OPERATIONS RESEARCH
REPORT 2013-03

Lineáris optimalizálás elmélete és pivot algoritmusai

Illés Tibor

2013. október 31.

Eötvös Loránd Tudományegyetem
Operációkutatási Tanszék

ISSN 1215 - 5918

Lineáris optimalizálás elmélete és pivot algoritmusai

Illés Tibor

2013. október 31.

Tartalomjegyzék

Bevezetés

A jegyzet anyaga. A jegyzet vázát azok az előadás fóliák alkotják, amelyeket az elmúlt 15 évben készítettem, illetve azok, amelyek szemináriumi vagy konferencia előadásaimhoz, cikkeimhez kapcsolódnak. Az anyag felépítése teljesen az alapoktól indul, így azt remélem, hogy érdeklődő és kitartó középiskolást, aki lineáris egyenletrendszerek megoldásáról már tanult, végig tudom vezetni a lineáris programozás érdekes és szép témakörein.[1]

A lineáris algebrai bevezető célja, hogy a közös kiindulási alapokat lerakja, rávilágítson a pivot technika szerepére; a pivot tábla, mint modell szerepére és előkészítse a következő fejezetek anyagát. A második fejezet a lineáris egyenletrendszerek megoldhatóságával és megoldásával foglalkozik. Ebben a fejezetben jelenik meg az első alternatíva tétel a Rouché–Kronecker–Capelli lemma. Kedvencem a Klafszky Emiltől származó un. Farkas-Minty-féle pivot táblás változat, amely tömören, képszerűen foglalja össze az állítást, ha az olvasó már megbarátkozott a pivot táblák világával. A lemma elnevezése is Klafszky Emiltől származik és akkor válik igazán érthetővé, amikor a következő fejezetben a lemma előjeles változatát fogalmazzuk meg. Klafszky Emil, ennek a lemmának a kapcsán többször beszélt arról, hogy a Farkas-Minty előjeles lemma a Farkas lemmának és a Minty-féle színezési lemmának egy szép közös megfogalmazása, ha az előjeleket – a Minty lemma esetén – megfelelően értelmezzük.

A második fejezetben a megoldások méretével kapcsolatos eredmények nem szerepeltek a Klafszky Emillel és Terlaky Tamással elképzelt jegyzet témakörei között. Ezek szerepe a harmadik fejezetben található MBU-szimplex algoritmus elemzéséhez kapcsolódnak először, amelyik szintén az elmúlt évek alatt jelent meg a jegyzet témakörei között. A harmadik fejezet geometriai jellegű része az új struktúrában került előre. Így a harmadik fejezet már jelentősen eltér az eredeti tervektől, annak ellenére, hogy a jegyzet egyik legérdekesebb feladata (3.50. Feladat) Klafszky Emiltől származik.

A 4. és az 5. fejezetek annak ellenére, hogy klasszikus eredményeket tárgyalunk bennük, mégis – a mai napig is újszerű tárgyalásról van szó – hiszen a végtelenül egyszerű, Terlaky-féle criss-cross algoritmus végességének a bizonyításán alapul az erős dualitás tétel konstruktív bizonyítása. A criss-cross algoritmus végesség

[1] Ha ez mégsem sikerül, az inkább az én hibám, mint az övé, mert ha ez a helyzet, akkor még mindég nem tudom eléggé egyszerűen elmondani a lineáris programozást.

bizonyítása egyszerűbb, mint Terlaky Tamás eredeti, az 1980-as évek közepéről származó bizonyítása és Klafszky Emil egy észrevételén alapul.

A jegyzet bővítését tervezem olyan részekkel illetve fejezetekkel, amelyekben érdekes és hasznos témakörök kerülnének feldolgozásra. A teljesség igénye nélkül megemlíteném két témakört: a Klee-Minty féle példával elindult, a pivot algoritmusok – elméleti – hatékonyságát firtató komplexitási kérdések illetve a gyakorlatban fontos és nagyon hasznos érzékenységvizsgálattal kapcsolatos témakör.

A jegyzetet Utószó zárja.

Mit, hol és kiknek tanítottam lineáris programozásból ? A jegyzet teljes anyagát, az ELTE TTK matematikus és alkalmazott matematikus, 4. és 5. éves hallgatóinak *két 2 órás* előadás keretében adtam elő az elmúlt 15 év során, majdnem minden évben. Az előadásaimon természetesen felhasználtam azt a több száz előadás fóliát, amelyet az évek során készítettem és amelyek a jegyzet első latex változatának tekinthetők. Az ELTE-n, a mesterképzés bevezetése előtt programtervező matematikus hallgatóim is voltak, akik az első lineáris programozás kurzusomat vették fel. A *Lineáris Programozás I.* című tárgyam anyagának nagy részét lefedi ennek a jegyzetnek a tartalma. A Lineáris Programozás I. előadás végén, általában elmondtam az un. *teljes lépéses, belső pontos primál-duál büntetőfüggvényes Newton módszert*.)

A *Lineáris Programozás II.* című tárgyam anyagát a lineáris programozás belsőpontos módszerei és ezekhez kapcsolódó elméleti alapok képezték. Ezt az anyagot egy külön jegyzetben teszem majd közzé.

Előadásaimat időnként olyan doktorandusz hallgatók is látogatták, akik egyetemi diploma megszerzése előtt, 4. és 5. éves korukban nem vették fel az óráimat. Az elmúlt 12 tanévben, amióta ez az anyag képezi előadásaim nagy részét, a mester szintű lineáris programozás kurzusaimon, közel 200 hallgatót vizsgáztattam az ELTE-n.

Néhány éve, egész pontosan 2010 tavasza óta a BME-n, első éves Alkalmazott Matematikus/Matematikus mester képzésben résztvevő hallgatóknak tartok Lineáris programozás előadást heti 4 órában (3 előadás, 1 gyakorlat). Ennek az előadásnak az alapját, az ELTE-n éveken keresztül tartott Lineáris Programozás I. és II. kurzusaimhoz kidolgozott tananyag képezi. A BME TTK-n 2012 őszén, rekord mennyiségű hallgatóm volt, több, mint 40-en vették fel a Lineáris Programozás tárgyat. (Remélem nem bánták meg.)

Lineáris programozás előadást az 1990/91-es tanévben tartottam először az ELTE-n. Később az Eastern Mediterranean University-n, a Strathclyde University-n, majd pedig a BME-n is tanítottam, jelenleg is tanítok, lineáris programozást. Az ELTE TTK-n, egy évfolyam – az 1997/98-as ELTE matematikus és alkalmazott matematikus – kivételével, kizárólag 4. és 5. éves alkalmazott matematikus és matematikus, később mester szakos illetve doktorandusz hallgatókat oktattam. A jegyzet anyagának most elkészült változata az elmúlt több, mint 20 év alatt tisztázódott le, alakult ki. Az utolsó 10-15 évben a tananyag formálói – igényeikkel és kritikájukkal – mindinkább a tehetséges alkalmazott matematikus, matematikus hallgatóim, doktoranduszaim lettek. Számos lineáris optimalizálási témában vezettem szakdolgozatot. Legsikeresebb doktoranduszaimmal – Csizmadia Zsolt, Eisenberg-Nagy Marianna, Nagy Adrienn vagy a fiatalabbak közül Molnár-Szipai Richárd, Egri Attila – a lineáris optimalizálás témakörében közösen értünk el eredményeket, amelyeknek egyikét–másikát beépítettem a tananyagba illetve egy bővített jegyzet anyagába biztosan beépíteném eredeti formájukban vagy lineáris programozási feladatra specializáltan.

Köszönetnyilvánítás.

A jegyzetet jelenlegi változatának elkészítését, a TÁMOP-4.1.2.A/1-11/1-2011-0025 hivatkozási számú, *Interdiszciplináris és komplex megközelítésű digitális tananyagfejlesztés a természettudományi képzési terület mesterszakjaihoz* című pályázata támogatta.

A jegyzet írás nehéz és háládatlan feladatához mérhető a lektori munka is. A jegyzetemnek két lektora volt. E.-Nagy Marianna, volt tanítványom, aki részleteiben követve igyekezett segíteni a munkámat. Precízen bejelölve, a javítandó, szépítendő részeket. Igyekeztem a javításokat elvégezni, de azt hiszem, kihasználom a jegyzet elektronikus voltának az előnyeit, és a bennmaradó hibák javításával időről-időre foglalkozom majd. E.-Nagy Marianna segítségét és kitartó munkáját őszintén köszönöm.

Maros István, – volt tanárom – a hivatalos szakmai lektora a jegyzetemnek, amely nagy megtiszteltetés a számomra. Maros professzor úr maga is két könyv szerzője.[2] Szakmai véleménye, észrevételei és megjegyzései mindig is fontosak voltak a számomra. Maros professzor úrnak, a jegyzetem lektorálásával eltöltött idejét és munkáját, hasznos tanácsait nagyon szépen köszönöm.

Természetesen, hozzá kell tennem, hogy a jegyzetben megmaradó pontatlanságok és hibák kizárólag a szerző munkáját minősítik. Ahogyan észlelem ezeket, vagy felhívják a figyelmemet ilyenekre, folyamatosan javítani fogom a jegyzetemet és a pontosított változatokat a honlapomon is nyilvánosságra hozom.

Budapest, 2013. október 31.

Illés Tibor

[2] A *Computational Techniques of the Simplex Method* című Kluwer kiadónál 2003-ban megjelent könyve, nem csak a szimplex, de általában a lineáris programozás pivot algoritmusaival kapcsolatos számítási technikákat összefoglaló, legteljesebb mű az operációkutatás szakirodalmában. Sajnálattal jegyzem meg, hogy hasonló, a belső pontos algoritmusok számítási technikáinak a tárházát összefoglaló mű, még nem létezik.

1. fejezet

Lineáris algebra: rövid összefoglaló

A lineáris algebra a matematikának az a területe, amely a következő két alapvető matematikai objektum, a lineáris egyenletrendszerek és vektorok, vizsgálatából született.

A lineáris egyenletrendszerek első ismert alkalmazása, földterület nagyságának a meghatározásáról szól, és az ókori babiloni matematika szövegekben található meg. A lineáris algebra történetéről számos érdekesség ismert. A négyzetes lineáris egyenletrendszerek megoldására szolgáló első eljárást, *A matematikai művészet kilenc fejezete* című kínai könyvből ismerjük, amely feltehetően egy időszámításunk előtt 2000 évvel kidolgozott módszert ír le. Az eljárás kidolgozásának az alapja a mátrix fogalom, – ahogyan a kínaiak nevezték, *fang-cseng* – bevezetése volt. A fang-cseng négyzetes táblázatra utal, amelyben a négyzetes lineáris egyenletrendszer együtthatóit rendezték el.

Ebben a fejezetben, összefoglaljuk a lineáris algebra alapvető fogalmait és állításait, amelyek ismerete szükséges lineáris egyenletrendszerek, lineáris egyenlőtlenségrendszerek és lineáris programozási feladatok tárgyalásához. Tárgyalásmódunk követi Klafszky Emil és Terlaky Tamás által kidolgozott, a pivotálás műveletén alapuló, konstruktív felépítést.

1.1. Vektorterek

A szakirodalomban szokásos módon vezessük be a vektortér fogalmát.

1.1. Definíció. Az összeadással és skalárral való szorzással ellátott $\mathcal{L}$ nem üres halmaz *vektortere*t (más szóval *lineáris tere*t) alkot a Γ számtest felett, ha bármely $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z} \in \mathcal{L}$ és λ, $\mu \in \Gamma$ esetén a következő feltételek mindegyike teljesül:

$$\mathbf{x} + \mathbf{y} \in \mathcal{L} \tag{1.1}$$

$$\mathbf{x} + \mathbf{y} = \mathbf{y} + \mathbf{x} \tag{1.2}$$

$$(\mathbf{x} + \mathbf{y}) + \mathbf{z} = \mathbf{x} + (\mathbf{y} + \mathbf{z}) \tag{1.3}$$

létezik ún. *nullelem*, $\mathbf{0} \in \mathcal{L}$, amelyre

$$\mathbf{x} + \mathbf{0} = \mathbf{x} \tag{1.4}$$

minden $\mathbf{x} \in \mathcal{L}$ elemnek létezik *ellentett* eleme $-\mathbf{x} \in \mathcal{L}$, amelyre

$$(-\mathbf{x}) + \mathbf{x} = \mathbf{0}, \tag{1.5}$$

teljesül. Továbbá

$$\lambda\, \mathbf{x} \in \mathcal{L} \tag{1.6}$$

$$(\lambda + \mu)\, \mathbf{x} = \lambda\, \mathbf{x} + \mu\, \mathbf{x} \tag{1.7}$$

$$\lambda\, (\mathbf{x} + \mathbf{y}) = \lambda\, \mathbf{x} + \lambda\, \mathbf{y} \tag{1.8}$$

$$(\lambda\, \mu)\, \mathbf{x} = \lambda\, (\mu\, \mathbf{x}) \tag{1.9}$$

ha $1 \in \Gamma$ a test *egységeleme*, akkor

$$1\, \mathbf{x} = \mathbf{x}. \tag{1.10}$$

Természetes módon megadható a vektortér alterének a fogalma.

1.2. Definíció. Egy $\mathcal{L}' \subseteq \mathcal{L}$ nem üres részhalmazt az $\mathcal{L}$ lineáris tér *alter*ének nevezzük, ha az $\mathcal{L}'$ maga is eleget tesz a lineáris térrel szemben támasztott követelményeknek.

Az előző két definíció felhasználásával könnyen igazolható a következő állítás.

1.3. Állítás. *Legyen $\mathcal{L}$ lineáris tér a Γ számtest felett. Az $\mathcal{L}' \subset \mathcal{L}$ nem üres részhalmaza pontosan akkor altere az $\mathcal{L}$ lineáris térnek, ha bármely $\mathbf{x}$, $\mathbf{y} \in \mathcal{L}'$ és $\lambda \in \Gamma$ esetén $\mathbf{x} + \mathbf{y} \in \mathcal{L}'$ és $\lambda\, \mathbf{x} \in \mathcal{L}'$ teljesül.*

A vektortér definíciójában megfogalmazott tulajdonságok (összeadás és skalárral való szorzás zártsága; az összeadás asszociativitása) segítségével, a vektorok összege kiterjeszthető több, mint két tagra. Ezt felhasználva, bevezethetjük a végesen sok vektor által generált altér fogalmát.

1.4. Definíció. Legyen $\mathcal{L}$ vektortér és $\mathcal{L}' \subseteq \mathcal{L}$ altér és $\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_k \in \mathcal{L}'$ tetszőleges vektorok. A

$$\mathcal{V} = \{\mathbf{v} : \mathbf{v} = v_1\, \mathbf{a}_1 + v_2\, \mathbf{a}_2 + \cdots + v_k\, \mathbf{a}_k,\ \text{ahol}\ v_1, v_2, \cdots, v_k \in \Gamma\}$$

halmazt, amely az $\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_k$ vektorok által *generált alter*nek nevezzük, és $\mathcal{L}(\{\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_k\})$-val jelöljük.

Az előző definícióban szereplő $\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_k \in \mathcal{L}'$ vektorokat, *generáló vektorok*nak nevezzük.

Az előző definícióban meghatározott $\mathbf{v} \in \mathcal{V}$ vektorokat az $\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_k$ vektorok *lineáris kombináció*jának nevezzük.

Egyszerűen igazolható, hogy a végesen generált altér, lineáris altere a generáló vektorokat tartalmazó vektortérnek.

1.5. Állítás. *A $\mathcal{V} \subseteq \mathcal{L}'$ lineáris altér.*

Az előző állítást kicsit másképpen is megfogalmazhatjuk: egy vektortérből tetszőlegesen kiválasztott véges sok vektor, lineáris kombinációinak a halmaza, az eredeti vektortérnek egy alterét határozza meg.

A vektortérrel kapcsolatos alapvető fogalmak bevezetése után, három feladatot fogalmazunk meg a korábbi definíciók illusztrálására.

1.6. Feladat. Legyen $a, b \in \mathbb{R}$. Tekintsük az $f : [a, b] \to \mathbb{R}$ folytonos függvényeknek az $\mathcal{F}$ halmazát. Az $f, g \in \mathcal{F}$ függvények összegét értelmezzük az alábbi módon:

$$(f + g)(x) := f(x) + g(x)$$

teljesül, bármely $x \in [a, b]$ esetén. Függvény és valós szám szorzata pedig legyen

$$(u\, f)(x) := u\, f(x)$$

bármely $x \in [a, b]$ és tetszőleges $u \in \mathbb{R}$ esetén. Bizonyítsa be, hogy az $\mathcal{F}$ halmaz vektorteret alkot az összeadásra és szorzásra nézve.

Legyen $\mathcal{L}_1, \mathcal{L}_2 \subset \mathcal{L}$ az $\mathcal{L}$ vektortér, két adott altere. A két altér összegét az alábbi módon értelmezhetjük

$$\mathcal{V} = \mathcal{L}_1 + \mathcal{L}_2 = \{\mathbf{a}_1 + \mathbf{a}_2 \;:\; \text{bármely } \mathbf{a}_1 \in \mathcal{L}_1 \text{ és bármely } \mathbf{a}_2 \in \mathcal{L}_2 \text{ esetén } \}.$$

Ezek után igazolhatók a következő feladat állításai.

1.7. Feladat. Legyen $\mathcal{L}$ vektortér és az $\mathcal{L}_1, \mathcal{L}_2, \ldots \mathcal{L}_k \subset \mathcal{L}$ alterei. Bizonyítsa be, hogy

(i) $\mathcal{V} = \mathcal{L}_1 + \mathcal{L}_2 \subset \mathcal{L}$ altér;

(ii) $\mathcal{L}_1 + \mathcal{L}_2 + \cdots + \mathcal{L}_k$ altér és

(iii) $\mathcal{L}_1 \cap \mathcal{L}_2 \cap \cdots \cap \mathcal{L}_k$ altér.

A lineáris egyenletrendszerek, lineáris egyenlőtlenségrendszerek és lineáris programozás tárgyalása során olyan vektortereket, altereket használunk, amelyek esetén a Γ test, a valós számok, $\mathbb{R}$, alkotta test. A vektortér elemei, vektorai pedig a valós szám m-esek lesznek. A valós szám m-esek halmazát $\mathbb{R}^m$ jelöli. Könnyen belátható, hogy a valós szám m-esek halmaza, ha a $\Gamma = \mathbb{R}$, azaz a valós szám test, akkor teljesíti a vektortér definíciójában szereplő összes feltételt.

A következő feladatban, a valós szám m-esek által meghatározott vektortér végesen generált altereire vonatkozó tulajdonságokat kell igazolni.

1.8. Feladat. Legyen $\{\mathbf{a}_1, \mathbf{a}_2, ...\mathbf{a}_n\} \subset \mathbb{R}^m$ vektorrendszer. Bizonyítsa be, hogy

(i) $\mathcal{L}(\mathbf{a}_1, \ldots, \mathbf{a}_j, \ldots, \mathbf{a}_n) = \mathcal{L}(\mathbf{a}_1, \ldots, \lambda\, \mathbf{a}_j, \ldots, \mathbf{a}_n)$, ahol $\lambda \in \mathbb{R}$, $\lambda \neq 0$, és

(ii) $\mathcal{L}(\mathbf{a}_1, \ldots, \mathbf{a}_j, \ldots, \mathbf{a}_n) = \mathcal{L}(\mathbf{a}_1, \ldots, \mathbf{a}_j + \mathbf{a}_k, \ldots, \mathbf{a}_n)$, ahol $1 \leq k \leq n$.

1.2. Skaláris szorzat

Szükségünk lesz a vektorok közötti szorzás – skaláris szorzás – műveletére. A skaláris szorzás, egy olyan leképezés, amelyik tetszőleges vektor párhoz egy skalárt rendel és rendelkezik néhány tulajdonsággal

1.9. Definíció. Legyen adott egy $\mathcal{L}$ vektortér a Γ számtest felett. Tekintsük a

$$<,>: \mathcal{L} \times \mathcal{L} \to \Gamma$$

leképezést, amely az $\mathcal{L}$ elempárjaihoz egy számot rendel. A $<,>$ leképezést *skaláris szorzás*nak nevezzük, ha teljesíti az alábbi feltételeket:

$$\begin{aligned}
<\mathbf{a},\mathbf{b}> &= <\mathbf{b},\mathbf{a}> \\
<\mathbf{a},\mathbf{b}+\mathbf{c}> &= <\mathbf{a},\mathbf{b}> + <\mathbf{a},\mathbf{c}> \\
\lambda <\mathbf{a},\mathbf{b}> &= <\lambda\,\mathbf{a},\mathbf{b}> \\
<\mathbf{a},\mathbf{a}> &\geq 0 \\
<\mathbf{a},\mathbf{a}> &= 0 \ \text{ akkor és csakis akkor, ha } \ \mathbf{a} = \mathbf{0}
\end{aligned}$$

ahol $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$ tetszőleges elemei a vektortérnek, $\lambda \in \Gamma$ és $\mathbf{0} \in \mathcal{L}$ a vektortér nulleleme.

Gyakran fogjuk használni az $\mathbb{R}^m$ vektortér, elemeinek a skaláris szorzatát, amelyet az előző általános definíciónál egyszerűbben értelmezünk.

1.10. Definíció. Legyen $\mathbf{a}, \mathbf{b} \in \mathbb{R}^m$. Az $\mathbf{a}$ és $\mathbf{b}$ vektorok (valós szám m-esek) *skaláris szorzat*át a következő módon definiáljuk:

$$< \mathbf{a}, \mathbf{b} > = \ a_1\, b_1 + a_2\, b_2 + \cdots + a_m\, b_m = \mathbf{a}^T\, \mathbf{b}. \tag{1.11}$$

A következő állítás igazolja, hogy az $\mathbb{R}^m$ vektortér, esetén definiált leképezésre jogosan használjuk a skaláris szorzás elnevezést.

Az állítás bizonyításában, az 1.9. definícióban elvárt feltételek teljesülését kell kimutatni.

1.11. Állítás. *Az $\mathbb{R}^m$ téren a (1.11) képlettel adott leképezés valóban skaláris szorzás.*

A Cauchy-Schwarz-Bunyakovszkij-egyenlőtlenség alkalmazására többször kerül majd sor a lineáris programozás belsőpontos módszereinek az elemzése során.

1.12. Tétel. (Cauchy-Schwarz-Bunyakovszkij-egyenlőtlenség.) *Legyen adott az $\mathcal{L}$ vektortér az $\mathbb{R}$ számtest felett, és két tetszőleges $\mathbf{a}, \mathbf{b} \in \mathcal{L}$ vektor, akkor*

$$< \mathbf{a}, \mathbf{b} >^2 \leq < \mathbf{a}, \mathbf{a} >< \mathbf{b}, \mathbf{b} > \tag{1.12}$$

és egyenlőség pontosan akkor teljesül, ha $\mathbf{a} = \lambda\, \mathbf{b}$, valamely $\lambda \in \mathbb{R}$ esetén.

Bizonyítás. A negyedik tulajdonság alapján

$$< \mathbf{a} - \lambda\, \mathbf{b},\, \mathbf{a} - \lambda\, \mathbf{b} > \geq \ 0$$

és egyenlőség akkor és csakis akkor teljesül, ha $\mathbf{a} = \lambda\, \mathbf{b}$. Átalakítások után

$$\lambda^2 \ < \mathbf{b}, \mathbf{b} > -\lambda\, (2 \ < \mathbf{a}, \mathbf{b} >) + < \mathbf{a}, \mathbf{a} > \geq \ 0$$

teljesül bármely $\lambda \in \Gamma$ számra. Az előző egyenlőtlenségből következik, hogy a λ-ban másodfokú kifejezés diszkriminánsa nem pozitív, azaz

$$4 \ < \mathbf{a}, \mathbf{b} >^2 - 4 \ < \mathbf{a}, \mathbf{a} >< \mathbf{b}, \mathbf{b} > \leq \ 0.$$ ■

Az 1.6. feladatban bevezettünk egy érdekes vektorteret, az adott, zárt intervallumon folytonos, valósértékű függvények halmazáról mutattuk meg, hogy a feladatban értelmezett összeadásra és a valós számokkal való szorzásra nézve, vektorteret alkotnak.

Az alábbi feladattal azt szeretnénk illusztrálni, hogy a skaláris szorzás fogalma számos helyen – például az 1.6. feladatban megadott vektortéren – fordulhat elő.

1.13. Feladat. Legyen adott az $[a, b]$ intervallumon definiált folytonos függvények $\mathcal{F}$ halmaza. Az $f, g \in \mathcal{F}$ függvényeknek a valós számok halmazára történő leképezését értelmezzük az alábbi módon:

$$< f, g > := \int_a^b f(x)\, g(x)\, dx \tag{1.13}$$

Bizonyítsa be, hogy a (1.13) képlettel értelmezett leképezés skaláris szorzat az $\mathcal{F}$ halmazon definiált vektortéren.

A skaláris szorzatról szóló rövid összefoglalónkat, a Cauchy-Schwarz-Bunyakovszkij-egyenlőtlenség megfelelő, az adott, zárt intervallumon folytonos, valósértékű függvények vektorterén, az előző feladatban értelmezett skaláris szorzatára vonatkozóan fogalmazzuk meg.

1.14. Feladat. Legyenek $f, g \in \mathcal{F}$, ahol az $\mathcal{F}$ halmazt az előző feladatban definiáltuk.

(i) Bizonyítsa be a következő egyenlőtlenséget

$$\left(\int_a^b f(x)\, g(x)\, dx \right)^2 \le \left(\int_a^b f^2(x)\, dx \right) \left(\int_a^b g^2(x)\, dx \right) \tag{1.14}$$

(ii) Írja fel annak a szükséges és elégséges feltételét, hogy a (1.14) egyenlőtlenség egyenlőséggel teljesüljön.

1.3. Vektornorma, távolság

A valós szám m-esek vektorterén, $m = 2$ és $m = 3$ esetén, már középiskolában értelmeztük a vektor hosszát, intenzitását. A vektorok hosszához hasonló fogalmat vezetünk be most általános esetre.

1.15. Definíció. Legyen adott egy $\mathcal{L}$ vektortér a Γ skalárok teste felett. A $\|.\| : \mathcal{L} \to \Gamma$ leképezést *(vektor) normá*nak nevezzük, ha rendelkezik az alábbi tulajdonságokkal:

$$\begin{aligned} \|\mathbf{a}\| &\ge 0, & & \\ \|\mathbf{a}\| &= 0, & &\text{akkor és csakis akkor, ha } \mathbf{a} = \mathbf{0}, \\ \|\lambda\, \mathbf{a}\| &= |\lambda|\, \|\mathbf{a}\| & &\text{teljesül tetszőleges } \lambda \in \Gamma \text{ számra,} \end{aligned}$$

$$\|\mathbf{a}+\mathbf{b}\| \quad \leq \quad \|\mathbf{a}\| + \|\mathbf{b}\|\,,$$

és egyenlőség akkor és csak akkor áll fenn az utolsó egyenlőtlenségben, ha $\mathbf{a} = \lambda\, \mathbf{b}$ valamely $\lambda \in \Gamma$ számra.

Az általános vektornorma definícióját kielégítő speciális leképezést értelmezünk.

1.16. Definíció. Az alábbi normát *euklideszi normá*nak nevezzük:

$$\|\mathbf{a}\| \ := \ \sqrt{<\mathbf{a},\mathbf{a}>}$$

Szükséges igazolnunk, hogy az *euklideszi normá*nak elnevezettleképezés valóban teljesít a vektornorma tulajdonságait. A bizonyítás egyszerű, az 1.15. Definícióban elvárt tulajdonságok teljesülését kell leellenőrizni.

1.17. Állítás. *Az euklideszi normára teljesülnek a vektornormákra vonatkozó feltételek.*

Az 1.16. Definíció és az 1.17. Állítás alapján, az euklideszi normával ellátott vektortereket *Euklideszi ter*eknek nevezzük.

Ezek után értelmezhetjük a vektorok közötti távolságot.

1.18. Definíció. Legyen adott egy $\mathcal{L}$ vektortér a skalárok Γ teste felett. Egy

$$\mathrm{dis} : \mathcal{L} \times \mathcal{L} \to \Gamma$$

leképezést *távolság*nak (metrikának) nevezünk, ha teljesülnek az alábbi feltételek:

$$\begin{aligned}
\mathrm{dis}(\mathbf{a},\mathbf{b}) &= \mathrm{dis}(\mathbf{b},\mathbf{a}),\\
\mathrm{dis}(\mathbf{a},\mathbf{a}) &\geq 0,\\
\mathrm{dis}(\mathbf{a},\mathbf{b}) &= 0 \quad \Longleftrightarrow \quad \mathbf{a}=\mathbf{b},\\
\mathrm{dis}(\mathbf{a},\mathbf{b}) + \mathrm{dis}(\mathbf{b},\mathbf{c}) &\geq \mathrm{dis}(\mathbf{a},\mathbf{c}),
\end{aligned}$$

minden $\mathbf{a},\ \mathbf{b}, \mathbf{c} \in \mathcal{L}$ vektorra.

Távolságot (metrikát), könnyen definiálhatunk norma segítségével.

1.19. Definíció. Az alábbi leképezést *norma által generált távolság*nak nevezzük

$$\mathrm{dis}(\mathbf{a},\mathbf{b}) := \|\mathbf{a}-\mathbf{b}\|\,. \tag{1.15}$$

Természetesen igazolnunk kell, hogy a norma segítségével definiált leképezés teljesíti a távolsággal (metrikával) szemben támasztott követelményeket.

1.20. Állítás. *A norma által generált távolságra teljesülnek az előbbi feltételek.*

Ha egy halmazon adott egy olyan leképezés, amely a halmaz elempárjaihoz számot rendel és teljesíti a metrikától elvárt tulajdonságokat, akkor *metrikus tér*ről beszélünk.

1.4. Generáló táblák, pivotálás

A jegyzet további részében, a valós számok feletti vektortérre szűkítjük le vizsgálatainkat. Először bevezetjük a generáló rendszer fogalmát, majd a pivotálást és ezen keresztül a generáló táblákat.

