil$ iteráción belül éri el a feladat ε -megoldását.

Dantzig a Neumann-algoritmus konvergenciasebességének tanulmányozása folyamán csak azt az esetet vizsgálta, amikor a (2) feladat megengedett. Epelman és Freund kibővítették ezt a tanulmányt (Epelman–Freund 2000) arra az esetre is, amikor a (2) feladat nem megengedett.

Legyen $H = \{Px | e^T x = 1, x \geq 0\}$ a P mátrix oszlopainak konvex burka. A (2) feladat megengedett, ha $0 \in H$. Legyen r az origótól a H határán található legközelebbi pontig mért távolság, vagyis

$$r = \inf \{ \|h - 0\| \mid h \in \partial H \} = \inf \{ \|h\| \mid h \in \partial H \}$$

Ha az origó a P mátrix oszlopainak konvex burkán fekszik, akkor $r=0$. Ebben az esetben a (2) feladatnak létezik megengedett megoldása, de a $(P, 0)$ tetszőleges perturbációja ahhoz vezethet, hogy a (2) feladat megengedhetetlen lesz. Ebben az esetben a (2) feladat instabil, vagyis rosszul feltett. Ha viszont $r > 0$, akkor a (2) feladat jól feltett. Amikor a (2) feladatnak van megoldható megoldása, r a legnagyobb, az origó középpontú gömb sugaraként értelmezhető, amely teljes egészében a P mátrix oszlopainak konvex burkában található.

[4] Dantzig, G. B. (1992): An epsilon-Precise Feasible Solution to a Linear Program with a Convexity Constraint In: *1/epsilon² Iterations Independent of Problem Size*, pp. 1–18.

[5] Epelman, M.–Freund, R. M. (2000): Condition number complexity of an elementary algorithm for computing a reliable solution of a conic linear system. *Mathematical Programming*, 88., (3.), pp. 451–485. <https://doi.org/10.1007/s101070000136>

Ha a (2) feladatnak nincs megoldható megoldása, akkor r az origótól a P mátrix oszlopainak konvex burkáig mért távolság.

Ha a (2) feladat megengedett, Epelman–Freund [5] bebizonyították a Neumann-algoritmus lineáris konvergenciasebességét.

Lemma 4. (Epelman–Freund 2000): Ha a (2) feladatnak van megengedett megoldása és $r > 0$, akkor

$$\|b^k\| \leq e^{-kr^2/2}.$$

Általános esetben a Neumann-algoritmus konvergencia sebességére Epelman és Freund a következő tételt bizonyították be:

Tétel 5. [5]: Legyen $r > 0$ és $\varepsilon > 0$. Ha a (2) feladat megengedett, a Neumann-algoritmus az ε -megoldást legfeljebb

$$\left\lceil \frac{2}{r^2} \ln \frac{1}{\varepsilon} \right\rceil$$

iterációban éri el. Ha a (2) feladat nem megengedett, a Neumann-algoritmus legfeljebb

$$\left\lceil \frac{1}{r^2} \right\rceil$$

iterációban bizonyítja a nem megengedettséget.

A Neumann-algoritmus tesztelésekor Dantzig felismerte, hogy az algoritmus konvergencia sebessége nem elég gyors ahhoz, hogy gyakorlati alkalmazásokban hasznos legyen. A konvergencia sebességének javítása érdekében Dantzig módosításokat eszközölt az algoritmuson. Mivel a kézirat a Neumann-algoritmus eredeti változatára összpontosít, a továbbiakban azt vesszük figyelembe.

Karmarkar belsőpontos algoritmus

Narendra Karmarkar 1984-ben bemutatott projektív skálázású algoritmus [6], forradalmi áttörést jelentett a belsőpontos algoritmusok fejlődésében. Az algoritmus új, hatékony megközelítést kínál a lineáris programozási feladatok megoldására, lényegesen csökkentve a szükséges iterációk számát a korábbi módszerekhez képest, mint például a simplex-módszer. Az algoritmus polinomiális időben képes megoldani a nagy méretű lineáris programozási feladatokat, ami jelentős előnyt jelent az ipari és gazdasági alkalmazásokban. Karmarkar módszere nemcsak elméleti szempontból jelentős, hanem gyakorlati alkalmazásokban is széles körben elterjedt, megteremtve az alapot számos modern optimalizálási technika számára.