1.21. Definíció. Egy vektorhalmaz részhalmazát *generáló rendszer*nek nevezzük, ha a vektorhalmaz minden eleme előáll mint a generáló rendszer elemeinek lineáris kombinációja.

Legyenek adottak az $\mathbf{a}_i \in \mathcal{L}$ vektorok és jelölje $\mathcal{J} = \{1, 2, ..., n\}$ az indexeik halmazát. Az adott véges vektorhalmaz valamely generáló rendszerét megadhatom a generáló rendszerben szereplő $\mathbf{a}_i$ vektorok indexhalmazának, $\mathcal{J}_G$ megadásával.

Ekkor

$$\mathbf{a}_j = \sum_{i \in \mathcal{J}_G} t_{ij}\,\mathbf{a}_i, \qquad \forall j \in \mathcal{J},$$

ahol t_{ij} az $\mathbf{a}_i$, $i \in \mathcal{J}_G$ vektor együtthatója az $\mathbf{a}_j$, $j \in \mathcal{J}$ vektor előállításában. A t_{ij} együtthatókból készített mátrixot *(hosszú) generáló táblá*nak nevezzük:

$$\begin{array}{cc|ccc|}
 & & & \mathcal{J} & \\
 & & & \mathbf{a}_j & \\
\cline{3-5}
 & & & \vdots & \\
\mathcal{J}_G & \mathbf{a}_i & \dots & t_{ij} & \dots \\
 & & & \vdots & \\
\cline{3-5}
\end{array}$$

Mivel az $\mathbf{a}_i$ generáló elemek triviálisan is kifejezhetők (előállításukhoz nincsen más generáló elemre szükség önmagukon kívül), ezért gyakran csak az alábbi *rövid generáló táblá*t írjuk fel, ahol $\bar{\mathcal{J}}_G = \mathcal{J} \setminus \mathcal{J}_G$ (a nem generáló – vagy másnéven generált – vektorok halmaza):

$$\begin{array}{cc|ccc|}
 & & & \bar{\mathcal{J}}_G & \\
 & & & \mathbf{a}_j & \\
\hline
 & & & \vdots & \\
\mathcal{J}_G & \mathbf{a}_i & \dots & t_{ij} & \dots \\
 & & & \vdots & \\
\hline
\end{array}$$

A következő egyszerű példa mutatja, hogy egy generáló rendszerhez tartozó generáló tábla nem feltétlenül egyértelmű. Legyenek adottak az $\mathbf{a}_1^T = (1,0,3)$, $\mathbf{a}_2^T = (0,1,-1)$, $\mathbf{a}_3^T = (1,0,1)$, $\mathbf{a}_4^T = (0,2,-1)$, $\mathbf{a}_5^T = (1,1,1)$, és $\mathbf{a}_6^T = (3,5,5)$ vektorok. Ekkor a $\mathcal{J}_G = \{1,2,3,5\}$ generáló rendszerhez az alábbi két generáló tábla is tartozhat:

	$\mathbf{a}_4$	$\mathbf{b}$
$\mathbf{a}_1$	0	3
$\mathbf{a}_2$	1	4
$\mathbf{a}_3$	-1	-1
$\mathbf{a}_5$	1	1

T tábla

	$\mathbf{a}_4$	$\mathbf{b}$
$\mathbf{a}_1$	2	4
$\mathbf{a}_2$	5	6
$\mathbf{a}_3$	1	0
$\mathbf{a}_5$	-3	-1

$\hat{T}$ tábla

A generáló rendszer egy egyszerű tulajdonsága a bővíthetősége. Ezt fogalmazza meg a következő feladat.

1.22. Feladat. Legyen adott az $\{\mathbf{a}_1, \mathbf{a}_2 \dots \mathbf{a}_n\} \subset \mathbb{R}^m$ véges vektorrendszer. Jelölje $\mathcal{J}_G$ a generáló vektorok index halmazát és $\bar{\mathcal{J}}_G = \mathcal{J} \setminus \mathcal{J}_G$ a generáló rendszerhez nem tartozó vektorokét. Tegyük fel, hogy $\bar{\mathcal{J}}_G \neq \emptyset$ és legyen $\hat{\mathcal{J}}$ tetszőleges, nem üres részhalmaza a $\bar{\mathcal{J}}_G$ index halmaznak. Igazolja, hogy a $\mathcal{J}_G \cup \hat{\mathcal{J}}$ indexek halmazzal meghatározott vektorrendszer is generáló rendszere az $\mathbf{a}_1, \mathbf{a}_2 \cdots \mathbf{a}_n$ vektoroknak.

A gneráló rendszer bővíthetőségének egy másik módját írja le az alábbi feladat.

1.23. Feladat. Legyen adott az $\{\mathbf{a}_1, \mathbf{a}_2 \cdots \mathbf{a}_n\} \subset \mathbb{R}^m$ véges vektorrendszer. Jelölje $\mathcal{J}_G'$ illetve $\mathcal{J}_G''$ a generáló vektorok két, különböző index halmazát. Bizonyítsa be, hogyha

$$|\mathcal{J}_G'| < |\mathcal{J}_G''|$$

akkor létezik olyan $k \in \mathcal{J}_G'' \setminus \mathcal{J}_G'$ index, hogy $\mathcal{J}_G'' \setminus \{k\}$ is generáló részrendszerét definiálja az adott vektorrendszernek.

Az előző feladatokban generáló rendszerek konstruálására adtunk eljárásokat. Érdekesebb tulajdonsága a generáló rendszereknek az, hogy bizonyos feltételek mellett egymásba transzformálhatók. A következő tétel, azt mutatja meg, hogy egy adott generáló rendszerből, milyen feltételek melett és hogyan lehet előállítani egy újabb generáló rendszert.

1.24. Tétel. *Legyen adott az* $\{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n\} \subset \mathcal{L}$ *véges vektorrendszer, ahol* $\mathcal{L}$ *a valós számok teste feletti vektortér. Ha* $t_{rs} \neq 0$ *akkor a* $\mathcal{J}_G$ *generáló rendszerben lévő* $\mathbf{a}_r$ *vektort kicserélhetjük a generáló rendszerben nem szereplő* $\mathbf{a}_s$ $(s \in \bar{\mathcal{J}}_G)$ *vektorral az alábbi módon:*

$$\begin{array}{rclr} t'_{ij} & = & t_{ij} - \dfrac{t_{rj}\, t_{is}}{t_{rs}} & i \in \mathcal{J}'_G,\ i \neq s,\ j \in \bar{\mathcal{J}}'_G,\ j \neq r \\ t'_{sj} & = & \dfrac{t_{rj}}{t_{rs}} & j \in \bar{\mathcal{J}}'_G,\ j \neq r \\ t'_{ir} & = & -\dfrac{t_{is}}{t_{rs}} & i \in \mathcal{J}'_G,\ i \neq s \\ t'_{sr} & = & \dfrac{1}{t_{rs}}, & \end{array}$$

ahol $\mathcal{J}'_G = \mathcal{J}_G \setminus \{r\} \cup \{s\}$ *az új generáló rendszer indexhalmaza és* t'_{ij} *az előállításban szereplő új együtthatók.*

Bizonyítás. Kiinduló generáló táblát az alábbi formában adhatjuk meg

$$\begin{array}{cc|ccccc|} & & & & \bar{\mathcal{J}}_G & & \\ & & & \mathbf{a}_j & & \mathbf{a}_s & \\ \cline{3-7} & & & \vdots & & \vdots & \\ & \mathbf{a}_i & \dots & t_{ij} & \dots & t_{is} & \dots \\ \mathcal{J}_G & & & \vdots & & \vdots & \\ & & & \vdots & & \vdots & \\ & \mathbf{a}_r & \dots & t_{rj} & \dots & t_{rs} & \dots \\ & & & \vdots & & \vdots & \\ \cline{3-7} \end{array}$$

A tételben leírtaknak megfelelően, az $\mathbf{a}_r$ vektor távozik a generáló rendszerből és az $\mathbf{a}_s$ vektor pedig bekerül az új generáló rendszerbe. A vektorok felcserélése után, az új generáló táblához jutunk:

$$
\begin{array}{ccc}
 & & \bar{\mathcal{J}}'_G \\
 & & \begin{array}{ccccc} & \mathbf{a}_j & & \mathbf{a}_r & \end{array} \\
\mathcal{J}'_G & \begin{array}{c} \\ \mathbf{a}_i \\ \\ \\ \mathbf{a}_s \\ \\ \end{array} & \left|\begin{array}{ccccc}
 & \vdots & & \vdots & \\
\dots & t'_{ij} & \dots & t'_{ir} & \dots \\
 & \vdots & & \vdots & \\
 & \vdots & & \vdots & \\
\dots & t'_{sj} & \dots & t'_{sr} & \dots \\
 & \vdots & & \vdots &
\end{array}\right|
\end{array}
$$

Induljunk ki az $\mathbf{a}_s$ vektor előállításából, amelyet a kiinduló generáló táblából olvashatunk ki:

$$\mathbf{a}_s = t_{rs}\,\mathbf{a}_r + \sum_{i \in J_G \setminus \{r\}} t_{is}\,\mathbf{a}_i$$

Felhasználva azt a feltételt, hogy $t_{rs} \neq 0$, fejezzük ki az $\mathbf{a}_r$ vektort:

$$\mathbf{a}_r = \frac{1}{t_{rs}}\,\mathbf{a}_s + \sum_{i \in J_G \setminus \{r\}} \left(-\frac{t_{is}}{t_{rs}}\right) \mathbf{a}_i.$$

A kifejezésben szereplő együtthatók már az új generáló rendszerben megadott együtthatók, azaz

$$t'_{sr} = \frac{1}{t_{rs}} \quad \text{és} \quad t'_{ir} = -\frac{t_{is}}{t_{rs}}.$$

Ezzel beláttuk a 3. és 4. összefüggéseket.

Az $\mathbf{a}_j,\ j \in \bar{\mathcal{J}}_G,\ j \neq s$ vektor felírásából kiindulva hasonlóan igazolható az 1. és 2. összefüggés is. ■

Az előző tételben megfogalmazott számítási eljárást az $\mathbf{a}_r$ generáló vektor és az $\mathbf{a}_s$ generált vektor *kicserélés*nek vagy a t_{rs} pozíción való *pivotálás*nak nevezzük. A t_{rs} elemet *pivot elem*nek, míg az r indexszel jelölt sort *pivot sor*nak, az s indexszel jelölt oszlopot pedig *pivot oszlop*nak nevezzük.

A pivotálási tételt, valós vektorterek esetére fogalmaztuk meg és igazoltuk. A bizonyítás során semmilyen speciális, a valós számokra jellemző tulajdonságot nem használtunk ki, tehát az állítás tetszőleges testek feletti vektorterekre is megfogalmazható és a test illetve a vektortér axiómákat felhasználva igazolható.

A pivotálás során elvégzett aritmetikai műveletek számát, nagyságrendjét határozzuk meg a következő lemmában.

1.25. Lemma. *Legyen adott az* $\{\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_n\} \subset \mathcal{L}$ *véges vektorrendszer, ahol* $\mathcal{L}$ *a valós számok teste feletti vektortér. Legyen adott a T (rövid) pivot tábla, ahol* $|\mathcal{J}_G| = m$ *és* $|\bar{\mathcal{J}}_G| = n - m$.

Ekkor egy pivotálás elvégzéséhez legfeljebb $(n-m-1)\,(m-1)$ *összeadást és legfeljebb* $2\,m\,(n-m) - n + 1$ *szorzást kell elvégezni.*

Bizonyítás. Az 1.24. Tétel négy képletéből csak az elsőben szerepel összeadás. Tehát a rövid pivot tábla elemeinek kiszámítása esetén, legfeljebb egy összeadásra kerül sor azon együtthatók kiszámításakor, amelyek nem szerepelnek sem a pivot sorban, sem a pivot oszlopban. Ilyen elemből összesen $(m-1)\,(n-m-1)$ van, tehát legfeljebb ennyi összeadás lehetséges a pivotálás végrehajtása során.

A rövid tábla, pivot sorában és pivot oszlopában összesen

$$(n-m) + m - 1 = n - 1$$

elem van, amelyekhez egy osztás tartozik. Azoknak az elemeknek a kiszámítása során, amelyek nem a pivot sorban vagy pivot oszlopban vannak két szorzást kell elvégeznünk. Így a szorzások száma legfeljebb

$$2\,(m-1)\,(n-m-1) + n - 1 = 2\,m\,(n-m) - n + 1$$

lehet. (A megadott érték, azért felső korlát, mert a rövid táblában szereplő nulla értékű elemek csökkentik a szükséges szorzások számát.) ∎

A következő feladatban megfogalmazott kérdések segítségével leellenőrizheti, hogy jól értette-e meg a bevezetett alapvető fogalmakat és el tudja-e végezni a pivotálást az 1.24. Tétel felhasználásával.

1.26. Feladat. Legyenek adottak az

$$\mathbf{a}_1^T = (1, \ 1, \ 1)\,, \mathbf{a}_2^T = (2, \ 1, -1)\,, \mathbf{a}_3^T = (-1, \ 6, \ 4)\,, \mathbf{a}_4^T = (-2, \ 1, \ 2)\,,$$

$$\mathbf{a}_5^T = (-6, \ 3, \ 4)\,, \mathbf{a}_6^T = (1, \ 2, \ 1)\,, \mathbf{a}_7^T = (-1, \ 1, \ 0)$$

vektorok.

a.) Vizsgálja meg, hogy az adott rövid pivot táblák közül melyik az $\{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_7\}$ vektorrendszerhez tartozó pivot tábla.

b.) Hány helyen pivotálhatunk a T_1 táblán ?

c.) Adja meg a T_2 (illetve a T_1) rövid pivot tábla felhasználásával a t_{12} és t_{43} (illetve a t_{35} és t_{17}) pivot elemek értékét.

d.) Végezze el a pivotálást a T_2 táblán a t_{72} elemen.

	$\mathbf{a}_5$	$\mathbf{a}_7$	$\mathbf{a}_6$
$\mathbf{a}_1$	1	0	1
$\mathbf{a}_2$	-1	1	1
$\mathbf{a}_4$	1	-1	-1
$\mathbf{a}_3$	-1	0	-1

T_1 (rövid) pivot tábla

	$\mathbf{a}_2$	$\mathbf{a}_3$	$\mathbf{a}_5$
$\mathbf{a}_1$	1	0	1
$\mathbf{a}_6$	0	2	-1
$\mathbf{a}_7$	1	1	2
$\mathbf{a}_4$	-1	1	2

T_2 (rövid) pivot tábla

	$\mathbf{a}_3$	$\mathbf{a}_4$	$\mathbf{a}_5$	$\mathbf{a}_6$	$\mathbf{a}_7$
$\mathbf{a}_1$	1	-1	2	2	-1
$\mathbf{a}_2$	2	2	1	-1	1

T_3 (rövid) pivot tábla

1.5. Lineáris függetlenség, bázis

Az előző részben bevezettük egy adott véges vektorrendszer generáló részrendszerét. A generáló rendszernek, néhány jó tulajdonsága mellett, van egy kifejezetten rossz is, az, hogy adott generáló rendszer esetén, az is előfordulhat, hogy a generált vektorok előállítása nem egyértelmű.

A lineáris függetlenség és a bázis fogalmának bevezetésével, már olyan generáló rendszereket kereshetünk, amelyek lineárisan független vektorokból állnak és így bázist alkotnak. A lineárisan függetlenség illetve a bázis fogalma, a további fejezetekben rendszeresen használt, fontos fogalmak.

1.27. Definíció. A legalább két elemből álló $\{\mathbf{a}_j \mid j \in \mathcal{J}\}$ vektorrendszert *lineárisan független*nek nevezzük, ha nem létezik olyan $\mathbf{a}_r$, $r \in \mathcal{J}$ vektor, amely előáll a többi (azaz az $\{\mathbf{a}_j \mid j \in \mathcal{J} \setminus \{r\}\}$) vektor lineáris kombinációjaként.

Jelölje $\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_m \in \mathbb{R}^m$ az *egységvektor*okat, amelyeket úgy készítünk el, hogy a valós szám m-esek, indexszel jelölt helyére 1-es kerül, míg a többi $(m-1)$ helyre, 0. Könnyen belátható, hogy az így konstruált m darab vektor (valós szám m-es), lineárisan független lesz.

Egy $\{\mathbf{a}_j \mid j \in \mathcal{J}\}$ vektorrendszert *lineárisan összefüggő*nek mondunk, ha nem teljesíti az előző definíciót.

Ha az előző gondolatmenetben szereplő $\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_m \in \mathbb{R}^m$ egységvektorokat, kiegészítjük, egy tetszőleges valós szám m-essel, $\mathbf{a} \in \mathbb{R}^m$, akkor az így előálló

$\{\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_m, \mathbf{a}\}$ véges vektorrendszer ($m+1$ elemű, valós szám m-es), már lineárisan összefüggő lesz, hiszen

$$\mathbf{a} = a_1\,\mathbf{e}_1 + a_2\,\mathbf{e}_2 + ... + a_m\,\mathbf{e}_m,$$

ahol $a_1, a_2, ..., a_m$ a valós szám m-es első, második, ... m. pozícióján álló valós szám.

A következő két lemma az 1.27. Definíció, egyszerű, következménye.

1.28. Lemma. *A legalább két elemből álló $\{\mathbf{a}_j \mid j \in \mathcal{J}\}$ vektorrendszer akkor és csakis akkor lineárisan független, ha valamely $\mathbf{b}$ vektort előállít lineáris kombinációként, akkor azt egyértelműen állítja elő.*

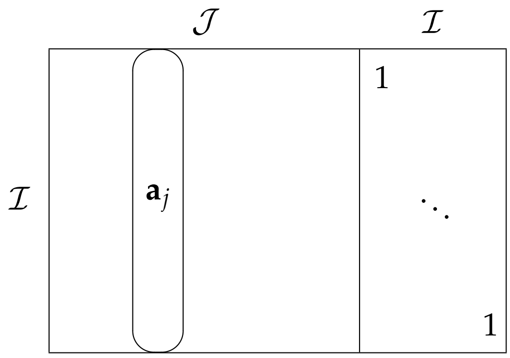
A következő lemma állításának a tagadását szokták a lineáris összefüggőség fogalmának a meghatározására használni.

1.29. Lemma. *Az $\{\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_n\}$ vektorrendszer pontosan akkor lineárisan független, ha a*

$$\sum_{j=1}^{n} \lambda_j\,\mathbf{a}_j = \mathbf{0}$$

előállításból következik, hogy bármely j esetén $\lambda_j = 0$ teljesül.

Egy $\mathcal{J} = \{\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_n\} \subset \mathbb{R}^m$ vektorrendszer lineáris függetlenségét pivotálás segítségével a következő módon tesztelhetjük: egészítsük ki az $\mathcal{I} = \{\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_m\}$ egységvektorokkal, és válasszuk az így keletkező $n+m$ elemű vektorrendszer $\mathcal{J} \cup \mathcal{I}$, generáló rendszerének az egységvektorokat, azaz $\mathcal{J}_G = \mathcal{I}$ lesz.



Az $\mathbf{a}_i$ vektorokat egyesével próbáljuk kicserélni a generáló rendszerben található $\mathbf{e}_j$ egységvektorokkal. A későbbiekben belátjuk, hogy ha sikerül minden $\mathbf{a}_i$ vektort kicserélni valamely $\mathbf{e}_j$ egységvektorral, akkor a kezdeti vektorrendszerünk, $\mathcal{J}$, lineárisan független vektorokból áll. Ha nem sikerült, akkor az alábbi rövid pivot táblához jutunk,

	$\bar{\mathcal{J}}_G$	$\bar{\mathcal{I}}_G$
$\mathcal{J}_G$		
$\mathcal{I}_G$	0	

ahol $\mathcal{J}_G \subset \mathcal{J}$, $\mathcal{I}_G \subset \mathcal{I}$ és $\mathcal{J}_G \cup \mathcal{I}_G$ jelöli a $\mathcal{J} \cup \mathcal{I}$ véges vektorrendszernek a táblán jelölt generáló részrendszerét. A rövid pivot tábla t_{ij} összes olyan pozícióján, amely esetén $i \in \mathcal{I}_G$ és $j \in \bar{\mathcal{J}}_G$ nulla áll, ami pontosan azt jelenti, hogy egyetlen egy $\mathbf{a}_j$, $j \in \bar{\mathcal{J}}_G$ vektor sem cserélhető ki valamely $\mathbf{e}_i$, $i \in \mathcal{I}_G$ vektorral. Sőt az is látható, hogy tetszőleges $\mathbf{a}_j$, $j \in \bar{\mathcal{J}}_G$ vektor esetén

$$\mathbf{a}_j = \sum_{k \in \mathcal{J}_G} t_{kj}\, \mathbf{a}_k + \sum_{i \in \mathcal{I}_G} t_{ij}\, \mathbf{e}_i = \sum_{k \in \mathcal{J}_G} t_{kj}\, \mathbf{a}_k.$$

A második egyenlőség pontosan, azért igaz, mert $t_{ij} = 0$ teljesül, bármely $i \in \mathcal{I}_G$ és $j \in \bar{\mathcal{J}}_G$ esetén.

Tehát, ha az $\mathbf{a}_j$ vektorok $\mathbf{e}_i$ vektorokkal történő kicserélés folyamatában az előző típusú tábla előfordul, akkor az $\mathcal{J} = \{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n\} \subset \mathbb{R}^m$ vektorrendszer lineárisan összefüggő vektorokból áll.

Előfordulhat az az eset is, hogy az összes $\mathbf{e}_i$ vektort kicseréltük valamely $\mathbf{a}_j$ vektorral, azaz a $\mathcal{J} \cup \mathcal{I}$ $n + m$ elemű vektorrendszer aktuális generáló rendszerének minden eleme a $\mathcal{J}$ vektorrendszer elemei közül kerül ki. Ekkor két eset lehetséges: $\bar{\mathcal{J}}_G = \emptyset$ vagy $\bar{\mathcal{J}}_G \neq \emptyset$. A második esetben létezik olyan $\mathbf{a}_k$ vektor, amelyik előáll a többi vektor lineáris kombinációjaként, tehát a $\mathcal{J}$ vektorrendszer, vektorai lineárisan összefüggők.

Ha $\bar{\mathcal{J}}_G = \emptyset$, akkor az összes $\mathbf{a}_j$ vektor bekerült a generáló vektorok közé. Az egységvektorokból álló generáló rendszere a $\mathcal{J} \cup \mathcal{I}$ vektorrendszernek lineáris független vektorokból állt. Tehát, annak az igazolása marad hátra, hogy a pivotálás során előálló generáló rendszer vektorai a kicserélések után is lineáris független vektorokból állnak.

Most vizsgáljuk meg az előbbiekben vázolt eljárás műveletigényét. Nyilvánvaló, hogy legfeljebb $\min\{m, n\}$ pivotálásra lesz szükség. Korábban láttuk, hogy egy pivotálás műveletigénye $\mathcal{O}(mn)$. Összegezve, az előző eljárás művelet igénye $\mathcal{O}(mn\ \min\{m, n\})$.

Ezzel beláttuk a következő lemmát.

1.30. Lemma. *Egy adott $\{\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_n\} \subset \mathbb{R}^m$ vektorrendszer lineáris függetlenségének a tesztelése legfeljebb* $\min\{m, n\}$ *lépésben elvégezhető és az algoritmus művelet igénye*

$$\mathcal{O}(m\, n\ \min\{m, n\}).$$

A leírt számolási eljárást a következő algoritmusban foglaljuk össze.

Algoritmus: a lineáris függetlenség tesztelésére

Bemenő adatok: az $\{\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_n\} \subset \mathbb{R}^m$ vektorok és $\mathcal{J} = \{1, 2, \cdots, n\}$ index halmaz;

az $\mathcal{I} = \{\hat{1}, \hat{2}, \cdots, \hat{m}\}$ az $\mathbb{R}^m$ tér egység vektorainak az index halmaza;

a $T = A = [\mathbf{a}_1 \vdots \mathbf{a}_2 \vdots \cdots \vdots \mathbf{a}_n]$ rövid pivot tábla

Begin
  $\mathcal{J}_G := \mathcal{I}$ és $\bar{\mathcal{J}}_G := \mathcal{J}$;
  while $\mathcal{J} \cap \bar{\mathcal{J}}_G \neq \emptyset$ **do**
    if $\mathcal{I} \cap \mathcal{J}_G \neq \emptyset$ **then**
      if $t_{\hat{i}j} = 0$ $(\forall\, \hat{i} \in \mathcal{I} \cap \mathcal{J}_G, \ \ \forall\, j \in \mathcal{J} \cap \bar{\mathcal{J}}_G)$ **then**
        stop: $\{\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_n\}$ lineárisan összefüggő;
      else
        pivotálás: $\mathcal{J}_G := (\mathcal{J}_G \setminus \{\hat{i}\}) \cup \{j\}$ és $\bar{\mathcal{J}}_G := (\bar{\mathcal{J}}_G \setminus \{j\}) \cup \{\hat{i}\}$, ahol
          $t_{\hat{i}j} \neq 0$ $(\hat{i} \in \mathcal{I} \cap \mathcal{J}_G, \ j \in \mathcal{J} \cap \bar{\mathcal{J}}_G))$;
      endif
    else
      stop: $\{\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_n\}$ lineárisan összefüggő;
    endif
  endwhile
  $\{\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_n\}$ lineárisan független;
end.

Az algoritmus alapötlete, az hogy cseréljünk ki annyi $\mathbf{a}_j$, $j \in \mathcal{J}$ vektort, a generáló rendszerben szereplő valamely $\mathbf{e}_i, i \in \mathcal{I}$ vektorral, amennyit csak lehet, visszatérő eleme lesz több – később tárgyalásra kerülő, – állítás, konstruktív bizonyításnak illetve különböző algoritmusoknak.

A lineáris függetlenség tesztelését mutatjuk be az alábbi példában, felhasználva a pivotálási tételt. Az eljárást a teljes pivot táblán mutatjuk be.

1.31. Példa. Döntsük el, hogy az adott A mátrix oszlop vektorai lineárisan függetlenek-e vagy sem, ahol

$$A = \begin{pmatrix} 0 & 1 & 1 & 2 \\ 1 & 0 & 1 & 2 \\ 2 & 1 & 1 & 4 \\ 0 & 1 & 1 & 2 \\ 1 & 0 & 1 & 2 \end{pmatrix}.$$

Készítsük el az $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \mathbf{a}_4$ és az $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \mathbf{e}_4, \mathbf{e}_5$ vektorokból álló teljes pivot táblát. (Alkalmazzuk az 1.24. Tételt a számítások elvégzésekor.)

$\mathbf{e}_1$	0	1	1	2	1	0	0	0	0
$\mathbf{e}_2$	**1**	0	1	2	0	1	0	0	0
$\mathbf{e}_3$	2	1	1	4	0	0	1	0	0
$\mathbf{e}_4$	0	1	1	2	0	0	0	1	0
$\mathbf{e}_5$	1	0	1	2	0	0	0	0	1

Most legyen $\mathcal{J} \cap \bar{\mathcal{J}}_G = \{\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \mathbf{a}_4\}$ és $\mathcal{I} \cap \mathcal{J}_G = \{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \mathbf{e}_4, \mathbf{e}_5\}$. Kicserélhetjük az $\mathbf{e}_2$ és $\mathbf{a}_1$ vektorokat, a $t_{21} = 1$ elemen keresztül.

	$\mathbf{a}_1$	$\mathbf{a}_2$	$\mathbf{a}_3$	$\mathbf{a}_4$	$\mathbf{e}_1$	$\mathbf{e}_2$	$\mathbf{e}_3$	$\mathbf{e}_4$	$\mathbf{e}_5$
$\mathbf{e}_1$	0	**1**	1	2	1	0	0	0	0
$\mathbf{a}_1$	1	0	1	2	0	1	0	0	0
$\mathbf{e}_3$	0	1	-1	0	0	-2	1	0	0
$\mathbf{e}_4$	0	1	1	2	0	0	0	1	0
$\mathbf{e}_5$	0	0	0	0	0	-1	0	0	1

Ekkor a $\mathcal{J} \cap \bar{\mathcal{J}}_G = \{2, 3, 4\}$ és $\mathcal{I} \cap \mathcal{J}_G = \{\hat{1}, \hat{3}, \hat{4}, \hat{5}\}$ lesz. Egy lehetséges pivot pozíció, amely megfelel algoritmus szabályának a $t_{12} = 1$.

	$\mathbf{a}_1$	$\mathbf{a}_2$	$\mathbf{a}_3$	$\mathbf{a}_4$	$\mathbf{e}_1$	$\mathbf{e}_2$	$\mathbf{e}_3$	$\mathbf{e}_4$	$\mathbf{e}_5$
$\mathbf{a}_2$	0	1	1	2	1	0	0	0	0
$\mathbf{a}_1$	1	0	1	2	0	1	0	0	0
$\mathbf{e}_3$	0	0	$\mathbf{-2}$	-2	-1	-2	1	0	0
$\mathbf{e}_4$	0	0	0	0	-1	0	0	1	0
$\mathbf{e}_5$	0	0	0	0	0	-1	0	0	1

A pivotálással nyert új táblán $\mathcal{J} \cap \bar{\mathcal{J}}_G = \{3,4\}$ és $\mathcal{I} \cap \mathcal{J}_G = \{\hat{3},\hat{4},\hat{5}\}$. Egy újabb lehetséges pivot pozíció a $t_{33} = -2$.

	$\mathbf{a}_1$	$\mathbf{a}_2$	$\mathbf{a}_3$	$\mathbf{a}_4$	$\mathbf{e}_1$	$\mathbf{e}_2$	$\mathbf{e}_3$	$\mathbf{e}_4$	$\mathbf{e}_5$
$\mathbf{a}_2$	0	1	0	1	$\frac{1}{2}$	-1	$\frac{1}{2}$	0	0
$\mathbf{a}_1$	1	0	0	1	$-\frac{1}{2}$	0	$\frac{1}{2}$	0	0
$\mathbf{a}_3$	0	0	1	1	$\frac{1}{2}$	1	$-\frac{1}{2}$	0	0
$\mathbf{e}_4$	0	0	0	0	-1	0	0	1	0
$\mathbf{e}_5$	0	0	0	0	0	1	0	0	1

Végül $\mathcal{J} \cap \bar{\mathcal{J}}_G = \{4\}$ és $\mathcal{I} \cap \mathcal{J}_G = \{\hat{4},\hat{5}\}$. De $t_{44} = 0$ és $t_{54} = 0$, ezért az $\{\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \mathbf{a}_4\}$ vektorok lineárisan összefüggők, és így

$$\mathbf{a}_4 = \mathbf{a}_1 + \mathbf{a}_2 + \mathbf{a}_3,$$

adódik.

A teljes pivot táblán történő számolásnak az a legfontosabb (és talán egyetlen) előnye, hogy nem szükséges az 1.24. Tételben szereplő transzformációs képletekre emlékezni, hanem elegendő azt megérteni, hogy mit reprezentál a teljes pivot tábla és egy generáló illetve generált elem kicserélés milyen számításokat igényel. Mivel a pivotálás műveletének a megértése alapvetően fontos lineáris egyenletrendszerek, lineáris egyenlőtlenségrendszerek és lineáris programozási feladatokat megoldó algoritmusok megértése és alkalmazása során, ezért az előző példa első lépését kielemezzük részletesebben is. Ennek érdekében tekintsük újra az előző példa, első teljes pivot tábláját.

	$\mathbf{a}_1$	$\mathbf{a}_2$	$\mathbf{a}_3$	$\mathbf{a}_4$	$\mathbf{e}_1$	$\mathbf{e}_2$	$\mathbf{e}_3$	$\mathbf{e}_4$	$\mathbf{e}_5$
$\mathbf{e}_1$	0	1	1	2	1	0	0	0	0
$\mathbf{e}_2$	1	0	1	2	0	1	0	0	0
$\mathbf{e}_3$	2	1	1	4	0	0	1	0	0
$\mathbf{e}_4$	0	1	1	2	0	0	0	1	0
$\mathbf{e}_5$	1	0	1	2	0	0	0	0	1

A generáló rendszer elemei pontosan olyan sorrendben vannak felsorolva, amilyen indexű egységvektorral reprezentáltuk azokat. Ennek következtében kezdetben a generáló vektorok felsorolása megegyezik az egységvektorok felsorolásával. A vektorok kicserélése során ez a tulajdonság megmarad, azaz a például a második generáló vektort, mindég a második egységvektorral reprezentáljuk, de megjegyezzük azt, hogy melyik $\mathbf{a}_k$ vektor szerepel másodikként a generáló rendszerben. Ebben az esetben, a $t_{2j},\ j \in \mathcal{J}$ értékek jelölik majd az $\mathbf{a}_j$ vektor lineáris kombinációként való előállításában, a második generáló vektor, $\mathbf{a}_k$, együtthatóját. (Hasonlóan, ha $t_{2j},\ j \in \mathcal{I}$, akkor ez lesz a második generáló vektor, $\mathbf{a}_k$, együtthatója, az $\mathbf{e}_j$ egységvektor előállításában, a jelenlegi generáló rendszert használva.)

Az algoritmusunk az első oszlopban reprezentált $\mathbf{a}_1$ vektort szeretné kicserélni valamelyik generáló rendszerben szereplő $\mathbf{e}_i$ egységvektorral. Ennek az a feltétele, hogy $t_{ij} \neq 0$ teljesüljön. A lehetséges pivot pozíciók közül az elsőt (legkisebb sor indexűt) választjuk. Tehát az $\mathbf{a}_1$ és $\mathbf{e}_2$ vektorok cseréjét szeretnénk elvégezni, a $t_{21} = 1$ pivot pozíció felhasználásával. A kicserélés végén, az $\mathbf{a}_1$ a generáló rendszer másodikként reprezentált eleme lesz, azaz az $\mathbf{a}_1$ vektor oszlopában a második egységvektor kerül. Ehhez a következő számításokat kell elvégeznünk:

- a pivot pozíción szereplő elemet 1-re kell transzformálni,
- a többi pozíción lévő, nem nulla elemeket, nulla értékűvé kell transzformálni.

Ezeket a transzformációkat, természetesen, ki kell terjeszteni a teljes pivot tábla, megfelelő sorára (soraira). Az elimináció során használt tarnszformációkat elemi sor transzformációknak nevezzük. (Ezeket a transzformációkat, mátrixos formában is megadhatjuk. Az elemi sor transzformációkhoz tartozó mátrixokat, *elemi mátrix*oknak nevezzük. Segítségükkel, a pivotálás műveletét, elemi mátrixokkal való mátrixszorzások sorozataként is megadhatjuk a teljes pivot táblán.)

Mivel a $t_{21} = 1$, ezért a pivot sor transzformálására nincsen szükség.

Figyelembe véve, hogy $t_{31} = 2$ és $t_{51} = 1$ és az új generáló rendszerben ezek helyén nulla kell, hogy álljon a következő elemi sor transzformációkat kell végrehajtanunk:

- a második sort megszorozzuk -2-vel és hozzáadjuk a harmadik sorhoz,
- a második sort megszorozzuk -1-gyel és hozzáadjuk az ötödik sorhoz.

Ezek elvégzése után jutunk ahhoz az 1.31. Példa második teljes pivot táblájához, amelyben a generáló rendszer elemei rendre $\mathbf{e}_1, \mathbf{a}_1, \mathbf{e}_3, \mathbf{e}_4, \mathbf{e}_5$.

A lineáris algebra alapvető tétele, a Steinitz-féle kicserélési lemmaként ismerté vált állítás, amelyet a pivotálás segítségével igazolunk.

1.32. Tétel. *(Steinitz-tétel, 1913.) Legyen $\{\mathbf{a}_j \mid j \in \mathcal{J}\}$ vektorrendszer, és jelölje $\mathcal{J}_F$ egy tetszőleges lineárisan független, míg $\mathcal{J}_G$ egy tetszőleges generáló rendszerének az indexhalmazát. Ekkor*

$$|\,\mathcal{J}_F\,| \leq |\,\mathcal{J}_G\,|$$

teljesül.

Bizonyítás. Feltehetjük, hogy $\mathcal{J} = \mathcal{J}_G \cup \mathcal{J}_F$ (ugyanis, ha a $\mathcal{J}_G \cup \mathcal{J}_F \subset \mathcal{J}$, de $\mathcal{J}_G \cup \mathcal{J}_F \neq \mathcal{J}$ lenne, akkor a $\mathcal{J}_G$ természetesen a $\mathcal{J}_G \cup \mathcal{J}_F$ index halmazhoz tartozó vektoroknak is a generáló rendszerét alkotja). A vektorrendszert reprezentáljuk a rövid generáló táblájával.

$$
\begin{array}{cc|ccc|}
 & & & \mathcal{J}_F \cap \bar{\mathcal{J}}_G & \\
 & & & \mathbf{a}_j & \\
\cline{3-5}
 & & & \vdots & \\
\mathcal{J}_G \cap \mathcal{J}_F & & & \vdots & \\
 & & & \vdots & \\
\cline{3-5}
 & & & \vdots & \\
\mathcal{J}_G \cap \bar{\mathcal{J}}_F & \mathbf{a}_i & \dots & t_{ij} & \dots \\
 & & & \vdots & \\
\cline{3-5}
\end{array}
$$

A fenti tábla az $\{\mathbf{a}_j \mid j \in \bar{\mathcal{J}}_G\}$ vektorok előállítását mutatja az $\{\mathbf{a}_i \mid i \in \mathcal{J}_G\}$ generáló vektorokkal.

Pivotáljunk a $t_{ij} \neq 0$ helyen, ha $i \in \mathcal{J}_G \cap \bar{\mathcal{J}}_F$ és $j \in \bar{\mathcal{J}}_G \cap \mathcal{J}_F$. Ismételjük addig amíg lehet, jelöle $\mathcal{J}'_G$ az ekkor kapott generáló rendszer indexhalmazát:

	$\mathcal{J}_F \cap \bar{\mathcal{J}}'_G$	$\bar{\mathcal{J}}_F \cap \bar{\mathcal{J}}'_G$
$\mathcal{J}'_G \cap \mathcal{J}_F$		
$\mathcal{J}'_G \cap \bar{\mathcal{J}}_F$	0	

Ha $i \in \mathcal{J}'_G \cap \bar{\mathcal{J}}_F$ és $j \in \mathcal{J}_F \cap \bar{\mathcal{J}}'_G$ akkor a $t_{ij} = 0$, hiszen az eljárás leállt. Elvileg, ekkor két eset lehetséges: $\bar{\mathcal{J}}'_G \cap \mathcal{J}_F = \emptyset$ vagy $\bar{\mathcal{J}}'_G \cap \mathcal{J}_F \neq \emptyset$. Ha

$$\bar{\mathcal{J}}'_G \cap \mathcal{J}_F = \emptyset, \tag{1.16}$$

akkor készen vagyunk, mert az összes $\mathcal{J}_F$-beli vektor benne van a generáló rendszerben, vagyis $|\mathcal{J}_F| \leq |\mathcal{J}_G|$.

Bebizonyítjuk, hogy a második eset $\bar{\mathcal{J}}'_G \cap \mathcal{J}_F \neq \emptyset$ nem lehetséges.

Ismét eset szétválasztást alkalmazunk, hiszen újra két esetet kell megvizsgálnunk: $\mathcal{J}'_G \cap \mathcal{J}_F = \emptyset$ vagy $\mathcal{J}'_G \cap \mathcal{J}_F \neq \emptyset$. Az első esetben $\bar{\mathcal{J}}'_G \cap \mathcal{J}_F \neq \emptyset$ és $\mathcal{J}'_G \cap \mathcal{J}_F = \emptyset$, így a következő rövid pivot táblához jutunk

	$\mathcal{J}_F \cap \bar{\mathcal{J}}'_G$	$\bar{\mathcal{J}}_F \cap \bar{\mathcal{J}}'_G$
$\mathcal{J}'_G \cap \bar{\mathcal{J}}_F$	0	

azaz $\bar{\mathcal{J}}'_G \cap \mathcal{J}_F \neq \emptyset$ miatt a zérusvektor eleme lenne a független vektorok rendszerének, ami ellenmondást ad.

$\bar{\mathcal{J}}'_G \cap \mathcal{J}_F \neq \emptyset$ és $\mathcal{J}'_G \cap \bar{\mathcal{J}}_F \neq \emptyset$ összefüggéseket felhasználva azt kapjuk, hogy a $\bar{\mathcal{J}}'_G \cap \mathcal{J}_F$ indexekhez tartozó vektorokat a $\mathcal{J}'_G \cap \mathcal{J}_F$ indexekhez tartozókkal állítottuk elő, ami ellentmond a $\mathcal{J}_F$ indexhalmazhoz tartozó vektorok függetlenségének. ■

Miután megteremtettük, egy adott véges vektorrendszer, lineárisan független és generáló rendszereinek az elemszáma közötti kapcsolatot a Steinitz-tétel igazolásával, készen állunk a bázis fogalmának a bevezetésére.

1.33. Definíció. Legyenek $\{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n\} \subset \mathbb{R}^m$ tetszőleges vektorok és $\mathcal{J} = \{1, 2, \ldots, n\}$ index halmazuk, valamint $\mathcal{J}_B \subset \mathcal{J}$. Az $\{\mathbf{a}_i \mid i \in \mathcal{J}_B\}$ vektorrendszert, az $\{\mathbf{a}_j \mid j \in \mathcal{J}\}$ vektorrendszer *bázis*ának nevezzük, ha lineárisan független és generáló rendszere az $\{\mathbf{a}_j \mid j \in \mathcal{J}\}$ vektorrendszernek.

Tetszőleges vektornak, egy adott bázissal történő előállítása már egyértelmű.[1]
A Steinitz-tételből következik az alábbi tétel a bázisok elemszámára.

1.34. Következmény. *(Bázis tétel.) Legyen adott $\{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n\} \subset \mathbb{R}^m$ véges vektorrendszer és jelölje a vektorok indexeinek a halmazát $\mathcal{J}$. Ha $\mathcal{J}'_B, \mathcal{J}''_B \subset \mathcal{J}$ két tetszőleges bázisának az indexhalmaza, akkor* $\mid \mathcal{J}'_B \mid = \mid \mathcal{J}''_B \mid$.

A bázis előnye a generáló rendszerrel szemben az, hogy a bázis segítségével a nem bázis vektorok előállítása egyértelmű. Ez a bázis vektorok lineáris függetlensége miatt van így.
Mivel a bázisok elemszáma azonos, bevezethetjük a véges vektorrendszer rangját.

1.35. Definíció. Egy vektorrendszer *rang*ja egyenlő egy tetszőleges bázisának elemszámával.

Legyen adott $\{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n\} \subset \mathbb{R}^m$ véges vektorrendszer és jelölje a vektorok indexeinek a halmazát $\mathcal{J}$. Ekkor az adott véges vektorrendszer rangját $\operatorname{rang}(\mathcal{J})$ jelöli.

A generáló rendszerek, lineárisan függetlenség, bázis és rang fogalmak tárgyalását két érdekes, a véges vektorrendszerek lineáris függetlenségével és rangával kapcsolatos feladattal zárjuk le.

[1] Ez generátor rendszerek esetén nem volt így. Az egyértelmű előállításhoz a generáló rendszer elemeinek a függetlensége is kell.

1.36. Feladat. Legyen $\mathcal{J}_F', \mathcal{J}_F'' \subseteq \mathcal{J}$ lineárisan független vektorok két, különböző index halmaza. Bizonyítsa be, hogyha

$$|\mathcal{J}_F'| < |\mathcal{J}_F''|$$

akkor létezik olyan $k \in \mathcal{J}_F'' \setminus \mathcal{J}_F'$ index, hogy $\mathcal{J}_F' \cup \{k\}$ is lineárisan független részrendszerét definiálja az adott vektorrendszernek.

1.37. Feladat. Legyen adott az $\{\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_n\} \subset \mathbb{R}^m$ vektorrendszer és az indexhalmazát jelölje $\mathcal{J}$. Tetszőleges $\mathcal{J}_1 \subseteq \mathcal{J}$ és $\mathcal{J}_2 \subseteq \mathcal{J}$ esetén

$$\mathrm{rang}(\mathcal{J}_1 \cup \mathcal{J}_2) \leq \mathrm{rang}(\mathcal{J}_1) + \mathrm{rang}(\mathcal{J}_2),$$

sőt

$$\mathrm{rang}(\mathcal{J}_1 \cup \mathcal{J}_2) + \mathrm{rang}(\mathcal{J}_1 \cap \mathcal{J}_2) \leq \mathrm{rang}(\mathcal{J}_1) + \mathrm{rang}(\mathcal{J}_2)$$

egyenlőtlenség is fennáll.

1.6. Vektor rendszerek rangja

Az előző részben tárgyalt fontos fogalmakat (generáló rendszer, lineáris függetlenség, bázis és rang) kiterjeszthetjük nem véges elemszámú vektorrendszerekre is.

Az egyik lehetséges módja a kiterjesztésnek, – a bennünket érdeklő esetben, a $\mathbb{R}^m$ vektortér esetén, – a végesen generált lineáris vektorterek fogalmán keresztül történhet meg. Tekintettel arra, hogy nem célunk a lineáris algebra felépítése, hanem csak a számunkra fontos fogalmak és eredmények, minél rövidebb összefoglalása, ezért a felépítés részleteit nem dolgozzuk ki. Az érdeklődő olvasó figyelmét felhívjuk a Tankönyvtárban elérhető Wettl Ferenc, Lineáris algebra című könyvére.

Megmutatjuk, hogy ha a generáló rendszernek egy bázist választunk, akkor pivotálásokkal át tudunk térni egy tetszőleges másik bázisra.

1.38. Lemma. *Legyen adott az $\{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n\} \subset \mathbb{R}^m$ vektorrendszer és az indexhalmazát jelölje $\mathcal{J}$. A $\mathcal{J}_B \subset \mathcal{J}$ és $\mathcal{J}_B' \subset \mathcal{J}$ az $\{\mathbf{a}_j \mid j \in \mathcal{J}\} \subset \mathbb{R}^m$ vektorok két különböző bázis index halmaza. Ekkor a $\mathcal{J}_B'$ bázisról egy pivotálással áttérhetünk, olyan $\mathcal{J}_B''$ bázisra, amelyre*

$$| \mathcal{J}_B \cap \mathcal{J}_B'' | = | \mathcal{J}_B \cap \mathcal{J}_B' | + 1,$$

teljesül.

Bizonyítás. Adjuk meg a véges vektorrendszer rövid pivot tábláját, a $\mathcal{J}'_B$ bázisa segítségével.

Ha találunk egy $t_{kl} \neq 0$ elemet, ahol $k \in \mathcal{J}'_B \setminus \mathcal{J}_B$ és $l \in \bar{\mathcal{J}}'_B \cap \mathcal{J}_B$, akkor ezen az elemen pivotálva egy új bázishoz jutunk: $\mathcal{J}''_B = (\mathcal{J}'_B \setminus \{k\}) \cup \{l\}$. Erre a bázisra $|\mathcal{J}''_B \cap \mathcal{J}_B| = |\mathcal{J}'_B \cap \mathcal{J}_B| + 1$ adódik.

Indirekt tegyük fel, hogy bármely $k \in \mathcal{J}'_B \setminus \mathcal{J}_B$ és $l \in \mathcal{J}_B \cap \bar{\mathcal{J}}'_B$ esetén $t_{kl} = 0$.

	$\bar{\mathcal{J}}'_B \cap \mathcal{J}_B$	$\bar{\mathcal{J}}'_B \setminus \mathcal{J}_B$
$\mathcal{J}'_B \setminus \mathcal{J}_B$	0	
$\mathcal{J}'_B \cap \mathcal{J}_B$		

Ekkor két eset lehetséges: $\mathcal{J}'_B \cap \mathcal{J}_B = \emptyset$ vagy $\mathcal{J}'_B \cap \mathcal{J}_B \neq \emptyset$.

Az első esetben, mivel $\mathcal{J}'_B \cap \mathcal{J}_B = \emptyset$ akkor, $\mathbf{a}_l = \mathbf{0}$, $l \in \bar{\mathcal{J}}'_B \cap \mathcal{J}_B$, ami ellentmondás, mert a $\mathcal{J}_B$ bázisnak nem lehet eleme a nullvektor.

A második esetben, $\mathcal{J}'_B \cap \mathcal{J}_B \neq \emptyset$ és a táblán látható állapot áll fenn, akkor bármely $\mathbf{a}_l$, $l \in \bar{\mathcal{J}}'_B \cap \mathcal{J}_B$ vektor, kifejezhető, az $\mathbf{a}_i$, $i \in \mathcal{J}'_B \cap \mathcal{J}_B$ vektorok lineáris kombinációjaként, ami ellentmond $\mathcal{J}_B$ függetlenségének. ∎

Az előző lemma ismételt alkalmazásával látható, hogy a bázisok pivotálásokkal egymásba transzformálhatók. Ennek számos hasznos következménye lesz.

Az $\mathbb{R}^m$ tér, tetszőleges, véges vektorrendszere által generált lineáris altér, a véges vektor rendszer bármelyik bázisának segítségével, egyértelműen megadható.

1.39. Definíció. Az olyan generáló táblát, amelyben a generáló rendszer egy bázis, *bázis táblá*nak nevezzük.

1.40. Példa. Tekintsük az $\{\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_8\} \subset \mathbb{R}^3$ vektorrendszert, amelyet a $\mathcal{J}_B = \{1, 2, 3\}$ (rövid) bázis táblával adtunk meg. El szeretnénk jutni a $\mathcal{J}'_B = \{4, 7, 8\}$ bázissal adott rövid pivot táblához. Ezt elérhetjük a következőképpen.

	$\mathbf{a}_4$	$\mathbf{a}_5$	$\mathbf{a}_6$	$\mathbf{a}_7$	$\mathbf{a}_8$
$\mathbf{a}_1$	-3	3	-3	3	-2
$\mathbf{a}_2$	3	-5	2	0	2
$\mathbf{a}_3$	2	-3	1	-2	**1**

	$\mathbf{a}_1$	$\mathbf{a}_5$	$\mathbf{a}_6$	$\mathbf{a}_7$	$\mathbf{a}_3$
$\mathbf{a}_4$	1	-3	-1	-1	2
$\mathbf{a}_2$	1	-2	-1	**3**	0
$\mathbf{a}_8$	-2	3	3	0	-3

	$\mathbf{a}_4$	$\mathbf{a}_5$	$\mathbf{a}_6$	$\mathbf{a}_7$	$\mathbf{a}_3$
$\mathbf{a}_1$	**1**	-3	-1	-1	2
$\mathbf{a}_2$	-1	1	0	4	-2
$\mathbf{a}_8$	2	-3	1	2	1

	$\mathbf{a}_1$	$\mathbf{a}_5$	$\mathbf{a}_6$	$\mathbf{a}_2$	$\mathbf{a}_3$
$\mathbf{a}_4$	4/3	-11/3	-4/3	1/3	2
$\mathbf{a}_7$	1/3	-2/3	-1/3	1/3	0
$\mathbf{a}_8$	-2	3	3	0	-3

A pivot táblákon vastagon szedett számmal jelöltük a pivot elemet, amelyek rendre t_{38}, t_{14}, t_{27}, ahol a t_{ij} elemek esetén $i \in \mathcal{J}_B$ és $j \in \bar{\mathcal{J}}_B$.

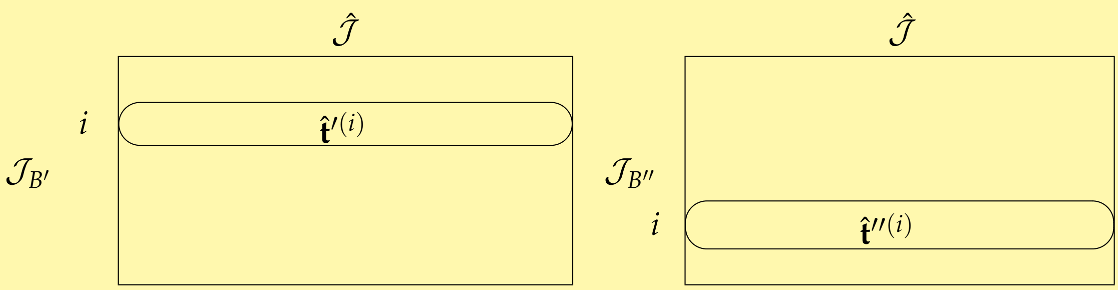
A bázis táblák egymásba pivotálását időnként kényelmesebb lehet hosszú táblán elképzelni. Egy hosszú táblán az, hogy egy elem a bázisban van, onnan látszik, hogy a neki megfelelő oszlopban egy egységvektor található. Így, ha $\mathbf{a}_i$ vektort be szeretnénk cserélni a bázisban lévő $\mathbf{a}_j$ helyére, akkor elemi sor transzformációkkal el kell érnünk, hogy az $\mathbf{a}_i$ oszlopában azt az egységvektort kapjuk meg, amelyik eddig $\mathbf{a}_j$ oszlopában állt.

A következő lemma igazolását az olvasóra bízzuk.

1.41. Lemma. *Legyen a $\mathcal{J}_{B'}$ és $\mathcal{J}_{B''}$ az $\{\mathbf{a}_j \mid j \in \mathcal{J}\}$ vektorok két különböző bázisindex halmaza és $\hat{\mathcal{J}} \subset \mathcal{J}$ tetszőleges, akkor*

$$\mathcal{L}(\hat{\mathbf{t}}'^{(i)} \mid i \in \mathcal{J}_{B'}) = \mathcal{L}(\hat{\mathbf{t}}''^{(i)} \mid i \in \mathcal{J}_{B''}),$$

ahol a $\hat{\mathbf{t}}'^{(i)}$ és $\hat{\mathbf{t}}''^{(i)}$ vektorokat az alábbi ábra mutatja



ahol $(\hat{\mathbf{t}}'^{(i)})_j = t'_{ij}$ és $(\hat{\mathbf{t}}''^{(i)})_j = t''_{ij}$, $j \in \hat{\mathcal{J}}$.

A lineáris algebra egy közismert és fontos tétele a mátrix rang tétel. Mielőtt kimondanánk és bizonyítanánk a tételt, nézzük meg egy konkrét példán a jelentését.

1.42. Példa. Legyen $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \mathbf{a}_4, \mathbf{a}_5$ az $A \in \mathbb{R}^{3\times 5}$ mátrix oszlop vektorai, és jelölje $\mathbf{a}^{(1)}, \mathbf{a}^{(2)}, \mathbf{a}^{(3)} \in \mathbb{R}^5$ az A mátrix sor vektorait. Először írjuk fel a mátrix oszlopvektoraiból képzett vektorrendszerhez tartozó teljes pivot táblát, ami azt jelenti, hogy az egységvektorokat hozzáadjuk a vektorrendszerhez. Az eljárás a szokásos: cseréljünk ki annyi $\mathbf{e}_i$ vektort $\mathbf{a}_j$ vektorral, amennyit csak lehet. Az alábbi, induló bázis táblából pivotálással állítjuk elő az oszlop vektorokból készített vektorrendszer bázisát.

	$\mathbf{a}_1$	$\mathbf{a}_2$	$\mathbf{a}_3$	$\mathbf{a}_4$	$\mathbf{a}_5$	$\mathbf{e}_1$	$\mathbf{e}_2$	$\mathbf{e}_3$
$\mathbf{e}_1$	1	1	1	2	4	1	0	0
$\mathbf{e}_2$	1	1	0	3	1	0	1	0
$\mathbf{e}_3$	1	0	0	−4	−1	0	0	1

	$\mathbf{a}_1$	$\mathbf{a}_2$	$\mathbf{a}_3$	$\mathbf{a}_4$	$\mathbf{a}_5$	$\mathbf{e}_1$	$\mathbf{e}_2$	$\mathbf{e}_3$
$\mathbf{a}_3$	0	0	1	−1	3	1	−1	0
$\mathbf{a}_2$	1	1	0	3	1	0	1	0
$\mathbf{e}_3$	1	0	0	−4	−1	0	0	1

	$\mathbf{a}_1$	$\mathbf{a}_2$	$\mathbf{a}_3$	$\mathbf{a}_4$	$\mathbf{a}_5$	$\mathbf{e}_1$	$\mathbf{e}_2$	$\mathbf{e}_3$
$\mathbf{a}_3$	0	0	1	−1	3	1	−1	0
$\mathbf{a}_2$	0	1	0	7	2	0	1	−1
$\mathbf{a}_1$	1	0	0	−4	−1	0	0	1

Az első bázis táblán az $\mathbf{e}_1$ és $\mathbf{a}_3$ vektorok cseréjénél nem volt szükség elemi sor transzformációkra, mert az $\mathbf{a}_3$ vektor az 1. egységvektor. Ezen a táblán, egy másik pivotálásra is sor kerülhetett, az $\mathbf{e}_2$ és az $\mathbf{a}_2$ vektorok cseréjére. A pivot pozíció a t_{22} és a pivotálás végrehajtása, egy elemi sor transzformációval elvégezhető (a 2. sor kell kivonni az 1. sorból). A második bázistáblán az $\mathbf{e}_3$ és az $\mathbf{a}_1$ vektorok cseréje a t_{31} pivot pozíció segítségével, és egy elemi sor transzformáció alkalmazásával végrehajtható.

Legyen $\hat{\mathbf{a}}^{(1)} = (0, 0, 1, -1, 3)$, és hasonlóan definiáljuk az $\hat{\mathbf{a}}^{(2)}$ és $\hat{\mathbf{a}}^{(3)}$ vektorokat is a harmadik bázistábla sorainak a segítségével. Ekkor

$$\mathcal{L}(\mathbf{a}^{(1)}, \mathbf{a}^{(2)}, \mathbf{a}^{(3)}) = \mathcal{L}(\hat{\mathbf{a}}^{(1)}, \hat{\mathbf{a}}^{(2)}, \hat{\mathbf{a}}^{(3)}), \quad \text{és így} \quad rang(\mathbf{a}^{(1)}, \mathbf{a}^{(2)}, \mathbf{a}^{(3)}) = 3.$$

Másfelől, $\{\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3\}$ vektorok az $\{\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \mathbf{a}_4, \mathbf{a}_5\}$ vektor rendszer bázisát alkotják, tehát

$$rang(\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \mathbf{a}_4, \mathbf{a}_5) = 3.$$

Ezzel beláttuk, hogy az adott $A \in \mathbb{R}^{3\times 5}$ mátrix oszlop illetve sor vektoraiból alkotott vektorrendszerek rangja egyenlő.

Tetszőleges $A \in \mathbb{R}^{m\times n}$ mátrix esetén, két véges vektorrendszert, az $\mathbf{a}_j \in \mathbb{R}^m$ oszlop vektorokból álló n elemű, illetve az $\mathbf{a}^{(i)} \in \mathbb{R}^n$ sor vektorokból álló m elemű vektorrendszert tudjuk kiolvasni. A mátrix rang tételt, az előző példa általánosításával a következő módon fogalmazhatjuk meg: a sor illetve oszlop vektorokból álló vektorrendszerek rangja egyenlő.

1.43. Tétel. *(Mátrix rang tétel.) Tetszőleges*

$$A = (\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n) = (\mathbf{a}^{(1)}, \mathbf{a}^{(2)}, \dots, \mathbf{a}^{(m)})^T \in \mathbb{R}^{m \times n}$$

mátrix esetén

$$rang(\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n) = rang(\mathbf{a}^{(1)}, \mathbf{a}^{(2)}, \dots, \mathbf{a}^{(m)}),$$

azaz a mátrix sor rangja megegyezik a mátrix oszloprangjával.

Bizonyítás. Az A mátrixot, az $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n, \mathbf{e}_1, \dots, \mathbf{e}_m \in \mathbb{R}^m$ vektorokból álló vektorrendszer (rövid) pivot táblájának tekinthetjük, amikor az egység vektorok alkotják a vektorrendszer bázisát.