Karmarkar [6] bemutatott módszere speciális LP-feladat megoldására alkalmas. Az LP-feladatnak teljesíteni kell a következő feltételeket:

1. Az LP-feladatnak létezik szigorúan megengedett megoldása és az optimális megoldások halmaza korlátos.
2. Az LP-feladat speciális kanonikus formában van:

$$\min_x c^T x, \text{ feltéve hogy } Ax = 0, e^T x = 1, x \geq 0, \quad (3)$$

ahol

$$x \in R^n, c \in R^n, A \in R^{m \times n} \text{ és } e = [1, 1, \dots, 1]^T \in R^n.$$

Ez a speciális kanonikus forma nem csökkenti a módszer alkalmazhatóságát, mivel egy új változó bevezetésével minden

$$\min_x \tilde{c}^T \tilde{x}, \text{ feltéve hogy } \tilde{A} \tilde{x} = \tilde{b}, \tilde{x} \geq 0$$

alakú LP-feladat a (3) feladatra transzformálható.

[6] Karmarkar, N. (1984): A new polynomial-time algorithm for linear programming. *Combinatorica*, 4., (4.), pp. 373–395. <https://doi.org/10.1007/bf02579150>.

3. A célfüggvény értéke az optimális megoldásban 0. Ez a feltétel erősen korlátozza az alkalmazhatóságot, de ha ismert az LP-feladat optimális célfüggvény c_m értéke, új célfüggvényként a $c^T x - c_m$ célfüggvény tekinthető.

Ha a célfüggvény minimuma ismeretlen, Karmarkar szerint módosítani lehet az algoritmust a csúszó célfüggvény bevezetésével.

Karmarkar projektív skálázású algoritmusának fő jellemzője a projektív transzformáció. A projektív transzformáció egy lineáris transzformáció, amely az aktuális pontot új koordinátarendszerbe viszi át, hogy a következő iterációkban közelebb kerüljön az optimális megoldáshoz. Ez a transzformáció olyan súlyozott változókat használ, amelyek biztosítják, hogy az új megoldás továbbra is a megengedett tartományban maradjon. Az új pont a megengedett tartomány és a szimplex metszetének terében helyezkedik el.

Karmarkar informális algoritmus:

1. **Inicializáció:** Legyen a kezdő approximáció $a = \frac{1}{n} e$.
2. **Projektív skálázás:** a lineáris transzformáció az aktuális LP-feladatot egy ekvivalens feladattá transzformálja egy másik térben.
3. **Keresési irány:** a legmeredekebb ereszkedés iránya.
4. **Lépéshossz meghatározása.**
5. **Következő iteráció meghatározása.**
6. **Inverz transzformáció:** a kapott pont inverz transzformációja.

A projektív transzformáció az x pillanatnyi iterációt a szimplex középpontjába vetíti.

Az $\bar{x}$ vetített pontot az

$$\bar{x} = \frac{X^{-1}x}{e^T X^{-1}x} \quad (4)$$

lineáris transzformáció segítségével határozzák meg, ahol $X = \text{diag}(x_1, \dots, x_n)$ átlós mátrix.