Alkalmazzuk a szokásos eljárásunkat, azaz cseréljünk ki annyi bázisbeli $\mathbf{e}_i$ vektort, bázison kívüli $\mathbf{a}_j$ vektorral, amennyit csak lehet. Amikor az eljárásunk leáll, az alábbi teljes pivot táblához jutunk.

$$\begin{array}{c|c|c|c|c|}
 & \mathcal{J}_B & \bar{\mathcal{J}}_B & \bar{\mathcal{I}}_B & \mathcal{I}_B \\ \hline
 & 1 & & & \\
\mathcal{J}_B & 1 & \hat{\mathbf{t}}^{(k)} & & 0 \\
 & 1 & & & \\ \hline
 & & & & 1 \\
\mathcal{I}_B & 0 & 0 & & \ddots \\
 & & & & 1 \\ \hline
\end{array}$$

Az ábrán látható tábla mutatja, hogy az A mátrix, sor vektoraiból $\mathbf{a}^{(1)}, \mathbf{a}^{(2)}, \dots, \mathbf{a}^{(m)}$, készített véges vektorrendszer bázisa $\mid \mathcal{J}_B \mid$ elemszámú (ennyi lineárisan független vektor van közöttük) és egy lehetséges bázisát, a $\hat{\mathbf{t}}^{(k)} \in \mathbb{R}^n$, $k \in \mathcal{J}_B$ vektorok reprezentálják. Felhasználjuk az előző, 1.41. Lemmát, amely szerint az $\mathbf{a}^{(1)}, \mathbf{a}^{(2)}, \dots, \mathbf{a}^{(m)}$ és $\hat{\mathbf{t}}^{(k)} \in \mathbb{R}^n$, $k \in \mathcal{J}_B$ véges vektorrendszerek ugyanazt az alteret generálják. Az altér dimenziója, megegyezik a bázis vektorok elemszámával, és így a sor vektorokból alkotott véges vektorrendszer bázisának az elemszámával is, ami nem más, mint a sor vektorokból álló véges vektorendszer rangja, tehát

$$rang(\mathbf{a}^{(1)}, \mathbf{a}^{(2)}, \dots, \mathbf{a}^{(m)}) = \mid \mathcal{J}_B \mid .$$

Másfelől, az A mátrix, oszlop vektoraiból $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n$, készített véges vektorrendszerről, az előállított teljes pivot tábla mutatja, hogy a bázisának elemszáma $\mid \mathcal{J}_B \mid$, hiszen $\mathbf{a}_k \in \mathbb{R}^m$, $k \in \mathcal{J}_B$ vektorok segítségével az A mátrix összes oszlop vektorai előállíthatók. Összegezve,

$$rang(\mathbf{a}^{(1)}, \mathbf{a}^{(2)}, \ldots, \mathbf{a}^{(m)}) = \mid \mathcal{J}_B \mid = rang(\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n).$$ ■

Részrendszerek rangjának meghatározása. Tekintsük az $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n \subset \mathbb{R}^m$ vektorrendszert és az indexhalmazát a $\mathcal{J}$ halmazt. Legyen $\mathcal{J}^* \subset \mathcal{J}$ annak a vektorrendszernek az indexhalmaza, amelynek a rangját meg akarjuk határozni. Legyen a $\mathcal{J}_B \subset \mathcal{J}$ a vektorrendszer egy ismert bázisának az indexhalmaza, és $\bar{\mathcal{J}}_B := \mathcal{J} \setminus \mathcal{J}_B$. Jelölje $\mathcal{J}_B^* = \mathcal{J}^* \cap \mathcal{J}_B$ és $\bar{\mathcal{J}}_B^* = \mathcal{J}^* \cap \bar{\mathcal{J}}_B$. Nyilván $\mathcal{J}^* = \mathcal{J}_B^* \cup \bar{\mathcal{J}}_B^*$.

Írjuk fel a $\mathcal{J}_B$ indexhalmazhoz tartalmazó bázis táblát:

	$\bar{\mathcal{J}}_B^*$	$\bar{\mathcal{J}}_B \setminus \mathcal{J}^*$
$\mathcal{J}_B^*$		
$\mathcal{J}_B \setminus \mathcal{J}_B^*$	$*$	

A szokásos módon, cseréljünk ki minél több vektort, azokon a t_{ij} pozíciókon keresztül, amelyek esetén $i \in \mathcal{J}_B \setminus \mathcal{J}_B^*$ és $j \in \bar{\mathcal{J}}_B^*$. Ezzel az eljárással maximalizálni szeretnénk az $\mathbf{a}_j$, $j \in \mathcal{J}^*$ bázis vektorok számát. Amikor az eljárásunk leáll, akkor az aktuális bázis táblán vagy $(\mathcal{J}_B \setminus \mathcal{J}_B^*) \times \bar{\mathcal{J}}_B^*$ részmátrix minden eleme zérus, vagy $\mathcal{J}_B \setminus \mathcal{J}_B^* = \emptyset$. Mindkét esetben

$$\mathrm{rang}(\mathcal{J}^*) = |\mathcal{J}_B^*|.$$

A vektorrendszerek bázisával és rangjával kapcsolatos részt zárjuk két feladat kitűzésével. A második feladatban bevezetjük a *co-bázis* fogalmát, és ehhez tartozó két állítást mondunk ki.

1.44. Feladat. Legyen a $\mathcal{J}_B$ és $\mathcal{J}_B'$ az $\{\mathbf{a}_j \mid j \in \mathcal{J}\} \subset \mathbb{R}^m$ vektorok két különböző bázis indexhalmaza. Bizonyítsa be, hogy legfeljebb m pivotálással egymásba transzformálhatók a bázisok.

1.45. Feladat. Legyen adott az $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n \subset \mathbb{R}^m$ vektorrendszert és az indexhalmaza a $\mathcal{J}$ halmaz. Valamely $\mathcal{J}_B \subset \mathcal{J}$ bázis esetén az $\{\mathbf{a}_i : i \in \bar{\mathcal{J}}_B\}$ vektorrendszert, az $\{\mathbf{a}_j : j \in \mathcal{J}\}$ vektorok egy *co–bázis*ának nevezzük. Tegyük fel, hogy $\bar{\mathcal{J}}_{B_1}$ és $\bar{\mathcal{J}}_{B_2}$ két co–bázisa az adott vektorrendszernek, ekkor

(i) bármely $s \in \bar{\mathcal{J}}_{B_1} \setminus \bar{\mathcal{J}}_{B_2}$ esetén létezik $r \in \bar{\mathcal{J}}_{B_2} \setminus \bar{\mathcal{J}}_{B_1}$ úgy, hogy $(\bar{\mathcal{J}}_{B_1} \setminus \{s\}) \cup \{r\}$ co–bázis, és

(ii) bármely $r \in \bar{\mathcal{J}}_{B_2} \setminus \bar{\mathcal{J}}_{B_1}$ esetén létezik $s \in \bar{\mathcal{J}}_{B_1} \setminus \bar{\mathcal{J}}_{B_2}$ úgy, hogy $(\bar{\mathcal{J}}_{B_2} \setminus \{s\}) \cup \{r\}$ co–bázis.

1.7. Merőlegesség

A merőlegesség (ortogonalitás), a matematikában általában, illetve a vektorok merőlegessége az Euklideszi terekben, a lineáris algebra fontos és sokat használt fogalma. A mi tárgyalásmódunkban is központi szerepet fog játszani az ortogonalitási tétel, amelyet véges vektorrendszerekkel kapcsolatban mondunk ki. Az ortogonalitási tétel, hatékony eszköze lesz, tételek bizonyításának illetve algoritmusok végességének igazolásakor. A kompozíciós tulajdonsággal kiegészítve, nagyméretű lineáris programozási feladatok szimplex módszerrel, – általában pivot algoritmussal, – való megoldásban, iterációnként, akár többször is, használják. A módosított szimplex módszer megfogalmazása és működése az ortogonalitási tételre és kompozíciós lemmára épül.

1.46. Definíció. Az $\mathbf{a}, \mathbf{b} \in \mathbb{R}^m$ vektorokokat *merőleges*nek (*ortogonális*nak), nevezzük, ha a skaláris szorzatuk nulla, azaz

$$\mathbf{a}^T\mathbf{b} = a_1\, b_1 + a_2\, b_2 + \ldots + a_m\, b_m = 0.$$

Legyen adott $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n \in \mathbb{R}^m$ vektorrendszer, és egy hozzátartozó bázistábla. Jelölje $\mathcal{J}$ a véges vektorrendszer indexhalmazát. Minden bázisbeli $\mathbf{a}_i$ vektorhoz definiálunk egy $\mathbf{t}^{(i)} \in \mathbb{R}^n$, $i \in \mathcal{J}_B$ vektort (a teljes bázis tábla megfelelő sorának a segítségével), amelynek j. eleme legyen t_{ij}.

Hasonlóan, minden nem bázisbeli $\mathbf{a}_j$ vektorhoz definiálunk egy $\mathbf{t}_j \in \mathbb{R}^n$, $j \in \bar{\mathcal{J}}_B$ vektort (a teljes bázis tábla megfelelő oszlopának a segítségével), a következő módon: a $\mathbf{t}_j$ vektor k. eleme, ahol $k \in \mathcal{J}$ legyen

$$\mathbf{t}_{j(k)} = \begin{cases} \mathrm{t}_{kj}, & \text{ha } \mathrm{k} \in \mathcal{J}_B \\ \text{-1}, & \text{ha } \mathrm{k} = \mathrm{j} \\ 0, & \text{különben} \end{cases} \tag{1.17}$$

ahol $\mathcal{J}_B \subset \mathcal{J}$ a bázis vektorok index halmaza és $\bar{\mathcal{J}}_B = \mathcal{J} \setminus \mathcal{J}_B$.

A teljesség igénye miatt, megadjuk a $\mathbf{t}^{(i)} \in \mathbb{R}^n,\ i \in \mathcal{J}_B$ vektor formális definícióját is, a $\mathbf{t}_j,\ j \in \bar{\mathcal{J}}_B$ vektorokéhoz hasonlóan

$$\mathbf{t}_k^{(i)} = \begin{cases} \mathrm{t}_{ik}, & \text{ha } \mathrm{k} \in \bar{\mathcal{J}}_B \\ 1, & \text{ha } \mathrm{k} = \mathrm{i} \\ 0, & \text{különben} \end{cases} \tag{1.18}$$

Figyelni kell arra, hogy például t_{23} nem feltétlenül a bázistábla második sorának harmadik eleme, hanem az $\mathbf{a}_2$ együtthatója az $\mathbf{a}_3$ előállításában. A következő példán szemléltetjük az előbb definiált vektorokat.

1.47. Példa. Tekintsük az $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_5 \in \mathbb{R}^3$ vektor rendszernek az alábbi bázis tábláját.

	$\mathbf{a}_1$	$\mathbf{a}_2$	$\mathbf{a}_3$	$\mathbf{a}_4$	$\mathbf{a}_5$
$\mathbf{a}_3$	0	0	1	−1	3
$\mathbf{a}_2$	0	1	0	7	2
$\mathbf{a}_1$	1	0	0	−4	−1

A fent definiált vektorok a következők lesznek:

$$\mathbf{t}^{(1)} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ -4 \\ -1 \end{pmatrix}, \quad \mathbf{t}^{(2)} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 7 \\ 2 \end{pmatrix}, \quad \mathbf{t}^{(3)} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ -1 \\ 3 \end{pmatrix}, \quad \mathbf{t}_4 = \begin{pmatrix} -4 \\ 7 \\ -1 \\ -1 \\ 0 \end{pmatrix}, \quad \mathbf{t}_5 = \begin{pmatrix} -1 \\ 2 \\ 3 \\ 0 \\ -1 \end{pmatrix}.$$

Megfigyelhetjük, hogy $\mathbf{t}_j^T \mathbf{t}^{(i)} = 0$ teljesül, bármely $i \in \mathcal{J}_B$ és $j \in \mathcal{J}_N$ esetén.

Az előző példa által illusztrált jelenséget általánosítja az ortogonalitási tétel, Klafszky és Terlaky által bevezetett változata.

1.48. Tétel. *(Ortogonalitási tétel.) Tetszőleges $\{\mathbf{a}_j \mid j \in \mathcal{J}\}$ vektorrendszerhez tartozó B' és B'' bázisokra igaz a következő összefüggés*

$$\mathbf{t}'^{(i)^T} \mathbf{t}''_j = 0, \qquad \forall i \in \mathcal{J}_{B'} \text{ és } \forall j \in \bar{\mathcal{J}}_{B''},$$

ahol $\mathcal{J}_{B'}$ és $\mathcal{J}_{B''}$ indexhalmazok a B' illetve B'' bázisokhoz tartoznak.

Bizonyítás. Először megmutatjuk, hogy egy táblán belül igaz az állítás, vagyis a B'' bázishoz tartozó pivot tábla esetén

$$\mathbf{t}''^{(i)^T} \mathbf{t}''_j = 0 \qquad \forall i \in \mathcal{J}_{B''} \text{ és } \forall j \in \bar{\mathcal{J}}_{B''}.$$

Ugyanis a két vektor a következőképpen néz ki:

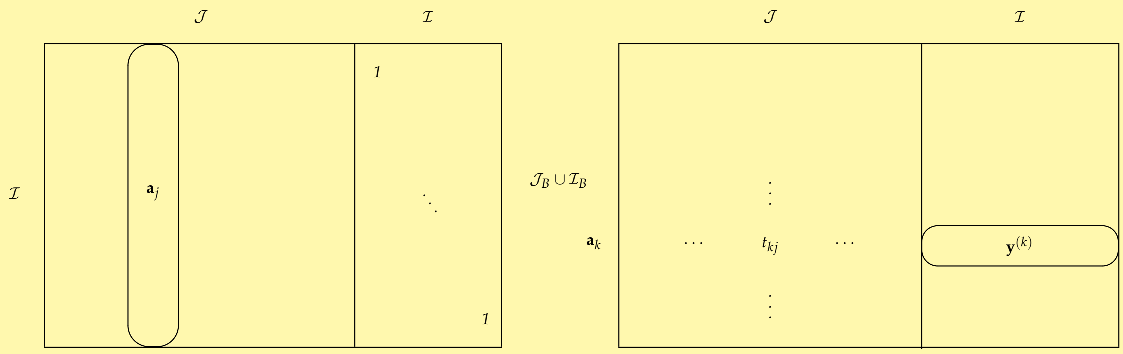
$$\begin{array}{r|ccc|c|ccc|ccc|c|ccc|}
 & & & & i & & & & & & & j & & & \\
\cline{2-16}
\mathbf{t}''^{(i)} = & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 & * & \cdots & * & t''_{ij} & * & \cdots & * \\
\cline{2-16}
\end{array}$$

$$\begin{array}{r|ccc|c|ccc|ccc|c|ccc|}
 & & & & \mathcal{J}_{B''} & & & & & & & \bar{\mathcal{J}}_{B''} & & & \\
\cline{2-16}
\mathbf{t}''_j = & * & \cdots & * & t''_{ij} & * & \cdots & * & 0 & \cdots & 0 & \text{-1} & 0 & \cdots & 0 \\
\cline{2-16}
\end{array}$$

azaz $\mathbf{t}''^{(i)^T} \mathbf{t}''_j = 1\, t''_{ij} + (-1)\, t''_{ij} = 0$. Ebből adódik, hogy a $\mathbf{t}''_j$ merőleges a B'' bázis tábla sorterére, de mivel a bázisok egymásba transzformálhatók, (1.38. Lemma miatt), ezért a B' bázishoz tartozó pivot tábla sor vektorai, a B'' bázis tábla sorvektorainak a lineáris kombinációjaként állíthatók elő. Ennek köszönhetően, a $\mathbf{t}''_j$ vektorok merőlegesek a $\mathbf{t}'^{(i)}$ vektorokra. ∎

Végül következzen az ortogonalitási tétel hasznos következménye, a kompozíciós lemma.

1.49. Következmény. *(Kompozíciós lemma.) Legyenek adottak az $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n$, $\mathbf{e}_1, \ldots, \mathbf{e}_m \in \mathbb{R}^m$ vektorok és jelölje $\mathcal{J}$ az $\mathbf{a}_j$ vektorok illetve $\mathcal{I}$ az egységvektorok indexhalmazát. Legyen $\mathcal{J}_B \cup \mathcal{I}_B$ tetszőleges bázis index halmaza, ahol $\mathcal{J}_B \subset \mathcal{J}$, és $\mathcal{I}_B \subset \mathcal{I}$. Legyen $\mathbf{y}^{(k)} \in \mathbb{R}^m$ a második bázis táblán látható sor vektor:*



Ekkor

$$t_{kj} = \mathbf{y}^{(k)^T} \mathbf{a}_j, \qquad \text{ahol } k \in \mathcal{J}_B \cup \mathcal{I}_B \text{ és } j \in \mathcal{J}.$$

Bizonyítás. Alkalmazzuk az ortogonalitási tételt (1.48. Tétel), az első bázis tábla $j \in \mathcal{J}$ nem bázis vektorára és a második bázis tábla $k \in \mathcal{J}_B \cup \mathcal{I}_B$ bázis vektorára. Tekintsük ezek struktúráját

$$\begin{array}{r|ccc|c|ccc|c|}
 & & & & j & & & & \\
\cline{2-9}
\mathbf{t'}_j = & 0 & \cdots & 0 & \text{-1} & 0 & \cdots & 0 & \mathbf{a}_j \\
\cline{2-9}
\end{array}$$

$$\begin{array}{r|ccc|c|ccc|c|}
 & & & & \mathcal{J} & & & & \mathcal{I} \\
\cline{2-9}
\mathbf{t''}^{(k)} = & * & \cdots & * & t_{kj} & * & \cdots & * & \mathbf{y}^{(k)} \\
\cline{2-9}
\end{array}$$

ekkor $0 = (\mathbf{t}'_j)^T \mathbf{t''}^{(k)} = -t_{kj} + \mathbf{y}^{(k)^T} \mathbf{a}_j$. ■

2. fejezet

Lineáris egyenletrendszerek

Ebben a fejezetben többváltozós lineáris egyenletrendszerek megoldhatóságával, általános és bázis megoldásával, a megoldáshalmaz struktúrájával, a megoldás módszerével (Gauss-Jordan elimináció) és a bázis megoldások méretével foglalkozunk.

A lineáris egyenletrendszerek megoldhatóságának tárgyalásakor a szokásostól eltérő módon egy alternatíva tétel megfogalmazására kerül sor. A tételt először a véges vektorrendszerek körében, bázis táblák struktúrális tulajdonságaléppen fogalmazzuk meg. Klafszky és Terlaky javaslatára Farkas-Minty lemmának nevezzük el, utalva arra, hogy Farkas Gyula illetve George J. Minty matematikusok – ma már egyszerűnek tekintett – alternatíva tételeket dolgoztak ki. Farkas Gyula (1894) lineáris egyenlőtlenségekre illetve George J. Minty (1957) irányított gráfokon megfogalmazott alternatíva tételt közölt.[1] A Farkas-Minty lemmának, a lineáris egyenlőtlenségrendszerek megoldhatóságának a tárgyalásánál, lesz egy un. előjeles változata. Valójában ez a lemma lesz a Farkas illetve a Minty lemmák közös általánosítása.[2]

Ebben a fejezetben tárgyalt Farkas-Minty lemmából vezetjük le a Rouché–Kronecker–Capelli–lemmának nevezett alternatíva tételt, amely a lineáris egyenletrendszerek megoldhatóságáról szól.

2.1. Lineáris egyenletrendszerek megoldhatósága

Bevezetjük a lineáris egyenletrendszer és a lineáris egyenletrendszer megoldhatóságának a fogalmát.

2.1. Definíció. Legyen adott az $A \in \mathbb{R}^{m\times n}$ mátrix és a $\mathbf{b} \in \mathbb{R}^m$ vektor. Az

$$A\,\mathbf{x} = \mathbf{b} \tag{2.1}$$

[1]Klafszky Emil Hálózati folyamok című könyvének a 20. oldalán mondja ki és bizonyítja be a Minty-féle alternatíva tételt.

[2]Persze ehhez, egy kicsit bele kellene bonyolódnunk az irányított matroidok és azokkal kapcsolatos optimalizálási feladatok világába.

egyenletet *n-változós, m egyenletből álló lineáris egyenletrendszer*nek (vagy röviden *lineáris egyenletrendszer*nek) nevezzük. Az $\mathbf{x}$ vektort, a lineáris egyenletrendszer, *ismeretlen* (vagy *változó*) vektorának hívjuk.

Azt mondjuk, hogy a (2.1) lineáris egyenletrendszer *megoldható*, ha létezik $\mathbf{s} \in \mathbb{R}^n$ vektor, amelyre

$$s_1\,\mathbf{a}_1 + s_2\,\mathbf{a}_2 + \cdots + s_n\,\mathbf{a}_n = \mathbf{b}$$

teljesül, ahol az $\mathbf{a}_i$ az A mátrix i. oszlop vektora. Az $\mathbf{x} = \mathbf{s}$ vektort, a *lineáris egyenletrendszer megoldásá*nak nevezzük.

Amennyiben, $\mathbf{b} = \mathbf{0}$ teljesül, akkor *homogén lineáris egyenletrendszer*ről beszélünk. A homogén lineáris egyenletrendszernek mindég van megoldása, mert az $\mathbf{x} = \mathbf{0}$ nyilvánvalóan megoldása az egyenletrendszernek.

A következő módon megfogalmazhatjuk a (2.1) lineáris egyenletrendszer *megoldás halmazá*t

$$\mathcal{A} = \{\mathbf{s} \in \mathbb{R}^n : A\,\mathbf{s} = \mathbf{b}\}.$$

Megmutatható, hogy a lineáris egyenletrendszer megoldás halmaza, *eltolt altér* (vagy másnéven *affin altér*).

2.2. Feladat. Igazoljuk, hogy bármely $A \in \mathbb{R}^{m\times n}$ mátrix esetén a

$$\mathcal{M} = \{\mathbf{s} \in \mathbb{R}^m \mid A\,\mathbf{s} = \mathbf{0}\}$$

lineáris alteret alkot.

Az $A\,\mathbf{x} = \mathbf{b}$ megoldhatóságát az alábbi módon jellemezhetjük, a klasszikus Rouché–Kronecker–Capelli–lemmával, felhasználva a generált altér illetve a véges vektorrendszerek (mátrixok) rangjának a fogalmát.

2.3. Feladat. Legyen $A \in \mathbb{R}^{m\times n}$ és $\mathbf{b} \in \mathbb{R}^m$. Az alábbi állítások ekvivalensek:

(i) az $A\,\mathbf{x} = \mathbf{b}$ megoldható;

(ii) $\mathbf{b} \in \mathcal{L}(\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_n)$ ahol $\mathbf{a}_i$ az A mátrix i. oszlop vektora;

(iii) $rang(\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_n) = rang(\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_n, \mathbf{b})$.

Capelli munkáiban, a 2.3. Feladat (iii) tulajdonságát *nem* teljesítő lineáris egyenletrendszereket, *inkonzisztens* (*nem megoldható*) lineáris egyenletrendszereknek nevezte el. Ezt a gondolatmenetet követve jutunk el annak a kérdésnek a vizsgálatához, hogy milyen további információt nyerhetünk az inkonzisztens lineáris egyenletrendszerek vizsgálatával.

Mielőtt a 2.3. Feladat (i) és (iii) tulajdonságának az ekvivalenciáját, a konzisztens és inkonzisztens (megoldható és nem megoldható) lineáris egyenletrendszerek egységes tárgyalásának érdekében átfogalmazzuk, egy alternatíva tétellé, tegyünk egy kis kitérőt a véges vektorrendszerek világába és fogalmazzunk meg azokra egy alternatíva tételt.

2.4. Lemma. *(Farkas–Minty típusú lemma.) Legyenek* $\{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n, \mathbf{b}\} \subset \mathbb{R}^m$ *adott vektorok és* $\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_m\} \subset \mathbb{R}^m$ *az* $\mathbb{R}^m$ *tér egységvektoraiból álló bázisa. Jelölje* $\mathcal{J}$ *az* $\mathbf{a}_j$ *és* $\mathcal{I}$ *az* $\mathbf{e}_i$ *vektorok index halmazát. Az alábbi két (rövid) bázis tábla közül pontosan az egyik fordulhat elő:*

$$
\begin{array}{c|c|c|c|}
 & \bar{\mathcal{J}}_{B'} & b & \bar{\mathcal{I}}_{B'} \\
\hline
\mathcal{J}_{B'} & & & \\
\hline
\mathcal{I}_{B'} & 0 & \begin{matrix} 0 \\ \vdots \\ 0 \end{matrix} & \\
\hline
\end{array}
$$

$$
\begin{array}{c|c|c|}
 & \bar{\mathcal{J}}_{B''} & \bar{\mathcal{I}}_{B''} \\
\hline
\mathcal{J}_{B''} & & \\
\hline
b & 0 \quad \cdots \quad 0 & \\
\hline
\mathcal{I}_{B''} & 0 & \\
\hline
\end{array}
$$

ahol B' *és* B'' *két különböző bázis az* $\{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n, \mathbf{b}, \mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_m\}$ *vektorrendszernek és a* $\mathcal{J}_{B'}, \mathcal{J}_{B''} \subset \mathcal{J}$ *illetve a* $\mathcal{I}_{B'}, \mathcal{I}_{B''} \subset \mathcal{I}$ *a* B' *és* B'' *bázisokhoz tartozó megfelelő index halmazok. A nem bázis vektorok index halmazát rendre* $\bar{\mathcal{J}}_{B'}, \bar{\mathcal{J}}_{B''}, \bar{\mathcal{I}}_{B'}$*, és* $\bar{\mathcal{I}}_{B''}$ *jelöli.*

Bizonyítás. Először azt kell igazolnunk, hogy a két bázis tábla egyszerre nem fordulhat elő. Ennek igazolásához az 1.48. Tételt (ortogonalitási tételt) használjuk fel. Mindkét bázis táblából, a kitüntetett szerepű **b** vektor oszlopából (nem bázis vektor) illetve sorából (bázis vektor) kiindulva határozzuk meg a $\mathbf{t}_b'$ és $\mathbf{t}_b''$ vektorokat. Az ortogonalitási tétel szerint a két vektor merőleges egymásra. A $\mathbf{t}_b'$ és $\mathbf{t}_b''$ vektorokat az alábbi ábrákkal illusztráljuk

$$\begin{array}{r|c|c|c|c|c|}
 & \mathcal{J}_{B'} & \bar{\mathcal{J}}_{B'} & b & \mathcal{I}_{B'} & \bar{\mathcal{I}}_{B'} \\
\hline
\mathbf{t_b}' = & * \;\cdots\; * & 0 \;\cdots\; 0 & -1 & 0 \;\cdots\; 0 & 0 \;\cdots\; 0 \\
\hline
\end{array}$$

$$\begin{array}{r|c|c|c|c|c|c|c|}
\hline
\mathbf{t_b}'' = & 0 \;\cdots\; 0 & 0 \;\cdots\; 0 & 0 \;\cdots\; 0 & 1 & * \cdots * & 0 \cdots 0 & * \cdots * \\
\hline
 & \multicolumn{3}{c}{\mathcal{J}_{B''}} & & \multicolumn{3}{c}{\mathcal{I}_{B''}} \\
\end{array}$$

Három lényeges dolgot kell észrevennünk a bázis táblák struktúrájának értelmezésekor illetve a $\mathbf{t}_b'$ és $\mathbf{t}_b''$ vektorok meghatározásakor:

- A $\mathbf{t}_b'$ vektor esetén, $t_{ib}' = 0$ teljesül, bármely $i \in \mathcal{I}$ indexre. Ha $i \in \mathcal{I}_{B'}$ akkor az első bázis tábla struktúrája miatt igaz, hogy $t_{ib}' = 0$. Míg, amikor $i \in \bar{\mathcal{I}}_{B'}$ akkor a $\mathbf{t}_b'$ vektor definíciója miatt lesz a $t_{ib}' = 0$.
- A $\mathbf{t}_b''$ vektor esetén, $t_{bj}'' = 0$ teljesül, bármely $j \in \mathcal{J}$ indexre. Ha $j \in \bar{\mathcal{J}}_{B''}$ akkor a második bázis tábla struktúrája miatt igaz, hogy $t_{bj}'' = 0$. Míg, amikor $j \in \mathcal{J}_{B''}$ akkor a $\mathbf{t}_b''$ vektor definíciója miatt lesz a $t_{bj}'' = 0$.
- A $t_{bb}' = -1$ és $t_{bb}'' = 1$, a $\mathbf{t}_b'$ és $\mathbf{t}_b''$ vektorok meghatározása miatt igaz.

Ekkor a

$$\mathbf{t}_b'^{\,T} \mathbf{t}_b'' = \sum_{j \in \mathcal{J}} t_{bj}''\, t_{jb}' + t_{bb}''\, t_{bb}' + \sum_{i \in \mathcal{I}} t_{bi}''\, t_{ib}' = t_{bb}''\, t_{bb}' = -1,$$

de ez ellentmond az ortogonalitási tételnek, tehát egyetlen egy $\{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n, \mathbf{b}\} \subset \mathbb{R}^m$ véges vektorrendszer esetén sem fordulhat elő mindkét bázistábla.

Két tábla közül az egyik fennáll: alkalmazzuk a szokásos eljárásunkat, azaz cseréljünk ki $\mathbf{e}_i$ (bázis) vektort $\mathbf{a}_j$ vektorral ameddig lehet

$$\begin{array}{c|c|c|c|}
 & \bar{\mathcal{J}}_B & b & \bar{\mathcal{I}}_B \\
\hline
\mathcal{J}_B & & & \\
\hline
\mathcal{I}_B & 0 & \begin{matrix} * \\ \vdots \\ * \end{matrix} & \\
\hline
\end{array}$$

Amikor a kicserélési eljárásunk leáll, akkor két eset lehetséges: $\mathcal{I}_B = \emptyset$ vagy $\mathcal{I}_B \neq \emptyset$. Az első esetben nyilván a lemmában szereplő első bázistáblához jutottunk. (A $\mathbf{b}$ vektor előállításához nem szükséges az $\mathbf{e}_i$ egységvektorok közül egy sem. Ezt úgy is megfogalmazhatnánk, hogy $\mathbf{b} \in \mathcal{L}(\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_n)$.)

Ha $\mathcal{I}_B \neq \emptyset$ akkor is két eset lehetséges:

1. Bármely $i \in \mathcal{I}_B$ esetén $t_{ib} = 0$. Ekkor az első bázis táblához jutunk.
2. Létezik $k \in \mathcal{I}_B$ úgy, hogy $t_{kb} \neq 0$. Ekkor a t_{kb} pozíción pivotálva a $\mathbf{b}$ vektor bekerül a bázisba és a második bázis táblához jutunk.