A lineáris transzformáció inverze

$$x = \frac{X\bar{x}}{e^T X\bar{x}} \quad (5)$$

A vetített legmeredekebb ereszkedés irányát a következő módon határozzák meg. Legyen a B kibővített mátrix, amelyet az AX -mátrixból kapunk, ha kibővítjük egy sorral, amely minden eleme 1. Az algoritmus keresési irányként a projektált legmeredekebb ereszkedés irány egy fajtáját használja, amelyben a projekciót a

$$P_B = I - B^T(BB^T)^{-1}B \quad (6)$$

ortogonális projekció mátrixa végzi, még a gradiens szerepét az Xc vektor játssza. A vetített legmeredekebb ereszkedés iránya tehát

$$\Delta\bar{x} = -P_B Xc. \quad (7)$$

A következő iteráció

$$\bar{x}_{k+1} = a + \alpha \frac{\Delta\bar{x}}{\|\Delta\bar{x}\|}, \quad (8)$$

ahol $a=e/n$ a kezdeti approximáció, a lépéshossz pedig

$$\alpha = \theta \frac{1}{\sqrt{n(n-1)}}, \quad (9)$$

ahol $\theta=1/4$. Az $\frac{1}{\sqrt{n(n-1)}}$ kifejezés a szimplexbe írható legnagyobb gömb sugara.

Az $\bar{x}_{k+1}$ iterációt vissza kell vetíteni az eredeti LP feladat megengedett halmazába, vagyis inverz transzformációt kell végrehajtani az (5) képlet segítségével.

[7] Schittkowski, K. (1987): *More Test Examples for Nonlinear Programming Codes*. Springer.

Karmarkar projektív skálázású algoritmus a legrosszabb esetben polinomiális időben futó algoritmus, vagyis

$$\mathcal{O}(n^{3.5}L^2)$$

ahol n az LP a feladat dimenziója, L a bemenet összes bitjének száma.

Karmarkar algoritmus elindította a belsőpontos algoritmusok fejlődését, de az újabb algoritmusok felülmúlták, mert jobb számítási komplexitással és jobb gyakorlati teljesítménnyel rendelkeznek.

Tesztelési kísérletek

Ez a fejezet a Neumann-algoritmus és Karmarkar projektív skálázású algoritmusának numerikus vizsgálatát mutatja be. A kísérleteket Matlab 2024b szoftverben végeztük, tesztfeladatként pedig a [7] által javasolt LP problémát alkalmaztuk különböző dimenziókban. A feladat célfüggvénye $f(x)=c^T x$, ahol a célfüggvény együtthatóinak c vektora a

$$c_i = \sum_{j=1}^n \frac{1}{i+j-1}, \quad i = 1, \dots, n,$$

elemekkel adott.

A megszorítások halmaza $Ax+y-b \geq 0$, $x \geq 0, y \geq 0, x, y \in \mathbb{R}^n$ alakú, ahol $A \in \mathbb{R}^{n \times n}$ mátrix elemei

$$a_{ij} = \frac{1}{i+j-1}, \quad i, j = 1, 2, \dots, n,$$

a $b \in \mathbb{R}^n$ szabad tagok vektora

$$b_i = \sum_{j=1}^n \frac{1}{i+j-1}, \quad i = 1, 2, \dots, n,$$

Kezdeti megoldásként az $x_0=0$ R^n vektort használtuk. A feladat pontos megoldása az $x^*=1 \in R^n$. A tesztelés folyamán az $n=4,6,8,10$ dimenziós feladatok lettek használva.

Mivel a Neumann-algoritmus a (2) alakú megszorítások halmazán keresi a megengedett megoldást, a tesztfeladatokat át kell transzformálni (2) alakra a következő módon (Goncalves 2004).

A $\min c^T x$, "feltéve, hogy" $Ax=b, x \geq 0$ primál LP-feladat duál párja $\max b^T y$, feltéve, hogy $A^T y + s = c, s \geq 0$. Ezekre a primál-duál párokra felírható a következő megengedettségi feladat:

$$Ax=b, A^T y + s = c, c^T x - b^T y = 0, x, s \geq 0.$$

A megengedettségi feladat megengedett megoldása egyen a primál-duál pár optimális megoldása.