Értelemszerűen, ha $\bar{\mathcal{J}}_B = \emptyset$ és a 2. eset fordul elő, akkor $\mathbf{b}$ vektor nem állítható elő az $\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_n$ vektorok segítségével. (Ezt az esetet kizárhatjuk a tárgyalásunkból, ha feltesszük, hogy $n \geq m$.)

Ezzel beláttuk, hogy a két tábla közül pontosan az egyik fordulhat elő. ∎

Ennyi előkészítés után, készen állunk, hogy megadjuk egy inkonzisztens (nem megoldható) $A\,\mathbf{x} = \mathbf{b}$ lineáris egyenletrendszer, konzisztens (megoldható) párját, amelyet ugyanazon az adatoknak a felhasználásával készítünk el.

2.5. Lemma. *(Rouché-Kronecker-Capelli lemma.) Legyen adott az $A \in \mathbb{R}^{m\times n}$ mátrix és a $\mathbf{b} \in \mathbb{R}^m$ vektor. Az alábbi két lineáris egyenletrendszer közül pontosan az egyik oldható meg:*

$$\left.\begin{aligned} A\mathbf{x} &= \mathbf{b} \end{aligned}\right\} \quad (\mathcal{E}_1) \qquad\qquad \left.\begin{aligned} \mathbf{y}^T A &= \mathbf{0} \\ \mathbf{y}^T \mathbf{b} &= 1 \end{aligned}\right\} \quad (\mathcal{E}_2)$$

Bizonyítás. A kettő egyszerre nem állhat fenn (indirekt bizonyítás): tegyük fel, hogy létezik $\mathbf{x} \in \mathbb{R}^n$ vektor, amelyik megoldja az $(\mathcal{E}_1)$ lineáris egyenletrendszert és létezik $\mathbf{y} \in \mathbb{R}^m$ vektor, amelyik megoldja az $(\mathcal{E}_2)$ lineáris egyenletrendszert. Induljunk ki az $(\mathcal{E}_1)$ lineáris egyenletrendszerből, és annak minden egyenletét szorozzuk meg, a megfelelő y_i értékkel és végül összegezzük az egyenleteket, ekkor az $\mathbf{y}^T A\,\mathbf{x} = \mathbf{y}^T\mathbf{b}$ egyenlethez jutunk. A mátrix szorzás asszociativitását felhasználva a jobboldali kifejezést megbelelően csoportosítva és az $(\mathcal{E}_2)$ rendszer feltételeit kihasználva a következőt kapjuk

$$0 = \mathbf{0}^T\mathbf{x} = (\mathbf{y}^T A)\,\mathbf{x} = \mathbf{y}^T(A\,\mathbf{x}) = \mathbf{y}^T\mathbf{b} = 1$$

Az ellentmondás mutatja, hogy az $(\mathcal{E}_1)$ és $(\mathcal{E}_2)$ lineáris egyenletrendszereknek egyszerre nem lehet megoldásuk.

Alkalmazzuk a Farkas-Minty tipusú lemmát, 2.4. Lemma, az A mátrix oszlop vektoraira és a $\mathbf{b}$ vektorra. Ha az első bázis táblát kapjuk, akkor abból kiolvasható az $(\mathcal{E}_1)$ lineáris egyenletrendszer $\mathbf{x}$ megoldása, a következő formában

$$x_i = \begin{cases} \mathrm{t}_{ib}, & \text{ha } \mathrm{i} \in \mathcal{J}_{B'} \\ 0, & \text{ha } \mathrm{i} \in \bar{\mathcal{J}}_{B'} \end{cases}$$

Ha a második bázis táblát kapjuk, akkor azt célszerű kiegészíteni a teljes bázis táblává

$$
\begin{array}{c|c|c|c|c|c|}
\multicolumn{1}{c}{} & \multicolumn{2}{c}{J} & \multicolumn{1}{c}{b} & \multicolumn{2}{c}{I} \\
\cline{2-6}
\mathcal{J}_{B''} & \begin{matrix}1&&&&\\&\ddots&&&\\&&1&&\\&&&\ddots&\\&&&&1\end{matrix} & & & & 0 \\
\cline{2-6}
b & 0 \ \dots \ 0 & 0 \ \dots \ 0 & 1 & & 0 \ \dots \ 0 \\
\cline{2-6}
\mathcal{I}_{B''} & 0 & 0 & & & \begin{matrix}1&&\\&\ddots&\\&&1\end{matrix} \\
\cline{2-6}
\multicolumn{1}{c}{} & \multicolumn{1}{c}{\mathcal{J}_{B''}} & \multicolumn{1}{c}{\bar{\mathcal{J}}_{B''}} & \multicolumn{1}{c}{} & \multicolumn{1}{c}{\bar{\mathcal{I}}_{B''}} & \multicolumn{1}{c}{\mathcal{I}_{B''}}
\end{array}
$$

A teljes bázis táblából kiolvasható az $(\mathcal{E}_2)$ lineáris egyenletrendszer $\mathbf{y}$ megoldása. Legyen

$$y_i = \begin{cases} 0, & \text{ha } \mathrm{i} \in \mathcal{I}_{B''} \\ \mathrm{t}_{bi}, & \text{ha } \mathrm{i} \in \bar{\mathcal{I}}_{B''} \end{cases}$$

A kompozíciós lemmával, 1.49. Következmény, megmutatható, hogy az $\mathbf{y}$ megoldja az $(\mathcal{E}_2)$ lineáris egyenletrendszert. (Kiolvasható az előző teljes tábla b-vel indexelt sorából.) ■

Eddig inkább a lineáris egyenletrendszerek megoldhatóságával foglalkoztunk, annak ellenére, hogy egy egyszerű eljárást a lineáris egyenletrendszerek megoldására, lényegében, vázoltunk a Farkas-Minty típusú lemma (2.4. Lemma) bizonyításában. Ha azt a gondolatmenetet pontosítjuk egy algoritmussá, akkor a Gauss-Jordan eliminációs módszer egy változatát kapjuk. Mielőtt ezt az algoritmust bemutatnánk és egy példán illusztrálnánk, először újabb rövid kitérőt kell tennünk lineáris algbera területére.

2.2. Gauss-Jordan eliminációs módszer

Először foglalkozzunk olyan lineáris egyenletrendszerek megoldásával, ahol az egyenletek száma megegyezik a változók számával, azaz legyen $A \in \mathbb{R}^{m \times m}$ adott mátrix és $\mathbf{b} \in \mathbb{R}^m$ tetszőleges vektor. Ha az A mátrix oszlop vektorairól feltesszük, hogy lineárisan függetlenek akkor könnyen megmutatható, hogy az $\mathbb{R}^m$ vektortér egy bázisát alkotják. Mivel egy $m \times m$-es mátrixnál lényeges, hogy az oszlop vektorai lineáris függetlenek vagy sem, célszerű bevezetni a következő fogalmakat.

2.6. Definíció. Legyen az $A \in \mathbb{R}^{m \times m}$ mátrix.

1. Az A mátrixot *szinguláris*nak nevezzük, ha az oszlop vektorai lineárisan összefüggnek. A nem szinguláris A mátrixot, *reguláris*nak nevezzük.

2. Az A mátrixot *invertálható*nak nevezzük, ha létezik egy $B \in \mathbb{R}^{m \times m}$ mátrix, amelyre
$$A\ B = B\ A = I$$
teljesül, ahol az $I \in \mathbb{R}^{m \times m}$ egység mátrix. A B mátrixot az A mátrix *inverz*ének nevezzük.

3. Az A mátrix *determináns*án a
$$det(A) = \sum_{\sigma \in S_m} (-1)^{i(\sigma)}\ a_{1\sigma(1)}\ a_{2\sigma(2)}\ \cdots\ a_{m\sigma(m)}$$
számot értjük, ahol S_m az $\{1, 2, \cdots, m\}$ elemek permutációinak a halmazát jelöli, σ egy adott permutáció és $i(\sigma)$ a σ permutáció *inverzió*inak a száma.

Az egység mátrix diagonális elemei 1-esek, a többi eleme nulla. Oszlop vektorai pedig megegyeznek az $\mathbb{R}^m$ tér összes egység vektorával, azaz $\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_m$ vektorokkal.

A permutáció és inverzió fogalmak ismeretét feltételezzük. Ugyanúgy, ahogyan a szinguláris- és reguláris mátrixok, az inverz mátrix és a determináns fogalmának ismeretét is. Ezeket a fogalmakat, azért vezettük be, mert az un. négyzetes lineáris egyenletrendszerek megoldhatóságánál és megoldás módszerénél, előkerülnek és a továbbiakban használni fogjuk.

2.7. Feladat. Legyen $A \in \mathbb{R}^{m \times m}$. A következő állítások ekvivalensek:

(i) Az A mátrix reguláris.

(ii) Az A mátrix invertálható.

(iii) $det(A) \neq 0$.

(iv) Az $A\,\mathbf{x} = \mathbf{b}$ lineáris egyenletrendszer, bármely $\mathbf{b} \in \mathbb{R}^m$ vektor esetén, megoldható.

Reguláris A mátrix, inverz mátrixát, A^{-1} jelöljük. Az előző feladat (iv) pontjában kimondott megoldhatóságnál többet tudunk, elvileg, könnyen kiszámolhatjuk a megoldást, azaz $\mathbf{x} = A^{-1}\,\mathbf{b}$. (Az 'elvileg' szót úgy értettük, hogy az inverz mátrix kiszámítása, numerikusan érzékeny, számítási eljárás, tehát a gyakorlatban, körültekintően kell eljárni.)

Az $A \in \mathbb{R}^{m\times m}$ reguláris mátrix inverzét a következő módon is kiszámolhatjuk: Tekintsük az A mátrix oszlop vektoraiból $\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_m$ álló véges vektorrendszert és egészítsük ki ezt az $\mathbb{R}^m$ tér egység vektoraival, azaz az $\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_m$ vektorokkal. Az így előálló $2\,m$ elemű véges vektorrendszernek írjuk fel a teljes bázis tábláját, úgy, hogy kezdetben az egység vektorok legyenek a bázis elemei. Mivel az A mátrix reguláris, így oszlop vektorai lineárisan függetlenek, ezért az összes $\mathbf{a}_j$ vektor m pivotálás segítségével bevonható valamely $\mathbf{e}_i$ helyére a bázisba. Amikor a számítást befejeztük, akkor az $\mathbf{a}_j$ vektorokat megfelelő egység vektorok reprezentálják. Ha a bázis tábla első $m \times m$-es részén nem az egység mátrix áll, hanem valamilyen permutációja, akkor a sorok megfelelő felcserélésével (elemi sor transzformációkkal), egység mátrixszá tarnszformálható. Az így kapott bázis tábla második $m \times m$-es részén ekkor az A^{-1} mátrix áll.

Az A mátrix determinánsának a kiszámításával, még annyit se foglalkozunk, mint az inverzének a kiszámításával. Ehelyett, feladat formájában megfogalmazunk egy módszert, amellyel a determináns kiszámítható illetve ezzel az eljárással kapcsolatos érdekes tulajdonságot, amely felhívja a figyelmet a hibázás lehetőségére.

2.8. Feladat. Legyen $A \in \mathbb{R}^{m\times m}$ mátrix és jelölje C_{ij} a mátrix a_{ij} eleméhez tartozó előjeles minort. Igazoljuk a következő állításokat:

1. $det(A) = \sum_{j=1}^{m} a_{lj}\,C_{lj}$ és $det(A) = \sum_{i=1}^{m} a_{ik}\,C_{ik}$
   teljesül, bármely $1 \leq l, k \leq m$ esetén.

2. Legyen $1 \leq l, k \leq m$ és $l \neq k$ rögzített indexek, ekkor $\sum_{j=1}^{m} a_{lj}\,C_{kj} = 0.$

Bennünket, első sorban, nem a négyzetes lineáris egyenletrendszerek megoldása érdekel, hanem az általános eset. Azon belül is a következő

$$A\,\mathbf{x} = \mathbf{b}, \quad \text{ahol } A \in \mathbb{R}^{m\times n},\ \mathbf{b} \in \mathbb{R}^m,\ \text{és } m \leq n. \tag{2.2}$$

Tegyük fel azt, hogy $rang(A) = m$. Ekkor az $A \in \mathbb{R}^{m\times n}$ mátrix $m \times m$-es reguláris, A_B részmátrixát *bázis*nak nevezzük. A (2.2) lineáris egyenletrendszert a következő alakban írhatjuk fel

$$A\,\mathbf{x} = A_B\,\mathbf{x}_B + A_N\,\mathbf{x}_N = \mathbf{b},$$

ahol az $A_N \in \mathbb{R}^{m\times(n-m)}$ az A mátrix, nem bázis részmátrixa és $\mathbf{x}_N$ jelöli a nem bázis változók vektorát, míg az $\mathbf{x}_B$ a bázsiváltozókét. Ekkor az $\mathbf{x}_B = A_B^{-1}\,\mathbf{b}, \quad \mathbf{x}_N = \mathbf{0}$ megoldást, a lineáris egyenletrendszer *bázis megoldás*ának nevezzük.

A lineáris egyenletrendszer *általános megoldás*át úgy számíthatjuk ki, hogy az $\mathbf{x}_N = \mathbf{u} \in \mathbb{R}^{n-m}$ tetszőleges vektort, behelyettesítjük a nem bázis változók helyére, majd rendezzük az egyenletet és kifejezzük a bázis változók értékét

$$\mathbf{x}_B = A_B^{-1}\,(\mathbf{b} - A_N\,\mathbf{u}).$$

Az

$$\mathbf{x} = (\mathbf{x}_B, \mathbf{x}_N) = (A_B^{-1}\,(\mathbf{b} - A_N\,\mathbf{u}), \mathbf{u})$$

megoldást a (2.2) lineáris egyenletrendszer általános megoldásának hívjuk.

Az $A\,\mathbf{x} = \mathbf{b}$ egyenlet adatait az alábbi módon foglalhatjuk (rövid) pivot táblába

$$\begin{array}{c|cccc|c|}
 & \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_n & \mathbf{b} \\
\hline
\mathbf{e}_1 & a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\
\mathbf{e}_2 & a_{21} & a_{22} & & a_{2n} & b_2 \\
\vdots & \vdots & \vdots & & \vdots & \vdots \\
\mathbf{e}_m & a_{m1} & a_{m2} & \cdots & a_{mn} & b_m \\
\hline
\end{array}$$

A Gauss-Jordan eliminációs módszert a következő módon magyarázhatjuk el: Az általánosság korlátozása nélkül feltehető, hogy az A mátrixnak nincsen nulla oszlopa. Tekintsük az A mátrix első oszlopát. Ebben az oszlopban található nem nulla elem. Ha az a nem nulla elem, amelyet pivot pozíciónak választottunk ki nem az első helyen áll az oszlopában, akkor elemi sor transzformációval, két sor felcserélésvel, az első pozícióra hozzuk. Ezután az első pozíción elvégezzük a pivotálást, a teljes bázis táblára nézve. Úgy is elmondhatjuk a számítási eljárást, hogy elemi sor transzformációkkal, az első helyen álló elem segítségével, elimináljuk az oszlopban található nem nulla elemeket.

Ez a lépés után, az első sort és első oszlopot, a következő pivot elem kiválasztásakor már nem használjuk.

Ha a második oszlopban, az első sor elemén kívül, van nem nulla elem, akkor ezek közül választunk pivot elemet. A pivot elemet a vizsgált részmátrix bal felső sarkába transzformáljuk (a teljes mátrixra nézve ez a 2. sor 2. eleme), sorok felcserélésével. Ezután, az eliminációt, a második oszlop segítségével, a teljes pivot táblára nézve végezzük el.

Ha a második oszlopban, az első sor elemén kívül, minden elem nulla, – ez az előző eliminációk miatt előfordulhat, – akkor a második oszlopot is elhagyjuk a vizsgálandó részmátrixból (tehát az első két oszlopot kihagyjuk és az első sort is). Az így keletkező részmátrixra megismételjük a pivot elem kiválasztásának az eljárását.

A pivot elem kiválasztását mindaddig folytatjuk, amíg (i) minden egyenlethez rendeltünk pivot pozíciót vagy kimutattuk, hogy az egyenlet redundáns; (ii) találtunk egy ellentmondásos egyenletet.

Tárgyalásunk teljessé tétele érdekében bemutatjuk a Gauss-Jordan eliminációs algoritmus un. pszeudo-kódját és illusztráljuk egy példán is.

Algoritmus: Gauss–Jordan elimináció

Bemenő adatok: $m, n \in \mathbb{N}$; $\mathcal{J} = \{1, 2, \cdots, n\}$ index halmaz; $A \in \mathbb{R}^{m \times n}$ és $\mathbf{b} \in \mathbb{R}^m$.

Begin
- $k := 1$ és $\mathcal{J}_B = \emptyset$;
- **while** $k \leq m$ **do**
  - **if** $a_{kj} = 0 \ (\forall j \in \mathcal{J})$ és $b_k = 0$ **then**
    - a pivot tábla k. sorát töröljük;
    - legyen $m := m - 1$;
  - **else**
    - **if** $a_{kj} = 0 \ (\forall j \in \mathcal{J})$ és $b_k \neq 0$ **then**
      - **stop:** $\nexists \hat{\mathbf{x}} \in \mathbb{R}^n : \ A\,\hat{\mathbf{x}} = \mathbf{b}$;
    - **else**
      - $\exists l \in \mathcal{J} : \ a_{kl} \neq 0$;
      - pivotáljunk a (k, l) elemen és $\mathcal{J}_B := \mathcal{J}_B \cup \{l\}$;
      - $k := k + 1$;
    - **endif**
  - **endif**
- **endwhile**
- **stop:** $\exists \bar{\mathbf{x}} \in \mathbb{R}^n : \ A\,\bar{\mathbf{x}} = \mathbf{b}$;

End.

---

*Redundáns*nak nevezünk egy egyenletet, ha az eliminációs eljárás során, egy aktuális táblán, minden változó együtthatója nulla és a jobboldalon is nulla áll. Az ilyen egyenletet vizsgálatunkból kihagyhatjuk, mert a már vizsgált egyenletek lineáris kombinációjával előállítható, azaz függ az előzőleg vizsgált egyenletektől.

Egy egyenletet *ellentmondásos* egyenletnek nevezünk, ha az eliminációs eljárás során, egy aktuális táblán, minden változó együtthatója nulla, de a jobb oldalon nem nulla áll. (Ezt az esetet írja le a 2.4. Lemma második táblája és ez fordul elő a 2.5. Lemma bizonyításában, akkor amikor azt mutatjuk meg, hogy a második egyenletrendszernek van megoldása.)

Világosan látszik, hogy a Gauss-Jordan eliminációs algoritmus véges. (Legfeljebb n oszlop vizsgálata történik meg, mire m egyenlethez pivot pozíciót rendelünk.) Érdekes, hogy 1965-ben igazolták azt, hogy az algoritmus polinomiális komplexitással rendelkezik. Az algoritmus végrehajtása során $\mathcal{O}(m)$ pivotálást kell végrehajtanunk, és egy pivotálás $\mathcal{O}(mn)$ aritmetikai műveletből áll. Tehát az algoritmus futásideje: $\mathcal{O}(m^2 n)$.

Következzen a példa a Gauss-Jordan eliminációs algoritmus illusztrálására. Ebben a példában azt (is) kérdezzük, hogy a Gauss-Jordan eliminációs eljárás azon túl, hogy megoldja a lineáris egyenletrendszert, ha elvárás, hogy a változók nem negatívak legyenek akkor automatikusan talál ilyen megoldást vagy sem.

2.9. Példa. Oldjuk meg az alábbi feladatot:

$$\begin{array}{rrrrrrcr}
x_1 & -\,x_2 & +x_3 & +2x_4 & +\,x_5 & & = & 0 \\
 & & x_3 & +3x_4 & +3x_5 & +x_6 & = & 5 \\
2x_1 & -2x_2 & +x_3 & +\,x_4 & -\,x_5 & -x_6 & = & -5
\end{array}$$

ahol $x_1, x_2, x_3, x_4, x_5, x_6 \geq 0$.

Induljunk ki a lineáris egyenletrendszerekhez tartozó (rövid) pivot táblából és alkalmazzuk a Gauss-Jordan eliminációs módszert. Az első oszlop első eleme alkalmas lesz pivot elemnek.

x_1	x_2	x_3	x_4	x_5	x_6	b
1	−1	1	2	1	0	0
0	0	1	3	3	1	5
2	−2	1	1	−1	−1	−5

Az első oszlop harmadik elemét kell eliminálni, elemi sor tarnszformáció segítségével. (Az első sort szorozzuk meg −2-vel és adjuk hozzá a harmadik sorhoz.) Az elimináció végrehajtása után jutunk a következő pivot táblához.

	x_1	x_2	x_3	x_4	x_5	x_6	b
x_1	1	−1	1	2	1	0	0
	0	0	1	3	3	1	5
	0	0	−1	−3	−3	−1	−5

Ezen a táblán, az algoritmus a harmadik oszlop, második sorában álló elemet jelöli ki pivot elemnek. A harmadik oszlop, harmadik elemét kell eliminálni, elemi sor tarnszformáció segítségével. (A második sort adjuk hozzá a harmadik sorhoz.)

$$\begin{array}{c|cccccc|c}
 & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & b \\
\hline
x_1 & 1 & -1 & 1 & 2 & 1 & 0 & 0 \\
 & 0 & 0 & 1 & 3 & 3 & 1 & 5 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
\end{array}$$

Végül elimináljuk az első sor, harmadik elemét, azaz vonjuk ki a második sort az elsőből. Ekkor az alábbi táblához jutunk.

$$\begin{array}{c|cccccc|c}
 & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & b \\
\hline
x_1 & 1 & -1 & 0 & -1 & -2 & -1 & -5 \\
x_3 & 0 & 0 & 1 & 3 & 3 & 1 & 5 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
\end{array}$$

Látható, hogy a harmadik egyenlet redundáns, tehát ekhagyható. Ekkor az alábbi bázis táblához jutunk.

$$\begin{array}{c|cccccc|c}
 & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & b \\
\hline
x_1 & 1 & -1 & 0 & -1 & -2 & -1 & -5 \\
x_3 & 0 & 0 & 1 & 3 & 3 & 1 & 5 \\
\hline
\end{array}$$

Kiolvashatjuk a következő bázis megoldást

$$x_1 = -5,\, x_3 = 5, \quad x_2 = x_4 = x_5 = x_6 = 0.$$

2.3. Megoldások mérete

Ebben a részben az a célunk, hogy egy $A\,\mathbf{x} = \mathbf{b}$ rendszer, tetszőleges, bázis megoldásának méretére felső illetve alsó korlátot adjunk. Az eredmény eléréséhez fel kell elevenítenünk néhány lineáris algebrai definíciót és tételt.

2.10. Definíció. Legyen $A \in \mathbb{R}^{m\times m}$ mátrix. Az A mátrixból az i. sor és a j. oszlop törlésével nyert $(m-1)\times(m-1)$-es mátrixot A_{ij} jelöli. A $C_{ij} = (-1)^{i+j}\, det(A_{ij})$ értéket az a_{ij} elemhez tartozó *előjeles minor*ának nevezzük. Legyen

$$M = \begin{pmatrix} C_{11} & C_{12} & \dots & C_{1m} \\ C_{21} & C_{22} & \dots & C_{2m} \\ \vdots & & & \vdots \\ C_{m1} & C_{m2} & \dots & C_{mm} \end{pmatrix}$$

az előjeles minorokból elkészített mátrix. Az $adj(A) = M^T$ mátrixot az A mátrix *adjungált*jának nevezzük.

A következő állítást, az eddig átismételt lineáris algebrai fogalmak és állítások felhasználásával lehet igazolni.

2.11. Feladat. Legyen $A \in \mathbb{R}^{m\times m}$ reguláris mátrix. Igazoljuk a következő állítást:

$$A^{-1} = \frac{adj(A)}{det(A)}.$$

Következzen a Cramer-szabály felelevenítése, amely segítségével négyzetes lineáris egyenletrendszert, (m változó és m egyenlet), tudunk megoldani, abban az esetben, ha a lineáris egyenletrendszer mátrixa reguláris.

2.12. Tétel. *(Cramer-szabály.) Legyen $A \in \mathbb{R}^{m\times m}$ mátrix és $\mathbf{b} \in \mathbb{R}^m$ tetszőleges vektor. Ekkor az*

$$A\,\mathbf{x} = \mathbf{b}$$

lineáris egyenletrendszer megoldható, ha az A mátrix reguláris, és ekkor az

$$x_i = \frac{det(A_i)}{det(A)},$$

teljesül, bármely $i = 1, 2, \cdots, m$ indexre, ahol az $A_i \in \mathbb{R}^{m\times m}$ mátrixot úgy nyertük az A mátrixból, hogy az i. oszlopát, az $\mathbf{a}_i$ vektort, kicseréltük a $\mathbf{b}$ vektorral.

A Cramer-szabály bizonyítását a kedves olvasóra bízzuk.

2.13. Definíció. Legyen $A \in \mathbb{Z}^{m\times n}$, $\mathbf{b} \in \mathbb{Z}^m$ és $m, n \in \mathbb{N}$. Definiáljuk az alábbi módon az L értéket

$$L := \sum_{i=1}^{m}\sum_{j=1}^{n} (\log_2(|a_{ij}|+1)+1) + \sum_{i=1}^{m}(\log_2(|b_i|+1)+1).$$

Ez az érték egy felső korlátot ad az egész együtthatós lineáris egyenletrendszer adatainak a bináris tárolásához szükséges számítógépes memória területre. A feladat adatainak a tárolásához szükséges memória területet a feladat *tárigény*ének nevezzük és bitekben mérjük.

Világosan látható, hogy az általunk definiált L felsőkorlát, hiszen egy elem tárolásához szükséges bitek számára

$$\log_2(|a_{ij}|+1)+1$$

értéket adtuk meg. Ha az $a_{ij} = 0$, akkor ez pontosan 1 bitet jelent, és láthatjuk, hogy a $\log_2$ függvény argumentumában a $+1$ szerepe az, hogy kiszámolható legyen a logaritmikus függvény. Ezzel szemben, ha $a_{ij} \neq 0$, akkor nem volna szükség a $\log_2$ függvény argumentumában a $+1$-re. A külső $+1$ szerepe, általában az, hogy a szám előjelének a tárolásához biztosítson egy bitet.

Érzékelhetjük, hogy az L meghatározásakor használt adat tárolási modell, a legrosszabb lehetséges esetre próbál felkészülni, amikor a lehető legegyszerűbben tároljuk az adatokat, lényegében egymásután felsorolva. Annak ellenére, hogy lineáris egyenletrendszerek adatainak tárolására létezik hatékonyabb módszer is, ez bennünket nem érdekel, mert céljainknak – elméleti számítások elvégzése, – az előzőleg meghatározott L tárigény felsőkorlát, bőven, megfelel.

2.14. Lemma. *Legyen az $A\,\mathbf{x} = \mathbf{b}$ lineáris egyenletrendszer, ahol $A \in \mathbb{Z}^{m\times n}$ mátrix és $\mathbf{b} \in \mathbb{Z}^m$ vektor. Jelölje a C az A mátrixnak egy $k \times k$-as négyzetes részmátrixát, ekkor*

$$|det(C)| \leq 2^L.$$

Bizonyítás. A deteremináns definícióját használva

$$\begin{aligned} |\,det(C)\,| &= |\sum_{\sigma\in S_k} (-1)^{i(\sigma)}\, c_{1\sigma(1)}\, c_{2\sigma(2)} \cdots c_{k\sigma(k)}\,| \\ &\leq \sum_{\sigma\in S_k} |\,c_{1\sigma(1)}|\,|c_{2\sigma(2)}| \cdots |\,c_{k\sigma(k)}\,| \\ &\leq \prod_{i=1}^{k}\prod_{j=1}^{k} (1+|\,c_{ij}\,|) \leq \prod_{i=1}^{m}\prod_{j=1}^{n} (1+|\,a_{ij}\,|) \leq 2^L, \end{aligned}$$

ahol az S_k a k-ad rendű permutációk halmaza, és az $i(\sigma)$ a σ permutáció inverzióinak a száma. Az első egyenlőtlenség nyilvánvaló (számok abszolut értékének összegére és szorzatára vonatkozó szabályok alapján). A második egyenlőtlenség, azért igaz, mert a jobboldalon álló szorzat kifejtésével adódó összeg a baloldalon álló összeg minden tagját tartalmazza. A harmadik egyenlőtlenség egy durva felsőbecslés, hiszen egy a C mátrixot részmátrixként tartalmazó mátrixra írjuk fel a szorzatokat. A negyedik egyenlőtlenség pedig nyilvánvaló az L definíciójából. ■

Az előkészületek után, kimondjuk és igazoljuk a megoldások méretére vonatkozó tételt.

2.15. Tétel. *Tegyük fel, hogy $rang(A) = m$. Ekkor az $A\,\mathbf{x} = \mathbf{b}$ lineáris egyenletrendszer bármely B bázisához tartozó $\mathbf{x}$ megoldás koordinátáira*

$$x_j = 0 \quad \textit{vagy} \quad 2^{-L} \leq |x_j| \leq 2^L$$

teljesül.