Mivel az y_j változó korlátlan, ezért két új nemnegatív y_j^+ és y_j^- kerül bevezetésre úgy, hogy $y_j = y_j^+ - y_j^-$. Így a megengedettségi feladat felírható a következő alakban:

$$Ax = b, A^T y^+ - A^T y^- + s = c, c^T x - b^T y^+ + b^T y^- = 0, x, y^+, y^-, s \geq 0.$$

Tekintsük a következő mátrix-jelölést:

$$\tilde{A} = \begin{bmatrix} A & 0 & 0 & 0 \\ 0 & A^T & -A^T & I \\ c^T & -b^T & b^T & 0^T \end{bmatrix},$$

$$\alpha = [x, y^+, y^-, s]^T, \tilde{b} = [b, c, 0]^T.$$

Ekkor az előző megengedettségi feladat felírható a következő alakban:

$$\tilde{A}\alpha = \tilde{b}, \alpha \geq 0.$$

Bővítsük ki ezt a mátrixos alakot a következő megszorításokkal:

$$\tilde{A}\tilde{\alpha} - \tilde{b}\tilde{v} = 0, e^T\tilde{\alpha} + \tilde{v} = 1, \tilde{\alpha} \geq 0, \tilde{v} \geq 0,$$

A megengedettségi feladat és a kibővített megengedettségi feladat megengedett megoldásai egymás skálázásai. Legyen

$$\hat{A} = [\tilde{A}, -\tilde{b}], \hat{\alpha} = [\tilde{\alpha}, \tilde{v}]^T$$

Ekkor a kibővített megengedettségi feladat felírható a következő alakban:

$$\hat{A}\hat{\alpha} = 0, e^T\hat{\alpha} = 1, \hat{\alpha} \geq 0.$$

Legyen

$$P_j = \frac{\hat{A}_j}{\|\hat{A}_j\|}.$$

Ekkor a standard LP-feladat felírható a következő alakban:

$$P\bar{\alpha} = 0, e^T\bar{\alpha} = 1, \bar{\alpha} \geq 0,$$

ahol

$$\bar{\alpha} = [\bar{x}, \bar{y}^+, \bar{y}^-, \bar{s}, \bar{v}]^T.$$

A Neumann-algoritmus tesztelése előtt a tesztfeladatokat transzformáltuk lettek a (2) alakba. Mindkét algoritmusnál a megállási feltétel „tol”=10⁻⁴. Az iterációk száma a különböző dimenziójú LP tesztfeladatok megoldásánál az a Neumann-algoritmus és a Karmarkar-algoritmus esetén a következőképpen alakul.

- 4 dimenzió esetén: Neumann-algoritmus: 39, Karmarkar-algoritmus: 16.
- 6 dimenzió esetén: Neumann-algoritmus: 46, Karmarkar-algoritmus: 32.
- 8 dimenzió esetén: Neumann-algoritmus: 53, Karmarkar-algoritmus: 48.
- 10 dimenzió esetén: Neumann-algoritmus: 62, Karmarkar-algoritmus: 58.

A fenti értékek azt mutatják, hogy kisebb dimenzióknál a Karmarkar-algoritmus lényegesen jobb teljesítményt nyújtott, de a dimenzió növekedésével a két algoritmus közötti különbség fokozatosan csökken.

Összegzés

Ez a tanulmány Neumann János és Karmarkar belsőpontos algoritmusait vizsgálja, amelyek meghatározó szerepet játszottak a lineáris programozás fejlődésében. A szerző először bemutatja Neumann János korai belsőpontos megközelítését, majd elemzi Karmarkar algoritmusát, amely széles körben ismertté tette ezt a módszert. Noha mindkét algoritmus úttörő jelentőségű volt saját korában, a lineáris programozás fejlődésével újabb, hatékonyabb belsőpontos eljárások jelentek meg, amelyek szélesebb körű gyakorlati alkalmazást tettek lehetővé. A tanulmány összehasonlító elemzést nyújt a két algoritmus teljesítményéről különböző tesztfeladatokon, kiemelve történelmi jelentőségüket és modern alkalmazásuk korlátait.