Bizonyítás. A bázis megoldás miatt $\mathbf{x}_N = \mathbf{0}$ és így az egyenletrendszer az $A_B\,\mathbf{x}_B = \mathbf{b}$ alakú lesz. A Cramer-szabály miatt

$$x_i = \frac{det(A_i)}{det(A_B)}$$

ahol $i \in I_B$ és az A_i mátrixot az A_B mátrixból úgy nyertük, hogy az i. oszlopát a $\mathbf{b}$ vektorral cseréltük ki. Ha $x_i \neq 0$ akkor

$$|x_i| = \frac{|det(A_i)|}{|det(A_B)|} \leq |det(A_i)| \leq 2^L,$$

ahol $1 \leq |det(A_B)|$ összefüggést használtuk, ami az A_B elemeinek az egészértékűségéből adódik. A második becslés pedig az előző lemma miatt igaz. A fordított becsléshez hasonlóan jutunk:

$$|x_i| = \frac{|det(A_i)|}{|det(A_B)|} \geq \frac{1}{det(A_B)} \geq 2^{-L}.$$

■

3. fejezet

Lineáris egyenlőtlenségrendszerek

Könnyen belátható, hogy bármelyik lineáris egyenlőtlenségrendszer, az

$$A\,\mathbf{x} = \mathbf{b}, \quad \mathbf{x} \geq \mathbf{0} \tag{3.1}$$

alakúra hozható, ahol $A \in \mathbb{R}^{m\times n}$ mátrix és $\mathbf{b} \in \mathbb{R}^m$ vektor. Tehát elegendő, az ilyen típusú lineáris egyenlőtlenségrendszerek, megoldhatóságának a vizsgálatára koncentrálnunk és elegendő az ilyen típusú lineáris egyenlőtlenségrendszerekre kidolgozni megoldási eljárásokat.

Az $\mathbf{x} \geq \mathbf{0}$ feltételt miatt azt mondjuk, hogy a változók, *előjelkötött változó*k.

Ehhez mindössze három esetet kell megvizsgálnunk:

- Valamelyik feltétel egyenlőtlenség.
- Valamelyik változó előjelkötetlen változó.
- Valamelyik változó alsó és/vagy felsőkorláttal rendelkezik.

Először az egyenlőtlenséges feltételeket alakítjuk át egyenlőségekké. Tekintsük a következő egyenlőtlenséget, mondjuk az i. feltételt

$$a_{i1}\,x_1 + a_{i2}\,x_2 + \ldots + a_{in}\,x_n \leq b_i \tag{3.2}$$

vezessük be az s_i eltérés változót, amely méri az eltérést jobboldal és a baloldal között. Ekkor az előző egyenlőtlenséget az alábbi módon írhatjuk le

$$a_{i1}\,x_1 + a_{i2}\,x_2 + \ldots + a_{in}\,x_n + s_i = b_i, \;\; s_i \geq 0.$$

Hasonlóan járunk el a nagyobb-egyenlő típusú egyenlőtlenségek esetén is, csak ott az eltérés változót, annak érdekében, hogy egyenlőséges feltételt kapjunk, levonjuk a baloldalból. Nyilvánvaló, hogy ilyen módon kezelni tudjuk a változókra kirótt alsó és/vagy felső korlátokat is.

Tegyük fel, hogy az y_k egy olyan előjelkötetlen változó, amelyik egy vagy több lineáris feltételben szerepel. Először az összes egyenlőtlenséget alakítsuk át egyenletté. Tekintsünk egy olyan egyenletet, amelyben y_k együtthatója nem nulla. (Ilyen egyenlet nyilván van, mert különben y_k-t nem sorolnánk fel, mint változót.) Ebben az esetben az egyenlet a következő alakú lehet,

$$a_{i1}\, x_1 + a_{i2}\, x_2 + \ldots + a_{in}\, x_n + a\, y_k = b_i$$

ahol $a \neq 0$. Ekkor az y_k változó kifejezhető az egyenletből a többi változó segítségével a következő módon

$$y_k = \frac{b_i}{a} - \frac{a_{i1}}{a}\, x_1 - \frac{a_{i2}}{a}\, x_2 - \ldots - \frac{a_{in}}{a}\, x_n.$$

Az y_k kifejezés értéke, az $x_1, x_2, \ldots, x_n$ változók bármilyen helyettesítési értéke mellett elfogadható lesz, hiszen nincsen rá előjelkötés. Ha az y_k változónak további lineáris egyenletekben is van nem nulla együtthatója, akkor az y_k kifejezését behelyettesítjük, és a szükséges számolások elvégzése után olyan lineáris egyenleteket kapunk, amelyik nem tartalmazza az y_k változót.

Ezt az eljárást megismételve az összes előjelkötetlen változóra, egyesével eliminálhatjuk azokat az egyenletrendszerből. Minden egyes előjelkötetlen változóhoz meg kell jegyeznünk egy egyenlőséget, amelyiknek a segítségével végül kiszámolhatjuk az értékét. Cserébe, minden egyes előjelkötetlen változó eliminálásakor a lineáris egyenletrendszerünkben az egyenletek száma (legalább) eggyel csökken.

Az eljárás végén, csak előjelkötött változók maradnak az egyenletrendszerünkben, tehát a kívánt alakra hoztuk a lineáris egyenlőtlenségrendszerünket.

Tekintsünk egyetlen lineáris egyenlőtlenséget, ahogyan azt a (3.2) feltétellel megadtuk. A (3.2) megoldásainak a halmazát *affin féltér*nek (vagy egyszerűen *féltér*nek) nevezzük, ha $b_i \neq 0$, és az alábbi formában adjuk meg

$$\mathcal{F} = \{\mathbf{x} \in \mathbb{R}^n : a_{i1}\, x_1 + a_{i2}\, x_2 + \ldots + a_{in}\, x_n \leq b_i\}.$$

A (3.1) lineáris egyenlőtlenségrendszer struktúrája a következő: egy lineáris egyenletrendszer és előjelkötött változók.

Az előjelkötött változók halmazát a következő módon definiáljuk

$$\mathbb{R}^n_{\oplus} = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{x} \geq \mathbf{0}\}$$

és *pozitív ortáns*nak nevezzük. (Pontosabb lenne, ha nem-negatív ortánsnak neveznénk.)

A lineáris egyenletrendszer megoldásairól elmondtuk már, hogy un. affin teret alkotnak és a megoldás halmazt tömören a következő módon adhatjuk meg

$$\mathcal{A} = \{\mathbf{x} \in \mathbb{R}^n : A\, \mathbf{x} = \mathbf{b}\},$$

ahol $A \in \mathbb{R}^{m\times n}$ mátrix és $\mathbf{b} \in \mathbb{R}^m$ vektor.

Nyilván, ekkor a (3.1) lineáris egyenlőtlenségrendszer megoldáshalmazát úgy adhatjuk meg, hogy

$$\mathcal{M} = \{\mathbf{x} \in \mathbb{R}^n : A\, \mathbf{x} = \mathbf{b},\ \text{és}\ \mathbf{x} \geq \mathbf{0}\} = \mathcal{A} \cap \mathbb{R}^n_{\oplus}.$$

Mielőtt folytatnánk a (3.1) lineáris egyenlőtlenségrendszer megoldáshalmaza tulajdonságainak az elemzését, elevenítsünk fel néhány egyszerű fogalmat.

3.1. Definíció. Legyen adott az $\{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n\} \subset \mathbb{R}^m$ véges vektorrendszer. Ekkor a $\mathbf{b} = s_1\mathbf{a}_1 + s_2\mathbf{a}_2 + \ldots + s_n\mathbf{a}_n$ vektort, ahol $s_1 + s_2 + \ldots + s_n = 1$ teljesül, az $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n$ vektorok

1. *affin kombináció*jának nevezzük, ha az s_i valós számok;
2. *konvex kombináció*jának nevezzük, ha az s_i nemnegatív valós számok.

Egy $\mathcal{B} \subset \mathbb{R}^m$ halmazt *affin* illetve *konvex* halmaznak nevezzük, ha zárt az affin- illetve konvex kombináció képzésére.

Könnyen megmutatható, hogy elegendő a konvex halmazokat a következő módon definiálni.

3.2. Definíció. A $\mathcal{B} \subset \mathbb{R}^m$ halmazt, *konvex halmaz*nak nevezzük, ha bármely $\mathbf{b}_1, \mathbf{b}_2 \in \mathcal{B}$ és $\lambda \in [0, 1]$ esetén $(1 - \lambda)\,\mathbf{b}_1 + \lambda\,\mathbf{b}_2 \in \mathcal{B}$, teljesül.

Egyszerű érdekes tulajdonságok a következők.

3.3. Feladat.

1. Legyenek $\mathcal{F}$, $\mathcal{A}$, $\mathbb{R}^n_\oplus$, $\mathcal{M} \subset \mathbb{R}^m$ halmazok azok, amelyeket a (3.2) és a (3.1) feltételek segítségével definiáltunk. Igazolja, hogy a $\mathcal{F}$, $\mathcal{A}$, $\mathbb{R}^n_\oplus$, $\mathcal{M}$ konvex halmazok.
2. Legyenek $\mathcal{B}_1, \mathcal{B}_2, \ldots \mathcal{B}_k \subset \mathbb{R}^m$ konvex halmazok. Bizonyítsa be, hogy
$$\bigcap_{i=1}^{k} \mathcal{B}_i \subset \mathbb{R}^m$$
is konvex halmaz, azaz véges sok konvex halmaz metszete is konvex halmaz.

Mielőtt a (3.1) lineáris egyenlőtlenségrendszer, $\mathcal{M}$ megoldás halmazának, érdekes, de elemi struktúrális tulajdonságait megfogalmazzuk, szükségünk lesz még néhány (geometriai) fogalomra.

3.4. Definíció. Legyen $\mathcal{K} \subset \mathbb{R}^m$ halmaz.

1. A $\mathcal{K}$ halmazt *kúp*nak nevezzük, ha bármely $\lambda \geq 0$ és $\mathbf{x} \in \mathcal{K}$ esetén $\lambda\,\mathbf{x} \in \mathcal{K}$.

2. A $\mathcal{K}$ kúpot **0**-*csúcsú kúp*nak nevezzük, ha nem tartalmaz az origón kívül egyetlen egy alteret sem.

3. A $\mathcal{K}$ halmazt *konvex kúp*nak (**0**-*csúcsú konvex kúp*nak) nevezzük, ha konvex halmaz és kúp (**0**-csúcsú kúp) is egyben.

Ezek után könnyen igazolható, hogy a nemnegatív ortáns, az $\mathbb{R}^n_\oplus$ halmaz, **0**-csúcsú konvex kúp.

Most már készen állunk arra, hogy tisztázzuk az $\mathcal{M}$ halmaz struktúrális tulajdonságait. Ehhez már csak egy definícióra van szükségünk.

3.5. Definíció. Véges sok (affin) féltér metszetét *konvex poliéder*nek nevezzük.

Egyszerűen meggondolható, hogy a (3.1) lineáris egyenlőtlenségrendszer, $\mathcal{M}$ megoldás halmaza, előáll, mint $2\,m + n$ féltér metszete, hiszen az $\mathbb{R}^n_\oplus$ halmaz n darab féltér metszete és az $\mathcal{A}$ halmaz úgy is megadható, hogy

$$\mathcal{A} = \{\mathbf{x} \in \mathbb{R}^n : A\,\mathbf{x} \leq \mathbf{b} \quad \text{és} \quad A\,\mathbf{x} \geq \mathbf{b}\},$$

azaz $2\,m$ darab (zárt) féltér metszeteként. Tehát a a (3.1) lineáris egyenlőtlenségrendszer, $\mathcal{M}$ megoldás halmaza, konvex poliéder.

Ezentúl, a (3.1) lineáris egyenlőtlenségrendszer, megoldás halmazát $\mathcal{P}$ jelöli, utalva arra, hogy poliéderről van szó.

3.1. Konvex poliéderek geometriai jellemzése

Beláttuk, hogy a (3.1) lineáris egyenlőtlenségrendszer megoldás halmaza, konvex poliéder, $\mathcal{P}$. Szükségünk lesz a hipersík definíciójára.

3.6. Definíció. Legyen $\mathbf{a} \in \mathbb{R}^n \setminus \{\mathbf{0}\}$ vektor, és a $b \in \mathbb{R}$ adott szám. Ekkor a

$$\mathcal{H} = \left\{ \mathbf{x} \in \mathbb{R}^n : \mathbf{a}^T \mathbf{x} = b \right\}$$

halmazt *hipersík*nak nevezzük.

Egy poliéder csúcsait azzal a tulajdonságukkal tudjuk precízen definiálni, hogy minden csúcshoz található olyan a csúcsot tartalmazó hipersík, amely nem vágja ketté a poliédert.

3.7. Definíció. Legyen $\bar{\mathbf{x}} \in \mathcal{P}$. Az $\bar{\mathbf{x}}$ a $\mathcal{P}$ konvex poliéder *csúcsa*, ha $\exists \mathbf{a} \in \mathbb{R}^n \setminus \{\mathbf{0}\}$: $\mathbf{a}^T\bar{\mathbf{x}} > \mathbf{a}^T\mathbf{z}, \quad \forall \mathbf{z} \in \mathcal{P} \setminus \{\bar{\mathbf{x}}\}$.

Ez a definíció megfelel a szemléletes képnek, ahol az említett hipersík a $\mathbf{a}^T\mathbf{x} = \mathbf{a}^T\bar{\mathbf{x}}$ lesz. Ekkor a $\mathcal{H} = \left\{\mathbf{x} \in \mathbb{R}^n : \mathbf{a}^T\mathbf{x} = \mathbf{a}^T\bar{\mathbf{x}}\right\}$ hipersíkot *támaszhipersík*nak nevezzük. Támaszhipersíknak, általában, lehet több közös pontja egy $\mathcal{P}$ konvex poliéderrel, a lényeges tulajdonság, az, hogy a támaszhipesík által definiált zárt félterek egyikében helyezkedjen el a konvex poliéder. A támaszhipersík a konvex poliédert, valamelyik lapjában metszi, de csak a nulla dimenziós lapja (csúcs) esetén igaz, hogy van olyan hipersík, amelyik csak az adott csúcsot tartalmazza a konvex poliéderből.

3.8. Definíció. Egy $\bar{\mathbf{x}} \in \mathcal{P}$ pontot a $\mathcal{P}$ poliéder *extremális pont*jának nevezzük, ha nem áll elő, $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{P}, \quad \mathbf{x}_1 \neq \mathbf{x}_2$ pontok konvex kombinációjaként.

Egy $\bar{\mathbf{x}} \in \mathcal{P}$ pont esetén azt mondjuk, hogy az $\bar{\mathbf{x}}$ megoldás *használja* az A mátrix $\mathbf{a}_j$ oszlop vektorát, a $\mathbf{b}$ vektor előállítása során, ha $\bar{x}_j > 0$.

Az $\bar{\mathbf{x}} \in \mathcal{P} = \{\mathbf{x} \in \mathbb{R}^n : \ A\,\mathbf{x} = \mathbf{b}, \quad \mathbf{x} \geq \mathbf{0}\}$ esetén vezessük be a következő jelöléseket: $\mathcal{J}_+ := \{j \mid \bar{x}_j > 0\}$ és $\mathcal{J}_0 := \{j \mid \bar{x}_j = 0\}$. A $\mathcal{J}_+$ azon $\mathbf{a}_j$ oszlop vektorok indexeinek halmaza, amelyeket az $\bar{\mathbf{x}}$ megoldás a $\mathbf{b}$ vektor előállításakor használ.

3.9. Definíció. Az $\bar{\mathbf{x}} \in \mathcal{P} = \{\mathbf{x} \in \mathbb{R}^n : \ A\,\mathbf{x} = \mathbf{b}, \ \ \mathbf{x} \geq \mathbf{0}\}$ pontot *bázis megoldás*nak nevezzük, ha az $\{\mathbf{a}_j \in \mathbb{R}^m : \ j \in \mathcal{J}_+\}$ vektorok lineárisan függetlenek.

A következő tétel kapcsolatot teremt a konvex poliéder csúcsa, extremális pontja és a konvex poliédert definiáló (3.1) lineáris egyenlőtlenségrendszer bázis megoldása között.

3.10. Tétel. *Az $\bar{\mathbf{x}} \in \mathcal{P} = \{\mathbf{x} \in \mathbb{R}^n : \ A\,\mathbf{x} = \mathbf{b}, \quad \mathbf{x} \geq \mathbf{0}\}$ pontra az alábbiak állítások ekvivalensek:*

(1) az $\bar{\mathbf{x}}$ csúcsa a $\mathcal{P}$ konvex poliédernek,

(2) az $\bar{\mathbf{x}}$ extremális pontja a $\mathcal{P}$ konvex poliédernek,

(3) az $\bar{\mathbf{x}}$ bázis megoldása a (3.1) lineáris egyenlőtlenségrendszernek.

Bizonyítás. (1) $\Rightarrow$ (2): Mivel $\bar{\mathbf{x}}$ csúcsa a $\mathcal{P}$ konvex poliédernek, ezért $\exists \mathbf{a} \in \mathbb{R}^n \setminus \{\mathbf{0}\}$: $\mathbf{a}^T\bar{\mathbf{x}} > \mathbf{a}^T\mathbf{z}, \ \forall \mathbf{z} \in \mathcal{P} \setminus \{\bar{\mathbf{x}}\}$.

Tegyük fel, hogy $\bar{\mathbf{x}} = \lambda\, \mathbf{z}_1 + (1-\lambda)\, \mathbf{z}_2$, ahol $\mathbf{z}_1, \mathbf{z}_2 \in \mathcal{P}, \ \lambda \in (0,1)$. Ekkor $\mathbf{a}^T\bar{\mathbf{x}} = \lambda\, \mathbf{a}^T\mathbf{z}_1 + (1-\lambda)\, \mathbf{a}^T\mathbf{z}_2$, amiből $\mathbf{z}_1 = \mathbf{z}_2 = \bar{\mathbf{x}}$ következik, mert különben az $\mathbf{a}^T\bar{\mathbf{x}} > \mathbf{a}^T\mathbf{z}_1$ vagy $\mathbf{a}^T\bar{\mathbf{x}} > \mathbf{a}^T\mathbf{z}_2$ ellentmondást okozna.

(2) $\Rightarrow$ (3): Indirekt tegyük fel, hogy az $\bar{\mathbf{x}}$ nem bázis megoldása a (3.1) lineáris egyenlőtlenségrendszernek. Ekkor

$$\mathbf{b} = \sum_{j\in\mathcal{J}} \bar{x}_j\, \mathbf{a}_j = \sum_{j\in\mathcal{J}_0} \bar{x}_j\, \mathbf{a}_j + \sum_{j\in\mathcal{J}_+} \bar{x}_j\, \mathbf{a}_j = \sum_{j\in\mathcal{J}_+} \bar{x}_j\, \mathbf{a}_j$$

adódik az $\mathcal{J}_0$ definíciója miatt. Mivel az $\bar{\mathbf{x}}$ nem bázis megoldás, ezért az $\{\mathbf{a}_j \mid j \in \mathcal{J}_+\}$ vektorok lineárisan összefüggnek. Tehát léteznek $y_i \in \mathbb{R}$ nem mind nulla számok, amelyekre

$$\sum_{i\in\mathcal{J}_+} y_i\, \mathbf{a}_i = 0, \qquad \text{adódik, és ekkor} \qquad \mathbf{b} = \sum_{i\in\mathcal{J}_+} (\bar{x}_i + \lambda\, y_i)\, \mathbf{a}_i,$$

teljesül, bármely $\lambda \in \mathbb{R}$ szám esetén. Definiáljuk a $\mathcal{P}$ egy új $\mathbf{x}(\lambda)$ pontját az alábbi módon

$$x_j(\lambda) = \bar{x}_j + \lambda\, y_j, \ \text{ha} \ j \in \mathcal{J}_+ \qquad \text{és} \qquad x_j(\lambda) = 0, \ \text{ha} \ j \in \mathcal{J}_0.$$

Ekkor a lineáris egyenletek teljesülnek, de az előjelkötés, $\mathbf{x}(\lambda) \geq \mathbf{0}$ biztosításához, olyan $\lambda \in \mathbb{R}$ számot kell meghatároznunk, amelyre

$$x_j(\lambda) = \bar{x}_j + \lambda\, y_j \geq 0, \ \ \forall j \in \mathcal{J}_+, \tag{3.3}$$

teljesül. Az y_j előjele alapján két halmazt definiálhatunk:

$$\mathcal{K}_+ := \{j \in \mathcal{J}_+ \mid y_j > 0\} \quad \text{és} \quad \mathcal{K}_- := \{j \in \mathcal{J}_+ \mid y_j < 0\}. \tag{3.4}$$

Vizsgáljuk meg a (3.3) egyenlőtlenséget először a $j \in \mathcal{K}_+$, ekkor

$$x_j(\lambda) = \bar{x}_j + \lambda\, y_j \geq 0 \iff -\frac{\bar{x}_j}{y_j} \leq \lambda$$

adódik az $y_j > 0$ miatt. Ezeket az összefüggéseket felhasználva definiálhatjuk a

$$\lambda' := \begin{cases} \max\limits_{j\in\mathcal{K}_+} -\frac{\bar{x}_j}{y_j}, & \text{ha} \quad \mathrm{K}_+ \neq \emptyset \\ \qquad 0, & \text{ha} \quad \mathrm{K}_+ = \emptyset \end{cases} \tag{3.5}$$

valós számot. Nyilván $\lambda' \leq 0$ és $\lambda' = 0$ csak akkor teljesül, ha $\mathcal{K}_+ = \emptyset$. Másfelől, tekintsük a $j \in \mathcal{K}_-$, esetet, ekkor

$$x_j(\lambda) = \bar{x}_j + \lambda\, y_j \geq 0 \iff -\frac{\bar{x}_j}{y_j} \geq \lambda$$

adódik az $y_j < 0$ miatt. Ezeket az összefüggéseket felhasználva definiálhatjuk a

$$\lambda'' := \begin{cases} \min\limits_{j \in \mathcal{K}_-} -\frac{\bar{x}_j}{y_j}, & \text{ha} \quad \mathrm{K}_- \neq \emptyset \\ 0, & \text{ha} \quad \mathrm{K}_- = \emptyset \end{cases} \tag{3.6}$$

valós számot. Nyilván $\lambda'' \geq 0$ és $\lambda'' = 0$ csak akkor teljesül, ha $\mathcal{K}_- = \emptyset$. Nyilvánvalóan a λ' és λ'' számok közül legalább az egyik szám nem nulla, hiszen az $y_i \quad (i \in \mathcal{J}_+)$ számok közül legalább egy nem nulla és így a $\mathcal{K}_+$ illetve $\mathcal{K}_-$ index halmazok közül legalább az egyik nem üres.

Ha λ' és λ'' egyike sem nulla, akkor legyen $\lambda := \min\{-\lambda', \lambda''\}$, ellenkező esetben pedig legyen λ egyenlő, a nem nulla értéket felvevő korlát (λ' vagy λ'') abszolut értékével.

Az így kiválasztott $\lambda > 0$ valós szám segítségével előállított $\mathbf{x}(\lambda), \mathbf{x}(-\lambda)$ vektorok elemei a $\mathcal{P}$ konvex poliédernek. Másfelől, könnyen belátható, hogy

$$\bar{\mathbf{x}} = \frac{\mathbf{x}(\lambda) + \mathbf{x}(-\lambda)}{2},$$

ami ellentmond az $\bar{\mathbf{x}}$ extremális pont tulajdonságának, tehát az indirekt feltevésünket el kell vetnünk, azaz $\bar{\mathbf{x}}$ bázis megoldása a (3.1) lineáris egyenlőtlenségrendszernek.

(3) $\Rightarrow$ (1): Mivel az $\bar{\mathbf{x}}$ bázis megoldása a (3.1) lineáris egyenlőtlenségrendszernek, ezért az $\{\mathbf{a}_j : j \in \mathcal{J}_+\}$ véges vektor rendszer lineárisan független vektorokból áll. Definiáljuk az $\mathbf{a} \in \mathbb{R}^n$ vektort a következő módon

$$a_i = \begin{cases} \sum\limits_{k=1}^{m} a_{ki} & \mathrm{i} \in \mathcal{J}_+ \\ \text{- M} & \mathrm{i} \in \mathcal{J}_0 \end{cases}$$

ahol $M \in \mathbb{R}_+$ egy megfelelően rögzített nagy szám, és legyen $b = \sum\limits_{k=1}^{m} b_k$. (Ha racionális együtthatós lenne a (3.1) lineáris egyenlőtlenségrendszer, akkor $M > n\,2^L$ alkalmas választás lenne, ahol L a feladat tárigényére adott felsőkorlát és n a változók száma.)

Megmutatjuk, hogy a $\mathcal{H} = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{a}^T\mathbf{x} = b\}$ hipersík, támaszhipersík az $\bar{\mathbf{x}} \in \mathcal{P}$ pontban, ez pedig azt jelenti, hogy az $\bar{\mathbf{x}}$ a $\mathcal{P}$ konvex poliéder csúcsa.

Nyilvánvaló, hogy $\mathbf{a} \neq \mathbf{0}$. Először belátjuk, hogy $\bar{\mathbf{x}} \in \mathcal{H}$ teljesül, azaz

$$\mathbf{a}^T\bar{\mathbf{x}} = \sum_{i=1}^{n} a_i\,\bar{x}_i = \sum_{i \in \mathcal{J}_+} a_i\,\bar{x}_i = \sum_{i \in \mathcal{J}_+} \left(\sum_{k=1}^{m} a_{ki}\right) \bar{x}_i = \sum_{k=1}^{m} \left(\sum_{i \in \mathcal{J}_+} a_{ki}\,\bar{x}_i\right) = \sum_{k=1}^{m} b_k = b.$$

Legyen $\mathbf{z} \in \mathcal{P}$ és tegyük fel, hogy $\mathbf{z} \neq \bar{\mathbf{x}}$, valamint azt is, hogy $\mathbf{z} \in \mathcal{H}$, azaz $\mathbf{a}^T\mathbf{z} = b$. Ha $\exists i \in \mathcal{J}_0 : z_i > 0$, akkor az $a_i = -M,\ i \in \mathcal{J}_0$ választása miatt $\mathbf{a}^T\mathbf{z} < b$

adódik. Tehát szükségképpen $z_i = 0$ teljesül, bármely $i \in \mathcal{J}_0$ esetén, de ekkor a feltevésünk miatt azt kapjuk, hogy

$$\sum_{k=1}^{m} b_k = b = \mathbf{a}^T \mathbf{z} = \sum_{i \in \mathcal{J}_+} a_i \, z_i = \sum_{k=1}^{m} \left(\sum_{i \in \mathcal{J}_+} a_{ki} \, \bar{x}_i \right).$$

Figyelembe véve, azt, hogy $\mathbf{z} \in \mathcal{P}$, azaz $A\,\mathbf{z} = \mathbf{b}, \;\; \mathbf{z} \geq \mathbf{0}$ és az $\{\mathbf{a}_j : j \in \mathcal{J}_+\}$ véges vektor rendszer lineárisan független vektorokból áll, a

$$\mathbf{b} = \sum_{i \in \mathcal{J}_+} x_i \, \mathbf{a}_i$$

lineáris egyenletrendszernek két különböző megoldása volna, $\bar{\mathbf{x}}$ és $\mathbf{z}$, ami ellentmond az egyenletben szereplő $\mathbf{a}_i$ oszlop vektorok lineáris függetlenségének. Tehát

$$\mathcal{P} \cap \mathcal{H} = \{\bar{\mathbf{x}}\},$$

ami pontosan azt jelenti, hogy a $\mathcal{H}$ hipersík a $\mathcal{P}$ konvex poliéder támasz hipersíkja az $\bar{\mathbf{x}}$ pontban, vagyis az $\bar{\mathbf{x}}$ csúcsa a $\mathcal{P}$ konvex poliédernek. ■

A bizonyításban az $x_j(\lambda) = \bar{x}_j + \lambda\, y_j \geq 0$ előjelkötöttség vizsgálatánál, az y_j előjelétől függően, két *hányados teszt* elvégzésével definiáltuk a λ' és λ'' valós számokat, lásd (3.5) illetve (3.6) kifejezéseket. Az előjelkötöttség teljesülését vagy a megőrzésének a biztosítását, a továbbiakban is, hányados teszt (variánsainak a) segítségével érjük el.

Annak ellenére, hogy a három fogalom ekvivalens, mégis egy érdekes fontos esetet fogalmaz meg a következő definíció.

3.11. Definíció. Ha $|\mathcal{J}_+| < rang(A)$ és az $\{\mathbf{a}_j \in \mathbb{R}^m : \; j \in \mathcal{J}_+\}$ vektorok lineárisan függetlenek, akkor többféleképpen is kiegészíthetők bázissá. Ekkor több bázis állítja elő ugyanazt az extremális pontot (csúcsot). Ilyen esetben azt mondjuk, hogy az $A\mathbf{x} = \mathbf{b}, \;\mathbf{x} \geq \mathbf{0}$ rendszer *degenerált.*

Egy érdekes speciális esethez jutunk, amikor $\mathbf{b} = \mathbf{0}$ teljesül.

3.12. Definíció. Egy $A\,\mathbf{y} = \mathbf{0}, \;\; \mathbf{y} \geq \mathbf{0}$ alakú rendszert *homogén lineáris egyenlőtlenség rendszer*nek nevezünk. Ennek megoldáshalmazát jelölje

$$\mathcal{H} = \{\mathbf{y} \in \mathbb{R}^n \mid A\,\mathbf{y} = \mathbf{0}, \; \mathbf{y} \geq \mathbf{0}\}.$$

Azonnal látható, hogy a homogén lineáris egyenlőtlenség rendszernek, mindég van megoldása, ugyanis az $\mathbf{y} = \mathbf{0}$ mindég eleme a $\mathcal{H}$ halmaznak.

A homogén lineáris egyenlőtlenség rendszer további fontos strukturális tulajdonságait mondja ki a következő állítás, amelynek a bizonyítását az olvasóra bízzuk.

3.13. Állítás. *A $\mathcal{H} \neq \emptyset$, poliedrikus, konvex kúp. Ha $\mathbf{x} \in \mathcal{P}$ és $\mathbf{y} \in \mathcal{H}$ akkor $\mathbf{x} + \kappa\, \mathbf{y} \in \mathcal{P}$ teljesül, bármely $\kappa \in \mathbb{R}_{\oplus}$ esetén.*

A konvex poliéderek geometriai tulajdonságainak a bemutatását egy olyan tétel kimondásával és bizonyításával folytatjuk, amelyik a konvex poliéder tetszőleges pontjának struktúráját, előállításának a módját fogalmazza meg.

3.14. Tétel. *Bármely $\bar{\mathbf{x}} \in \mathcal{P} = \{\mathbf{x} \in \mathbb{R}^n : \; A\,\mathbf{x} = \mathbf{b}, \;\; \mathbf{x} \geq \mathbf{0}\}$ esetén létezik $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_r \in \mathcal{P}$ bázis megoldás, $\lambda_1, \lambda_2, \ldots, \lambda_r$ nemnegatív valós számok, amelyekre $\sum_{i=1}^{r} \lambda_i = 1$ és $\bar{\mathbf{y}} \in \mathcal{H}$ úgy, hogy*

$$\bar{\mathbf{x}} = \sum_{i=1}^{r} \lambda_i\, \mathbf{x}_i \; + \; \bar{\mathbf{y}}. \tag{3.7}$$

Bizonyítás. Legyen $\bar{\mathbf{x}} \in P$ adott. Alkalmazzunk indukciót a $|\mathcal{J}_+|$ alapján:

Kezdőlépés: ha $|\mathcal{J}_+| = 0$, akkor $\bar{\mathbf{x}} = \mathbf{0}$. Mivel a nullvektor egyrészt triviálisan bázis megoldás, másrészt eleme $\mathcal{H}$ halmaznak, ezért adódik $\bar{\mathbf{x}}$ előállítása. (Mellesleg, ebben az esetben $\mathbf{b} = \mathbf{0}$ is teljesül.)

Indukciós lépés: legyen $\bar{\mathbf{x}} \in \mathcal{P}$ olyan, hogy $|\mathcal{J}_+| = k$. Ha az $\{\mathbf{a}_j \in \mathbb{R}^m : \; j \in \mathcal{J}_+\}$ vektorok lineárisan függetlenek akkor készen vagyunk, mert ekkor $\bar{\mathbf{x}}$ bázis megoldás, azaz az $\bar{\mathbf{x}} = \bar{\mathbf{x}} + \mathbf{0}$ előállítás megfelelő. Ellenkező esetben léteznek $y_1, y_2, \ldots, y_k$ nem mind nulla valós számok, amelyekre

$$\sum_{i \in \mathcal{J}_+} y_i\, \mathbf{a}_i = \mathbf{0}$$

Ekkor bármely $\lambda \in \mathbb{R}$ esetén

$$\sum_{j \in \mathcal{J}_+} (\bar{x}_j + \lambda\, y_j)\, \mathbf{a}_j = \mathbf{b}$$

adódik. Az y_j számok előjeleitől függően két eset adódhat: 1. az előjelek megegyeznek, 2. az előjelek különböznek.

A 3.10. Tétel bizonyításában definiált $\mathcal{K}_+$ és $\mathcal{K}_-$ halmazok (3.4), segítségével vizsgálhatjuk meg az előző két esetet. Amikor az 1. esetnél vagyunk akkor valamelyik halmaz üres, míg a második esetnél, mindkét halmaz nem üres.

1. Az y_j valós számok előjelei megegyeznek. Tekintsük azt az esetet, hogy $\mathcal{K}_+ \neq \emptyset$ és nyilván ekkor $\mathcal{K}_- = \emptyset$. Ebben az esetben λ' értéke negatív és valamelyik $k \in \mathcal{K}_+$ esetén

$$\lambda' = -\frac{\bar{x}_k}{y_k} \quad \text{és} \quad \tilde{x}_k(\lambda') = \bar{x}_k + \lambda' y_k = \bar{x}_k + \left(-\frac{\bar{x}_k}{y_k}\right) y_k = 0,$$

teljesül. Az $\tilde{\mathbf{x}}(\lambda') \geq \mathbf{0}$ előjelkötést igazoltuk a 3.10. Tétel bizonyításában, a hányados teszt felhasználásával. Előzőleg igazoltuk, hogy legalább egy olyan $j \in \mathcal{J}_+$ index van, amelyik esetén $\tilde{x}_j(\lambda') = 0$ lesz, azaz az $\tilde{\mathbf{x}}(\lambda')$ megoldáshoz tartozó pozitív elemek száma legalább eggyel kisebb, mint az $\bar{\mathbf{x}}$ vektor pozitív elemeinek a száma. Az

$$\tilde{x}_j(\lambda') = \bar{x}_j + \lambda' y_j, \ \text{ha} \ j \in \mathcal{J}_+ \quad \text{és} \quad \tilde{x}_j(\lambda') = 0, \ \text{ha} \ j \in \mathcal{J}_0$$

módon adható meg. Definiálhatjuk az $\bar{\mathbf{y}}$ vektort a következő módon

$$\bar{y}_j = -\lambda' y_j, \ \text{ha} \ j \in \mathcal{J}_+ \quad \text{és} \quad \bar{y}_j = 0, \ \text{ha} \ j \in \mathcal{J}_0.$$

Ekkor az $\tilde{\mathbf{x}}(\lambda')$, $\bar{\mathbf{x}}$, $\bar{\mathbf{y}}$ definíciója miatt a következő összefüggést kapjuk

$$\tilde{\mathbf{x}}(\lambda') = \bar{\mathbf{x}} - \bar{\mathbf{y}}, \quad \text{ahol } \bar{\mathbf{y}} \in \mathcal{H} \tag{3.8}$$

adódik. Másfelől, az indukciós feltétel miatt

$$\tilde{\mathbf{x}}(\lambda') = \sum_{i=1}^{r} \lambda_i \, \mathbf{x}_i + \tilde{\mathbf{y}}, \tag{3.9}$$

ahol az $\mathbf{x}_i$ vektorok, bázis megoldások és $\tilde{\mathbf{y}} \in \mathcal{H}$. Felhasználva a (3.8) és (3.9) kifejezéseket, kapjuk, hogy

$$\bar{\mathbf{x}} - \bar{\mathbf{y}} = \sum_{i=1}^{r} \lambda_i \, \mathbf{x}_i + \tilde{\mathbf{y}}, \quad \text{azaz} \quad \bar{\mathbf{x}} = \sum_{i=1}^{r} \lambda_i \, \mathbf{x}_i + (\tilde{\mathbf{y}} + \bar{\mathbf{y}}),$$

amit igazolni kellett, hiszen $\tilde{\mathbf{y}} + \bar{\mathbf{y}} \in \mathcal{H}$ teljesül.

Hasonlóan lehetne bizonyítani azt az esetet, amikor $\mathcal{K}_- \neq \emptyset$ és nyilván ekkor $\mathcal{K}_+ = \emptyset$ teljesülne, mert ebben az esetben az előjelek azonosak.

2. Az y_j valós számok előjelei nem egyeznek meg, tehát $\mathcal{K}_+ \neq \emptyset$ és $\mathcal{K}_- \neq \emptyset$ teljesül. Ekkor a $\lambda' < 0$, mert $\mathcal{K}_+ \neq \emptyset$ és $\lambda'' > 0$, mert $\mathcal{K}_- \neq \emptyset$. Most egy helyett, két olyan vektort $\tilde{\mathbf{x}}(\lambda') \in \mathcal{P}$ és $\tilde{\mathbf{x}}(\lambda'') \in \mathcal{P}$ tudunk definiálni, amelyeknek legalább eggyel kevesebb pozitív eleme van, mint a $\bar{\mathbf{x}}$ megoldásnak.

Elegendő azt belátni, hogy $\tilde{\mathbf{x}}(\lambda') \in \mathcal{P}$. (Hasonlóan indokolható a $\tilde{\mathbf{x}}(\lambda'') \in \mathcal{P}$ is.)

Az előjelkötöttséget kell igazolni, de a konstrukció miatt, ez nyilván igaz a következő két egyszerűbb esetben: $\tilde{x}_j(\lambda')$, $j \in \mathcal{J}_0$ és $\tilde{x}_j(\lambda')$, $j \in \mathcal{K}_+$. Amikor $j \in \mathcal{K}_-$, akkor $y_j < 0$ és mivel $\lambda' < 0$, ezért $\tilde{x}_j(\lambda') = \bar{x}_j + \lambda' y_j > 0$, hiszen $j \in \mathcal{K}_- \subset \mathcal{J}_+$ miatt az összeg mindkét tagja pozitív.

Kihasználva a $\mathcal{P}$ halmaz konvexitását, megmutatható, hogy

$$\bar{\mathbf{x}} = \frac{\lambda''\, \tilde{x}_j(\lambda') + (-\lambda')\, \tilde{x}_j(\lambda'')}{\lambda'' - \lambda'} = \beta\, \tilde{x}_j(\lambda') + (1-\beta)\, \tilde{x}_j(\lambda''), \qquad (3.10)$$

adódik, ahol $\beta = \frac{\lambda''}{\lambda''-\lambda'}$ és nyilván $0 < \beta < 1$ teljesül. A (3.10) egyenlőség, nyilván igaz a $j \in \mathcal{J}_0$ indexekre. Tekintsük $j \in \mathcal{J}_+$ indexet, ekkor

$$\beta\, (\bar{x}_j + \lambda'\, y_j) + (1-\beta)\, (\bar{x}_j + \lambda''\, y_j) = \bar{x}_j + (\beta\, \lambda' + (1-\beta)\, \lambda'')\, y_j,$$

adódik és $\beta\, \lambda' + (1-\beta)\, \lambda'' = \lambda' \, \frac{\lambda''}{\lambda''-\lambda'} + \lambda'' \, \frac{-\lambda'}{\lambda''-\lambda'} = 0$ miatt igaz az állítás.

Mivel $\tilde{\mathbf{x}}(\lambda')$, $\tilde{\mathbf{x}}(\lambda'') \in \mathcal{P}$ és mindkét vektornak legalább eggyel kevesebb pozitív eleme van, mint a $\bar{\mathbf{x}} \in \mathcal{P}$ vektornak, ezért ezekre alkalmazható az indukciós feltevés, és így

$$\tilde{\mathbf{x}}(\lambda') = \mathbf{x}^1 + \mathbf{y}^1, \qquad \tilde{\mathbf{x}}(\lambda'') = \mathbf{x}^2 + \mathbf{y}^2, \qquad (3.11)$$

előállításokat kapjuk, ahol $\mathbf{x}^1,\ \mathbf{x}^2 \in \mathcal{P}$ vektorok, amelyek megengedett bázis megoldások konvex kombinációi illetve $\mathbf{y}^1,\ \mathbf{y}^2 \in \mathcal{H}$ vektorok.

Felhasználva a (3.10) és (3.11) kifejezéseket, azt kapjuk, hogy

$$\bar{\mathbf{x}} = \beta\, (\mathbf{x}^1 + \mathbf{y}^1) + (1-\beta)\, (\mathbf{x}^2 + \mathbf{y}^2) = (\beta\, \mathbf{x}^1 + (1-\beta)\, \mathbf{x}^2) + (\beta\, \mathbf{y}^1 + (1-\beta)\, \mathbf{y}^2),$$

teljesül, ahol $(\beta\, \mathbf{x}^1 + (1-\beta)\, \mathbf{x}^2) \in \mathcal{P}$ megengedett bázis megoldások konvex kombinációi és $(\beta\, \mathbf{y}^1 + (1-\beta)\, \mathbf{y}^2) \in \mathcal{H}$ vektorok. Így $\bar{\mathbf{x}}$ egy kívánt előállításához jutottunk. ∎

Nem foglalkoztunk eddig a konvex poliéderek korlátosságával. Az előző tétel állításában, egy tetszőleges megoldás előállítását adjuk meg, (3.7). Könnyen belátható, hogy

$$\bar{\mathbf{x}}(\lambda) = \sum_{i=1}^{r} \lambda_i\, \mathbf{x}_i \ + \ \lambda\, \bar{\mathbf{y}} \in \mathcal{P}, \quad \text{bármely} \ \ \lambda \geq 0,$$

esetén. Ha az előállításban szereplő $\bar{\mathbf{y}} \in \mathcal{H}$ vektor, nem a nulla vektor, akkor belátható, hogy az $\bar{\mathbf{x}}(\lambda)$ vektornak legalább egy koordinátája végtelenbe tart, amikor $\lambda \to +\infty$.

Ez azt mutatja, hogy érdemes megfogalmaznunk a konvex poliéderek korlátosságára kritériumot. Mielőtt ezt megtennénk definiáljuk a korlátos, konvex poliédereket.

3.15. Definíció. Legyen $\mathcal{Q} \subset \mathbb{R}^m$. A $\mathcal{Q}$ halmazt *(konvex) politóp*nak nevezzük, ha végesen sok, $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_r \subset \mathbb{R}^m$ vektor konvex burka, azaz

$$\begin{aligned} \mathcal{Q} &= conv(\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_r) \\ &= \{\mathbf{w} \in \mathbb{R}^n : \mathbf{w} = \sum_{i=1}^{r} \lambda_i\, \mathbf{a}_i, \ \sum_{i=1}^{r} \lambda_i = 1, \ \lambda_i \geq 0, \text{ bármely } i \text{ index esetén}\}. \end{aligned}$$

Nyilván, a politóp korlátos, konvex és nemüres halmaz. Másfelől a

$$\mathbf{w} \in \mathcal{Q} \qquad \Longleftrightarrow \qquad A\,\mathbf{x} = \mathbf{w}, \;\; \mathbf{e}^T\mathbf{x} = 1, \;\; \mathbf{x} \geq \mathbf{0}$$

egyenlőtlenségrendszer megoldható.

3.16. Feladat. (Caratheodory tétel, 1911.) Legyen tetszőlegesen, véges sok, $\mathcal{A} = \{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_r\} \subset \mathbb{R}^m$ vektor, és legyen $\mathcal{Q} = conv(\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_r)$. Bármely $\mathbf{x} \in \mathcal{Q}$ vektor esetén, létezik $\mathcal{A}' \subset \mathcal{A}$ halmaz, úgy, hogy $\mathbf{x} \in \mathcal{Q}' = conv(\mathcal{A}')$ és $\mathcal{A}'$ legfeljebb $m+1$ vektort tartalmaz az $\mathcal{A}$ halmazból.

A 3.14. Tétel bizonyításában, az indukciós lépés elfedi, hogy a (3.7) kifejezésben szereplő megengedett bázis megoldások előállítását, amelyeket az adott megengedett megoldásból kiindulva számolhatunk ki.

Ennél egy kicsit egyszerűbb feladattal foglalkozunk, azaz tetszőleges megengedett megoldásból előállítunk egy olyan megengedett bázis megoldást, amelynek a pozitív elemeinek az index halmaza, részhalmaza az eredetileg adott megengedett megoldás pozitív elemei index halmazának.

Algoritmus: megengedett bázis megoldás előállítása adott megengedett megoldásból

---

Bemenő adatok: $m, n \in \mathbb{N}$, $\mathbf{b} \in \mathbb{R}^m$ vektor;
az $\{\mathbf{a}_1, \mathbf{a}_2, ...\mathbf{a}_n\} \subset \mathbb{R}^m$ vektorok és a $\mathcal{J} = \{1, 2, \cdots, n\}$ index halmazuk;
az $\mathbb{R}^m$ tér, $\mathbf{e}_i$ egység vektorainak az index halmaza legyen $\mathcal{I}$;
adott az $\bar{\mathbf{x}}$ megoldás és a $T = [A|\mathbf{b}]$ (rövid) pivot tábla.

Begin
  $\mathcal{J}_+ := \{j \in \mathcal{J} \,|\, \bar{x}_j > 0\}$, $\mathcal{I}_B := \mathcal{I}$ és $\mathcal{J}_{B_+} := \emptyset$;
  if $\{\mathbf{a}_j \,|\, j \in \mathcal{J}_+\}$ lineárisan független **then STOP**
    else
      while ($\exists\, i \in \mathcal{J}_{B_+}$ és $j \in \mathcal{J}_+ \setminus \mathcal{J}_{B_+} : t_{ij} \neq 0$) **do**
        pivotálás a t_{ij} pozíción, $\mathcal{I}_B := \mathcal{I}_B \setminus \{i\}$, és $\mathcal{J}_{B_+} := \mathcal{J}_{B_+} \cup \{j\}$;
      endwhile
      while ($\exists\, i \in \mathcal{J}_{B_+}$ és $j \in \mathcal{J}_+ \setminus \mathcal{J}_{B_+} : t_{ij} \neq 0$) **do**
        $\lambda := \min\left\{ \frac{\bar{x}_i}{-t_{ij}} \;|\; t_{ij} < 0 \right\} = \frac{\bar{x}_k}{-t_{kj}}$;
        if $k \neq j$ **then**
          pivotálás a t_{kj} pozíción és $\mathcal{J}_{B_+} := (\mathcal{J}_{B_+} \setminus \{k\}) \cup \{j\}$;
        endif
        $\bar{\mathbf{x}} := \bar{\mathbf{x}} + \lambda\, \mathbf{t}_j$ és $\mathcal{J}_+ := \{j \in J \,|\, \bar{x}_j > 0\}$;
      endwhile

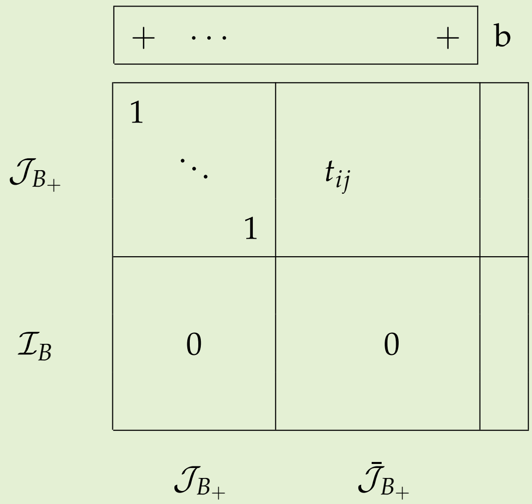
endif

end.

Az előző algoritmus helyes működését és komplexitását a következő állítás bizonyításában tisztázzuk.

3.17. Állítás. *Legyen adott az $A \in \mathbb{R}^{m \times n}$ mátrix és a $\mathbf{b} \in \mathbb{R}^m$ vektor. Az előző algoritmus egy $\bar{\mathbf{x}} \in \mathcal{P} = \{\mathbf{x} \in \mathbb{R}^n : \; A\,\mathbf{x} = \mathbf{b}, \; \mathbf{x} \geq \mathbf{0}\}$ adott megoldásból kiindulva, legfeljebb $\mathcal{O}(n)$ lépésben és legfeljebb $\mathcal{O}(m\;n^2)$ aritmetikai művelettel, előállít egy $\hat{\mathbf{x}} \in \mathcal{P}$ bázis megoldást.*

Bizonyítás. Az első **while** ciklus után a következő pivot táblához jutunk.



A második **while** ciklusban, sorra megvizsgáljuk annak a lehetőségét, hogy valamely $j \in \mathcal{J}_+ \setminus \mathcal{J}_{B_+}$ indexhez tartozó változó értékét 1. vagy nullává tegyük, 2. vagy az x_j változó kerüljön be a bázisba, valamely x_k változó helyére és az x_k változó értéke az új bázisban legyen nulla.

A két eset egyikének a bekövetkezése attól függ, hogy a λ érték kiszámításakor használt hányados nevezőjében a k index megegyezik a j indexszel vagy sem.

Ha $k = j$ akkor az első esetet kapjuk és az új megoldásban $\bar{x}_j = 0$ lesz.

Amennyiben $k \neq j$ teljesül, akkor a t_{kj} elemen pivotálunk és

$$\mathcal{J}_{B_+} := (\mathcal{J}_{B_+} \setminus \{k\}) \cup \{j\}$$

adódik. Megmutatjuk, hogy az új megoldásban $\bar{x}_k = 0$ lesz.

Nyilván $\mid \mathcal{J}_+ \mid \leq n,$ teljesül. Az első **while** ciklusban legfeljebb $m,$ míg a másodikban legfeljebb $\mid \mathcal{J}_+ \mid - m,$ pivotálásra kerül sor. Így összesen legfeljebb n pivotálást hajt végre az algoritmus. Egy pivotálás aritmetikai műveletigénye, az 1.25.

Lemma alapján, legfeljebb $n\,m$. Ezzel az algoritmus komplexitásával kapcsolatos eredményt megkaptuk.

Végezetül lássuk be, hogy az algoritmus által minden iterációban előállított megoldás, megengedett megoldása az adott lineáris egyenlőtlenségrendszernek.

Az első **while** ciklusban, az adott $\bar{\mathbf{x}}$ megengedett megoldás nem változik, hiszen ennek a célja egy olyan induló bázis előállítása, amelyben minél több $j \in \mathcal{J}_+$ indexű $\mathbf{a}_j$ vektor szerepel.

A második **while** ciklusban, az adott $\bar{\mathbf{x}}$ megengedett megoldás, minden iterációban megváltozik, minden iterációban, az új megoldás megengedett megoldás lesz és legalább eggyel csökken a pozitív elemeinek a száma. Igazoljuk először, hogy az új megoldás

$$\mathbf{x}^+ = \bar{\mathbf{x}} + \lambda\,\mathbf{t}_j, \quad \text{ahol } \lambda := \min\left\{ \frac{\bar{x}_i}{-t_{ij}} \mid t_{ij} < 0 \right\} = \frac{\bar{x}_k}{-t_{kj}},$$

megengedett megoldás lesz,

$$A\,\mathbf{x}^+ = A\,(\bar{\mathbf{x}} + \lambda\,\mathbf{t}_j) = A\,\bar{\mathbf{x}} + \lambda\,A\,\mathbf{t}_j = \mathbf{b},$$

hiszen az ortogonalitási tétel (1.48. Tétel) alapján $j \in \mathcal{J}_+ \setminus \mathcal{J}_{B_+}$ nem bázis indexhez tartozó $\mathbf{t}_j$ merőleges az A mátrix $\mathbf{a}^{(i)}$ sor vektoraira. (A kezdeti bázis tábla esetén, az $\mathbb{R}^m$ tér egység vektorai voltak a bázis vektorok.)

Be kell látnunk azt is, hogy $\mathbf{x}^+ \geq \mathbf{0}$ teljesül. Vizsgáljuk meg egy tetszőleges l index esetén az x_l^+ előjelét, figyelembe véve azt, hogy $\lambda > 0$, tehát

$$x_l^+ = \bar{x}_l + \lambda\,t_{lj} \geq 0,$$

bármely $l \in \mathcal{J}$ esetén, amikor $t_{lj} \geq 0$. Tehát elegendő, azokra a $j \in \mathcal{J}$ indexekre megviszgálni, amikor $t_{lj} < 0$, de pontosan ilyen indexek esetén alkalmaztunk hányados tesztet és határoztuk meg a $\lambda > 0$ értéket. Összegezve, $\mathbf{x}^+ \geq \mathbf{0}$ teljesül.

Végül, meg kell mutatnunk, hogy az új megoldásnak legalább eggyel kevesebb pozitív koordinátája van.

Ha $l \in \mathcal{J} \setminus (\mathcal{J}_{B_+} \cup \{j\})$, akkor $t_{lj} = 0$, tehát $x_l^+ = \bar{x}_l$ teljesül, vagyis a koordináták nem változhatnak meg az új megoldásban, kivéve, ha $l \in \mathcal{J}_{B_+} \cup \{j\}$, de ez egyben azt is jelenti, hogy az új megoldásnak nem lehet több pozitív eleme, mint amennyi a régi megoldásnak volt.

Ha $k = j,\ j \in \mathcal{J}_+$, akkor

$$x_j^+ = \bar{x}_j + \lambda\,t_{jj} = \bar{x}_j + \left(-\frac{\bar{x}_j}{t_{jj}}\right)\,t_{jj} = 0,$$

tehát az új megoldásnak legalább eggyel kevesebb pozitív koordinátája lesz.

Ammenyiben $k \neq j,\ j \in \mathcal{J}_+ \setminus \mathcal{J}_{B_+}$ és $k \in \mathcal{J}_{B_+}$, akkor a pivotálás miatt $\mathbf{a}_k$ távozik a bázisból és $\mathbf{a}_j$ belép a bázisba. Továbbá

$$x_k^+ = \bar{x}_k + \lambda\, t_{kj} = \bar{x}_k + \left(-\frac{\bar{x}_k}{t_{kj}}\right) t_{kj} = 0,$$

teljesül. Ebben az esetben is azt kaptuk, hogy eggyel kevesebb pozitív eleme van az új megoldásnak. ∎

Azzal az implicit feltevéssel éltünk, hogy a λ értékét a hányados teszt egyértelműen definiálja. Természetesen ez egy komoly megkötés, hiszen előfordulhatnak olyan esetek, amikor a λ értékét kettő vagy több hányados is meghatározza. Ekkor nyilván az új megoldásban kettő vagy több pozitív elem válik az új megoldásban nullává, sőt az új bázisban is lesz olyan $\mathbf{a}_j$ vektor, amelyikhez tartozó változó értéke nullává válik. Ebben az esetben módosítani kellene az algoritmust oly módon, hogyha $t_{ij} \neq 0,\ \ i \in \mathcal{J}_{B_0},\ j \in \mathcal{J}_+ \setminus \mathcal{J}_{B_+}$ esetén akkor az $\mathbf{a}_i$ bázis vektort cseréljük ki, az $\mathbf{a}_j$ bázison kívüli vektorral. Ezt tegyük meg mindaddig, amíg megfelelő pivot pozíció van.

Ez a módosítás nem befolyásolja az előző eredményünket, csak pontosítja az algoritmus működését.

3.2. Farkas-lemma

Ebben a részben a lineáris egyenlőtlenségrendszerek megoldhatóságával foglalkozunk. Megmutatjuk, hogy a 2.4. és a 2.5. Lemmákat általánosíthatjuk a (3.1) alakú lineáris egyenlőtlenségrendszerek vizsgálatára.

A lemma megfogalmazásakor, két új jelölést vezetünk be: a *nem negatív* valós számokat $\oplus$, míg a *nem pozitív*vakat $\ominus$ jelöli a bázis táblákon. Ezzel a jelöléseket később is használjuk, amikor optimalitási- és nem megengedettségi kritériumokat, un. majdnem leállási táblákat mutatunk be.

3.18. Lemma. *(Előjeles Farkas–Minty tipusú lemma.) Legyenek* $\{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n, \mathbf{b}\} \subset \mathbb{R}^m$ *adott vektorok és* $\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_m\} \subset \mathbb{R}^m$ *az* $\mathbb{R}^m$ *tér egységvektoraiból álló bázisa. Jelölje* $\mathcal{J}$ *az* $\mathbf{a}_j$ *és* $\mathcal{I}$ *az* $\mathbf{e}_i$ *vektorok index halmazát. Az alábbi két (rövid) bázis tábla közül pontosan az egyik fordulhat elő:*

$$
\begin{array}{c|c|c|c|}
 & \bar{\mathcal{J}}'_B & b & \bar{\mathcal{I}}'_B \\
\hline
\mathcal{J}'_B & & \oplus\ \vdots\ \oplus & \\
\hline
\mathcal{I}'_B & 0 & 0\ \vdots\ 0 & \\
\hline
\end{array}
\qquad
\begin{array}{c|c|c|}
 & \bar{\mathcal{J}}''_B & \bar{\mathcal{I}}''_B \\
\hline
\mathcal{J}''_B & & \\
\hline
b & \ominus \ \dots \ \ominus & \\
\hline
\mathcal{I}''_B & 0 & \\
\hline
\end{array}
$$

ahol B' és B'' két különböző bázis az $\{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n, \mathbf{b}, \mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_m\}$ vektorrendszernek és a $\mathcal{J}_{B'}, \mathcal{J}_{B''} \subset \mathcal{J}$ illetve a $\mathcal{I}_{B'}, \mathcal{I}_{B''} \subset \mathcal{I}$ a B' és B'' bázisokhoz tartozó megfelelő index halmazok. A nem bázis vektorok index halmazát rendre $\bar{\mathcal{J}}_{B'}, \bar{\mathcal{J}}_{B''}, \bar{\mathcal{I}}_{B'}$, és $\bar{\mathcal{I}}_{B''}$ jelöli.

Bizonyítás. Az előjeles Farkas–Minty tipusú lemma bizonyítása követi a 2.4. Lemmánál alkalmazott gondolatmenetet, azaz először az 1.48. Tétel felhasználásával megmutatjuk, hogy a két tábla nem fordulhat elő egyszerre, majd pedig megadunk egy algoritmust, amelynek a leállási táblái a lemmában megadott táblák. Az algoritmusról nem lesz az sem nyilvánvaló, hogy véges, ezért a végességét igazolnunk kell.

Mindkét bázis táblából, a kitüntetett szerepű $\mathbf{b}$ vektor oszlopából (nem bázis vektor) illetve sorából (bázis vektor) kiindulva határozzuk meg a $\mathbf{t}'_b$ és $\mathbf{t}''_b$ vektorokat. Az ortogonalitási tétel szerint a két vektor merőleges egymásra. A $\mathbf{t}'_b$ és $\mathbf{t}''_b$ vektorokat az alábbi ábrákkal illusztráljuk

$$
\begin{array}{r|c|c|c|c|c|}
 & \mathcal{J}'_B & \bar{\mathcal{J}}'_B & b & \mathcal{I}'_B & \bar{\mathcal{I}}'_B \\
\hline
\mathbf{t}'_b = & \oplus \dots \oplus & 0 \ \dots \ 0 & -1 & 0 \ \dots \ 0 & 0 \ \dots \ 0 \\
\hline
\end{array}
$$

$$
\begin{array}{r|c|c|c|}
\hline
\mathbf{t}''_b = & \ominus \dots \ominus & 1 & * \ \dots \ * \\
\hline
 & \mathcal{J}''_B & & \mathcal{I}''_B
\end{array}
$$

Négy lényeges dolgot kell észrevennünk a bázis táblák struktúrájának értelmezésekor illetve a $\mathbf{t}'_b$ és $\mathbf{t}''_b$ vektorok meghatározásakor:

- A $\mathbf{t}'_b$ vektor esetén, $t'_{ib} = 0$ teljesül, bármely $i \in \mathcal{I}$ indexre. Ha $i \in \mathcal{I}_{B'}$ akkor az első bázis tábla struktúrája miatt igaz, hogy $t'_{ib} = 0$. Míg, amikor $i \in \bar{\mathcal{I}}_{B'}$ akkor a $\mathbf{t}'_b$ vektor definíciója miatt lesz a $t'_{ib} = 0$.
- A $\mathbf{t}''_b$ vektor esetén, $t''_{bj} \leq 0$ teljesül, bármely $j \in \mathcal{J}$ indexre. Ha $j \in \bar{\mathcal{J}}_{B''}$ akkor a második bázis tábla struktúrája miatt igaz, hogy $t''_{bj} \leq 0$. Ha $j \in \mathcal{J}_{B''}$ akkor a $\mathbf{t}''_b$ vektor definíciója miatt lesz a $t''_{bj} = 0$.

- A $\mathbf{t}'_b$ vektor esetén, az első bázis tábla struktúráját figyelembe véve, $t'_{jb} \geq 0$ teljesül, bármely $j \in \mathcal{J}_{B'}$ indexre. Továbbá, $t'_{jb} = 0$ adódik, amikor $j \in \bar{\mathcal{J}}_{B'}$.

- A $t'_{bb} = -1$ és $t''_{bb} = 1$, a $\mathbf{t}'_b$ és $\mathbf{t}''_b$ vektorok meghatározása miatt igaz.

Ekkor figyelembe véve az előző megállapításokat

$$\mathbf{t}'^{\,T}_b \mathbf{t}''_b = \sum_{j \in \mathcal{J}} t''_{bj}\, t'_{jb} + t''_{bb}\, t'_{bb} + \sum_{i \in \mathcal{I}} t''_{bi}\, t'_{ib} \leq t''_{bb}\, t'_{bb} = -1, \tag{3.12}$$

de ez ellentmond az ortogonalitási tételnek, hiszen

$$0 = \mathbf{t}'^{\,T}_b \mathbf{t}''_b \leq t''_{bb}\, t'_{bb} = -1 < 0 \tag{3.13}$$

kifejezést kapjuk, tehát egyetlen egy $\{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n, \mathbf{b}\} \subset \mathbb{R}^m$ véges vektorrendszer esetén sem fordulhat elő mindkét bázis tábla.

Két tábla közül az egyik fennáll: alkalmazzuk a szokásos eljárásunkat, azaz cseréljünk ki $\mathbf{e}_i$ (bázis) vektort $\mathbf{a}_j$ vektorral ameddig lehet és használjuk fel a 2.4. Lemmát. Ha a lemma második bázis tábláját kapjuk akkor az állításunkban megfogalmazott második bázis tábla speciális esetét kapjuk, hiszen a nem negatív elemek $\ominus$ helyett nulla elemek állnak. Ha viszont a 2.4. Lemma első táblájához jutunk, akkor az alábbi táblát kapjuk

	$\bar{\mathcal{J}}_B$	b	$\bar{\mathcal{I}}_B$
$\mathcal{J}_B$		$*$ $\vdots$ $*$	
$\mathcal{I}_B$	0	0 $\vdots$ 0	

Ha $t_{jb} \geq 0$ teljesül bármelyik $j \in \mathcal{J}_B$ esetén, akkor a lemmában megfogalmazott első bázis táblát kapjuk. Különben létezik $j \in \mathcal{J}_B$ index, amelyre $t_{jb} < 0$ adódik. A második esetben az alábbi algoritmust (**criss-cross algoritmus**) alkalmazzuk:

1. Ha $\mathbf{b}$ vektor nincs a bázisban és a $\mathbf{b}$ oszlopában a $\mathcal{J}_B$ indexekhez tartozó együtthatók között csupa nem negatív szám van, akkor készen vagyunk: az első táblát kaptuk.

2. Ha $\mathbf{b}$ vektor nincs a bázisban és valamely $t_{rb} < 0,\ r \in \mathcal{J}_B$, akkor egy ilyen t_{rb} elemen pivotálunk: a $\mathbf{b}$ vektor ekkor bekerül a bázisba, és az $\mathbf{a}_r$ vektor távozik.

3. Ha a $\mathbf{b}$ vektor a bázisban van és a sorában a $\bar{\mathcal{J}}_B$ oszlopaiban csupa nem pozitív együttható található, akkor készen vagyunk: a második táblát kaptuk.

4. Ha a $\mathbf{b}$ vektor a bázisban van és valamely $t_{bs} > 0$, $s \in \bar{\mathcal{J}}_B$, akkor ezen az elemen pivotálunk: a $\mathbf{b}$ vektor ekkor távozik a bázisból, és az $\mathbf{a}_s$ vektor bekerül. Visszatérünk az **1.** ponthoz.

Azt kell még belátnunk, hogy a fenti algoritmus véges. A végesség igazolásához az előző koncepcionális algoritmust finomítanunk kell, azaz a **2.** lépésben legyen

$$r = \min\{i : i \in \mathcal{J}_B \ \text{ és } \ t_{ib} < 0\}.$$

Hasonlóan, a **4.** lépésben legyen

$$s = \min\{j : j \in \bar{\mathcal{J}}_B \ \text{ és } \ t_{bj} > 0\}.$$

Módosításunk lényege: a *minimál index szabály* alkalmazásával a pivot elem kijelölése a **2.** és **4.** lépésekben egyértelmű.

A minimál indexes criss-cross algoritmus végességét indirekt módon bizonyítjuk.

Tegyük fel, hogy ciklizál az algoritmus egy adott vektorrendszer esetén. Az ilyen ciklizálós példák közül vegyünk egy minimális méretű példát. A példa minimalitása miatt a ciklus során minden változó belép, majd (később) távozik a bázisból. Vizsgáljuk azokat az állapotokat, amikor a legnagyobb indexű $\mathbf{a}_n$ vektor belép a bázisba, illetve amikor távozik onnan. Ekkor a következő un. *majdnem leállási táblá*t kapjuk.

$$\begin{array}{c|c|c|c|}
 & \bar{\mathcal{J}}'_B & b & \bar{\mathcal{I}}'_B \\ \hline
\mathcal{J}'_B & & \begin{matrix}\oplus\\ \vdots \\ \oplus \\ -\end{matrix} & \\ \hline
\mathcal{I}'_B & 0 & \begin{matrix}0\\ \vdots \\ 0\end{matrix} & \\ \hline
\end{array}
\qquad
\begin{array}{c|c|c|}
 & \bar{\mathcal{J}}''_B & \bar{\mathcal{I}}''_B \\ \hline
\mathcal{J}''_B & & \\ \hline
b & \ominus \ \ldots \ \ominus \ + & \\ \hline
\mathcal{I}''_B & 0 & \\ \hline
\end{array}$$

A 1.48. Tétel (Ortogonalitási tétel) felhasználásával megmutatjuk, hogy a két tábla nem fordulhat elő ugyanarra a véges vektorrendszerre. A $\mathbf{t}'$ és $\mathbf{t}''$ oszlop illetve sor vektorokat kiolvasva az első illetve a második bázis táblából, majd a (3.12) és a (3.13) összefüggéseket esetünkre végigszámolva, a $t'_{nb} < 0$ és $t''_{bn} > 0$ elemeket kiemelve, a következő összefüggést kapjuk

$$0 = {\mathbf{t}'_b}^T \mathbf{t}''_b = \sum_{j \in \mathcal{J}} t''_{bj}\, t'_{jb} + t''_{bb}\, t'_{bb} + \sum_{i \in \mathcal{I}} t''_{bi}\, t'_{ib} \le t''_{bn}\, t'_{nb} + t''_{bb}\, t'_{bb} < t''_{bb}\, t'_{bb} = -1,$$

ami ellentmondás. Ezzel beláttuk, hogy a minimál indexes criss-cross algoritmus nem ciklizálhat, azaz véges sok lépésben leáll, mert csupán véges sok lehetséges bázisunk van. A criss-cross algoritmus leállási tábláit, az előjeles Farkas-Minty lemma állításában szereplő első és második bázis táblával adtuk meg az algoritmus **1.** és **3.** lépésében.

Összegezve, tetszőleges $\{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n, \mathbf{b}\} \subset \mathbb{R}^m$ véges vektor rendszer esetén az állításban bemutatott két tábla közül pontosan az egyik fordulhat elő, és azt, amelyik előfordul, a minimál indexes criss-cross algoritmus bemutatott variánsával, véges sok lépésben előállíthatjuk. ■

Az előjeles Farkas-Minty lemma bizonyításában, a Gauss-Jordán eliminációs módszernél, a változók előjelkötésének a teljesítése illetve az alternatív lineáris egyenlőtlenségrendszer egyenlőtlenségeinek a kielégítése érdekében, egy körültekintőbben megfogalmazott eliminációs módszert, pivot algoritmust, kellett definiálnunk. Ezt az algoritmust, a minimál indexes criss-cross módszert, Klafszky Emil és Terlaky Tamás fogalmazták meg és publikálták 1991-ben.

A következő pszeudo-kód formában összegezzük az algoritmust.

Minimál indexes criss-cross algoritmus (első variáns)

Bemenő adatok: $m, n \in \mathbb{N}$, $\mathbf{b} \in \mathbb{R}^m$ vektor;
az $\{\mathbf{a}_1, \mathbf{a}_2, \ldots \mathbf{a}_n\} \subset \mathbb{R}^m$ vektorok és a $\mathcal{J} = \{1, 2, \cdots, n\}$ index halmazuk;
az $\mathbb{R}^m$ tér, $\mathbf{e}_i$ egység vektorainak az index halmaza legyen $\mathcal{I}$;
adott $\mathcal{J}_B \subset \mathcal{J}$, $\mathcal{I}_B \subset \mathcal{I}$ bázis index halmazok és $T = [A|\mathbf{b}]$ (rövid) bázis tábla.

Begin
  while $(\exists i \in \mathcal{J}_B : \; t_{ib} < 0)$ **or** $(\exists j \in \bar{\mathcal{J}}_B : \; t_{bj} > 0)$ **do**
    if $b \in \bar{\mathcal{J}}_B$
      then
        begin
          $r = \min\{i : i \in \mathcal{J}_B \text{ és } t_{ib} < 0\}$,
          pivotálás a t_{rb} elemen,
          $\mathcal{J}_B := (\mathcal{J}_B \setminus \{r\}) \cup \{b\}$
        end
      else
        begin
          $s = \min\{j : j \in \bar{\mathcal{J}}_B \text{ és } t_{bj} > 0\}$,
          pivotálás a t_{bs} elemen,
          $\mathcal{J}_B := (\mathcal{J}_B \setminus \{b\}) \cup \{s\}$
        end
    endif

endwhile
if $b \in \bar{\mathcal{J}}_B$
then stop: $\exists \mathbf{x} \geq \mathbf{0} : \quad A\,\mathbf{x} = \mathbf{b}$
else stop: $\exists \mathbf{y} : \quad \mathbf{y}^T A \leq \mathbf{0},\ \mathbf{y}^T \mathbf{b} = 1$
endif
End.

A 2.5. Lemma az első, klasszikus, un. *alternatíva tétel*, amelyet lineáris egyenletrendszerek megoldhatóságával kapcsolatban bizonyítottunk. Az előző lemma igazolása után, készen állunk arra, hogy a Farkas-lemmát kimondjuk és kidolgozott eszközeinkkel igazoljuk. Farkas Gyula eredményét 1894-ben publikálta először magyar nyelven, majd 1902-ben, az un. *Farkas-tétel*lel együtt, németül is.

3.19. Lemma. *(Farkas lemma, 1894, 1902.) Legyen adott az $A \in \mathbb{R}^{m \times n}$ mátrix és a $\mathbf{b} \in \mathbb{R}^m$ vektor. Az alábbi két lineáris egyenlőtlenségrendszer közül pontosan az egyik oldható meg:*

$$\left.\begin{array}{rcl} A\mathbf{x} & = & \mathbf{b} \\ \mathbf{x} & \geq & \mathbf{0} \end{array}\right\} \quad (\mathcal{E}_1) \qquad\qquad \left.\begin{array}{rcl} \mathbf{y}^T A & \leq & \mathbf{0} \\ \mathbf{y}^T \mathbf{b} & = & 1 \end{array}\right\} \quad (\mathcal{E}_2)$$

Bizonyítás. A kettő egyszerre nem oldható meg, mert különben a következő ellentmondásra jutnánk:

$$0 \geq \mathbf{y}^T\ A\ \mathbf{x} = \mathbf{y}^T\ \mathbf{b} = 1.$$

Az egyik megoldható: alkalmazzuk az előző állítást (3.18. Lemma), a mátrix oszlop vektoraiból és a $\mathbf{b}$ vektorból álló $\{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n, \mathbf{b}\}$ véges vektor rendszerre.

Ha az előjeles Farkas-Minty lemma első bázis táblája fordul elő, akkor

$$\sum_{i \in \mathcal{J}_B} t_{ib}\, \mathbf{a}_i = \mathbf{b},$$

adódik, ahol $t_{ib} \geq 0$. Ebből az $(\mathcal{E}_1)$ lineáris egyenlőtlenségrendszer megoldását a következő módon írhatjuk fel

$$x_i = \begin{cases} \mathrm{t}_{ib}, & \text{ha} \quad \mathrm{i} \in \mathcal{J}_B, \\ 0, & \text{ha} \quad \mathrm{i} \in \bar{\mathcal{J}}_B. \end{cases}$$

Amennyiben a második bázis tábla fordul elő, akkor több információra lesz szükségünk, ezért írjuk ki a teljes pivot táblát

$$
\begin{array}{c|cc|c|c|c|c|}
 & \mathcal{J} & & b & \mathcal{I} & \\
\cline{2-6}
\mathcal{J}_B'' & \begin{matrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{matrix} & & & & 0 \\
\cline{2-6}
b & 0 \; \dots \; 0 & \ominus \; \dots \; \ominus & 1 & & 0 \; \dots \; 0 \\
\cline{2-6}
\mathcal{I}_B'' & 0 & 0 & & & \begin{matrix} 1 & & \\ & \ddots & \\ & & 1 \end{matrix} \\
\cline{2-6}
 & \mathcal{J}_B'' & \bar{\mathcal{J}}_B'' & & \bar{\mathcal{I}}_B'' & \mathcal{I}_B''
\end{array}
$$

amely segítségével az $(\mathcal{E}_2)$ rendszert megoldó $\mathbf{y} \in \mathbb{R}^m$ vektort, a következő összefüggéssel adjuk meg

$$
y_i = \begin{cases} t_{bi}, & \text{ha } \; i \in \bar{\mathcal{I}}_B'', \\ 0, & \text{ha } \; i \in \mathcal{I}_B''. \end{cases}
$$

A kompozíciós tulajdonság felhasználásával, 1.49. Következmény alapján, az $\mathbf{y}$ megoldja az $(\mathcal{E}_2)$ lineáris egyenlőtlenségrendszert. ■

Mielőtt lezárnánk a (3.1) lineáris egyenlőtlenségrendszer megoldhatóságának a vizsgálatát, figyelembe véve korábbi megjegyzésünket, hogy bármelyik lineáris egyenlőtlenségrendszer ilyen alakra hozható, nézzük meg, hogy a Farkas-lemma hogyan alkalmazható, formailag más alakú, lineáris egyenlőtlenségrendszerek megoldhatóságának a vizsgálatára. Tekintsünk egy nagyon egyszerű esetet.

3.20. Következmény. *Legyen adott az $A \in \mathbb{R}^{m\times n}$ mátrix és a $\mathbf{b} \in \mathbb{R}^m$ vektor. Az alábbi két lineáris egyenlőtlenségrendszer közül pontosan az egyik oldható meg:*

$$
\left. A\mathbf{x} \;\geq\; \mathbf{b} \right\} \quad (\mathcal{E}_1) \qquad\qquad \left. \begin{aligned} \mathbf{y}^T A &= \mathbf{0} \\ \mathbf{y}^T \mathbf{b} &= 1 \\ \mathbf{y} &\geq \mathbf{0} \end{aligned} \right\} \quad (\mathcal{E}_2)
$$

Bizonyítás. Alkalmazzuk a Farkas-lemmát, de ehhez az $(\mathcal{E}_1)$ rendszert olyan formára kell hoznunk, amilyen a 3.19. Lemma $(\mathcal{E}_1)$ rendszere. A gondot az okozza, hogy a feltételeink nem egyenlőséges feltételek, ezért bevezetve a $\mathbf{z}$ eltérés vektor változót, az állításunk $(\mathcal{E}_1)$ rendszerét az alábbi formában írhatjuk fel

$$
A\,\mathbf{x} - \mathbf{z} = \mathbf{b}, \quad \mathbf{z} \geq \mathbf{0}.
$$

Ez már majdnem olyan, mint Farkas-lemma $(\mathcal{E}_1)$ rendszere, csupán az a probléma, hogy az $\mathbf{x}$ előjelkötetlen változó. Tudjuk, hogy bármely valós szám felírható két nem negatív valós szám különbségeként, így $\mathbf{x} = \mathbf{x}^+ - \mathbf{x}^-$, ahol $\mathbf{x}^+, \mathbf{x}^- \geq \mathbf{0}$ teljesül. Ezt behelyettesítve az előző kifejezésbe és a szükséges átalakításokat elvégezve kapjuk, hogy

$$A\,(\mathbf{x}^+ - \mathbf{x}^-) - \mathbf{z} = \mathbf{b}, \;\; \mathbf{x}^+, \mathbf{x}^-, \mathbf{z} \geq \mathbf{0}$$

és az új változók segítségével az $(\mathcal{E}_1)$ lineáris egyenlőtlenségrendszert, az alábbi formára transzformáltuk

$$A\,\mathbf{x}^+ - A\,\mathbf{x}^- - \mathbf{z} = \mathbf{b}, \;\; \mathbf{x}^+, \mathbf{x}^-, \mathbf{z} \geq \mathbf{0},$$

amely pontosan olyan, mint (3.1), a Farkas-lemma első lineáris egyenlőtlenségrendszere. Bevezetve az $\mathbf{y} \in \mathbb{R}^m$ változókat, az $(\mathcal{E}_1)$ alternatív párját, a Farkas-lemma felhasználásával a következő formában írhatjuk le

$$\left.\begin{array}{rcl} \mathbf{y}^T A & \leq & \mathbf{0} \\ \mathbf{y}^T(-A) & \leq & \mathbf{0} \\ \mathbf{y}^T(-I) & \leq & \mathbf{0} \\ \mathbf{y}^T\mathbf{b} & = & 1. \end{array}\right\}$$

Az első és második lineáris egyenlőtlenségrendszerből következik az $\mathbf{y}^T A = \mathbf{0}$ lineáris egyenletrendszer, míg a harmadikból, a változóra vonatkozó előjelkötés, azaz $\mathbf{y} \geq \mathbf{0}$. Összegezve, az $(\mathcal{E}_2)$ lineáris egyenlőtlenségrendszert kapjuk meg. ∎

Számos szerző, fontosnak tartja megfogalmazni a lineáris egyenlőtlenségrendszerek megoldhatóságára vonatkozó általános alakú alternatíva tételt. Az állítást úgy értjük, hogy a mátrixok és vektorok mérete olyan, hogy az előírt mátrix szorzás és összeadás műveleteket elvégezhetjük.

3.21. Feladat. Az alábbi két rendszer közül pontosan az egyik oldható meg:

$$\left.\begin{array}{c} P\mathbf{x}_0 + A\mathbf{x}_1 = \mathbf{b}_0 \\ Q\mathbf{x}_0 + B\mathbf{x}_1 \leq \mathbf{b}_1 \\ \mathbf{x}_1 \geq 0 \end{array}\right\} \; (\mathcal{E}_1) \qquad \left.\begin{array}{c} \mathbf{y}_0^T P - \mathbf{y}_1^T Q = 0 \\ \mathbf{y}_0^T A - \mathbf{y}_1^T B \leq 0 \\ \mathbf{y}_0^T \mathbf{b}_0 - \mathbf{y}_1^T \mathbf{b}_1 = 1 \\ \mathbf{y}_1 \geq 0 \end{array}\right\} \; (\mathcal{E}_2)$$

Térjünk vissza a 3.18. Lemmához és tegyük fel, hogy az $A\,\mathbf{x} = \mathbf{b}$ lineáris egyenletrendszer megoldható, ekkor a 2.4. Lemma első bázis táblája fordul elő. Kérdés:

milyen előjelstruktúra jellemzi azt a táblát, amelyik kimutatja, hogy a lineáris egyenletrendszernek nem lehet olyan megoldása, amelyben minden változó előjelkötött, azaz nem negatív.

3.22. Feladat. Legyenek $\{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n, \mathbf{b}\} \subset \mathbb{R}^m$ adott vektorok és az $\mathbb{R}^m$ tér egység vektoraiból álló bázisa, $\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_m\} \subset \mathbb{R}^m$. Jelölje $\mathcal{J}$ az $\mathbf{a}_j$ és $\mathcal{I}$ az $\mathbf{e}_i$ vektorok index halmazát. Tegyük fel továbbá azt, hogy a

$$rang(\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_n) = rang(\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_n, \mathbf{b}),$$

ekkor az alábbi két bázis táblából pontosan egyik fordulhat elő

$$\begin{array}{c|c|c|c|}
 & \mathcal{J}'_B & b & \mathcal{I}'_B \\
\hline
\mathcal{J}'_B & & \begin{matrix}\oplus \\ \vdots \\ \oplus\end{matrix} & \\
\hline
\mathcal{I}'_B & 0 & \begin{matrix}0 \\ \vdots \\ 0\end{matrix} & \\
\hline
\end{array}
\qquad
\begin{array}{c|c|c|c|}
 & \mathcal{J}''_B & b & \mathcal{I}''_B \\
\hline
\mathcal{J}''_B & \oplus \; \ldots \; \oplus & - & \\
\hline
\mathcal{I}''_B & 0 & \begin{matrix}0 \\ \vdots \\ 0\end{matrix} & \\
\hline
\end{array}$$

A második bázis táblán látható előjel struktúrát *primál nem megengedettségi kritérium*nak nevezzük és fontos szerepet fog játszani a lineáris programozási feladatok tárgyalása során. Könnyen belátható, hogy bármely $\mathbf{x} \geq \mathbf{0}$ vektorral megszorzva a táblán jelzett i. sor $\mathbf{t}^{(i)}$ sor vektorát, akkor

$$(\mathbf{t}^{(i)})^T \mathbf{x} = \sum_{j \in \mathcal{J}} t_{ij}\, x_j \geq 0$$

adódik, ezzel szemben a $\bar{b}_i$ (jelenlegi bázisban felírt) értéke negatív. Figyelembe véve, hogy a bázisok egymásba transzformálhatók (1.38. Lemma), a különböző bázisokban reprezentált lineáris egyenletrendszerek, ugyanazt a megoldás halmazt (affin alteret) írják le. A jelenlegi bázis reprezentáció pedig egyértelműen azt mutatja, hogy az affin altér nem metszhet bele a nem negatív ortánsba, azaz a (3.1) lineáris egyenlőtlenségrendszernek, ebben az esetben, nincsen megoldása.

A következő bázis tábla pár, az alternatív rendszer megoldhatóságát elemzi.

3.23. Feladat. Legyenek $\{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n, \mathbf{b}\} \subset \mathbb{R}^m$ adott vektorok és az $\mathbb{R}^m$ tér egység vektoraiból álló bázisa, $\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_m\} \subset \mathbb{R}^m$. Jelölje $\mathcal{J}$ az $\mathbf{a}_j$ és $\mathcal{I}$ az $\mathbf{e}_i$ vektorok index halmazát. Tegyük fel továbbá azt, hogy a

$$rang(\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_n) = rang(\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_n, \mathbf{b}),$$

ekkor az alábbi két bázis táblából pontosan egyik fordulhat elő

$$
\begin{array}{c|c|c|}
 & \bar{\mathcal{J}}'_B & \bar{\mathcal{I}}'_B \\
\hline
\mathcal{J}'_B & \begin{matrix}\ominus \\ \vdots \\ \ominus\end{matrix} & \\
\hline
b & + & \\
\hline
\mathcal{I}'_B & 0 & \\
\hline
\end{array}
\qquad\qquad
\begin{array}{c|c|c|}
 & \bar{\mathcal{J}}''_B & \bar{\mathcal{I}}''_B \\
\hline
\mathcal{J}''_B & & \\
\hline
b & \ominus \; \ldots \; \ominus & \\
\hline
\mathcal{I}''_B & 0 & \\
\hline
\end{array}
$$

Az előző feladat első bázis tábláján látható előjel struktúrát *duál nem megengedettségi kritérium*nak szokás nevezni. Ez is fontos szerepet fog játszani lineáris programozási feladatok során.

A lineáris alternatíva tételek néhány speciális esete nagyobb érdeklődésre tett szert, mint más variánsok. Az első, azért érdekes, mert nemlineáris programozási algoritmusok elemzése során fordul elő.

3.24. Feladat. (Goldman tétel) Legyen az $A \in \mathbb{R}^{m\times n}$ mátrix és a $\mathbf{b} \in \mathbb{R}^m$, $\mathbf{c} \in \mathbb{R}^n$ vektorok. Ekkor az alábbi két lineáris egyenlőtlenségrendszer közül pontosan az egyiknek van megoldása:

(a) $A\mathbf{x} = \mathbf{b},\ \mathbf{c}^T\mathbf{x} = -1,\ \mathbf{x} \geq \mathbf{0},$

(b) $\mathbf{y}^T A \leq \mathbf{c}.$

A következő lineáris alternatíva tétel azért érdekes, mert az első lineáris egyenlőtlenségrendszere olyan alakú, amilyen megengedett megoldás halmazon a Karmarkar-féle projektív skálázású belsőpontos algoritmus (1984), lineáris programozási feladatot polinomiális lépésszámban megtud oldani.

Másfelől, azért is érdekes az első lineáris egyenlőtlenségrendszer, mert a változiról szigorú előjelkötöttséget, azaz pozitivitást követel meg.

3.25. Feladat. Legyen $A \in \mathbb{R}^{m\times n}$ tetszőleges mátrix, $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{y} \in \mathbb{R}^m$ vektorok és $y_0 \in \mathbb{R}$. Igazolja, hogy a következő két lineáris egyenlőtlenségrendszer közül pontosan az egyik oldható meg

(a) $A\,\mathbf{x} = \mathbf{0},\ \mathbf{e}^T\mathbf{x} = 1,\ \mathbf{x} > \mathbf{0},$

(b) $A^T\,\mathbf{y} + \mathbf{e}\,y_0 \leq \mathbf{0},\ y_0 \geq 0,$

ahol $\mathbf{e} = (1, 1, ..., 1) \in \mathbb{R}^n$.

A lineáris egyenlőtlenségrendszerek megoldhatóságával kapcsolatos vizsgálatok szempontjából érdekes a következő állítás, mivel azt mutatja, hogy

- a tételek alternatíva forma helyet ekvivalens állításokként is felírhatók, és
- az a tény, hogy vektorok közötti egyenlőtlenségek, időnként úgy is teljesülhetnek, hogy egy-egy koordináta esetén szigorú egyenlőtlenséggel teljesülnek, az alternatív rendszer esetén, szigorúan előjelkötött változót eredményez.

3.26. Feladat. Legyen $B \in \mathbb{R}^{m\times n}$, $C \in \mathbb{R}^{k\times n}$, $D \in \mathbb{R}^{l\times n}$, $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{y} \in \mathbb{R}^m$, $\mathbf{z} \in \mathbb{R}^k$ és $\mathbf{u} \in \mathbb{R}^l$. Bizonyítsa be, hogy a következő két állítás ekvivalens:

(a) $\exists \mathbf{x} : \ B\mathbf{x} \le \mathbf{0}, \ B\mathbf{x} \ne \mathbf{0}, \ C\mathbf{x} \le \mathbf{0}, \ D\mathbf{x} = \mathbf{0},$

(b) $\nexists (\mathbf{y}, \mathbf{z}, \mathbf{u}) : \ B^T\mathbf{y} + C^T\mathbf{z} + D^T\mathbf{u} = \mathbf{0}, \ \mathbf{y} > \mathbf{0}, \ \mathbf{z} \ge \mathbf{0}.$

3.3. Az MBU-szimplex algoritmus

A (3.1) lineáris egyenlőtlenségrendszer megoldhatóságát tárgyaltuk az előző részben. A 3.19. Farkas-lemmával, mint alternatíva tétellel, jellemeztük a (3.1) rendszer megoldhatóságát. Amennyiben a lineáris egyenlőtlenségrendszernek nincsen megoldása, az alternatív rendszer megoldását állítottuk elő. A 3.18. Lemma bizonyításában megfogalmaztuk a criss-cross algoritmust, amellyel megoldhatók a lineáris egyenletrendszerek. A criss-cross módszer végességét igazoltuk.

A (3.1) lineáris egyenlőtlenségrendszert, *(lineáris) megengedettségi feladat*nak nevezzük. Az általánosság korlátozása nélkül feltehetjük, hogy $rang(A) = m$, ugyanis a lineáris feltételrendszer megoldhatóságát ellenőrizhetjük, illetve a redundáns lineáris egyenleteket elhagyhatjuk a Gauss-Jordán elimináció segítségével.

A megengedettségi feladatok legismertebb megoldatlan kérdése az, hogy létezik-e erősen polinomiális futás idejű pivot algoritmus a (3.1) probléma megoldására.

A következő részben bemutatunk egy algoritmust, amelynek nem csak a végességét tudjuk igazolni, hanem bizonyos nem degeneráltsági feltételek mellett, az algoritmus lépés számára is adhatunk egy felsőkorlátot. A felsőkorlátunk – egészegyütthatós lineáris egyenlőtlenségrendszerek esetén – függni fog a feladat adatainak a leírásához szükséges tárigénytől (2.13. Definíció), az L értékétől.

Ebben a részben megmutatjuk, hogy az *Anstreicher-Terlaky monoton szimplex algoritmus*nak (vagy MBU-szimplex algoritmusnak) megfogalmazható egy olyan variánsa az (3.1) feladatra, amelynek a lépésszámát az $m\Delta$ adja meg, ahol Δ a feladat adataiból kiszámítható konstans. Annak ellenére, hogy a Δ konstans, a feladat leírásához szükséges számítógépes tárigénynek – az adatok bithosszának – egy polinomjával nem mindig korlátozható, és egy speciális *nem degeneráltsági feltevés*sel is élnünk kell az elemzésünk során, eredményünk mégis érdekes, hiszen hasonló nem ismert a szakirodalomból.

A (3.1) lineáris egyenlőtlenségrendszer és a lineáris programozási feladat között, amelyet később részletesen tárgyalunk majd, a lényeges különbség, hogy a lineáris programozási feladat esetén a lineáris megkötések mellett, lineáris célfüggvényt szeretnénk optimalizálni (minimalizálni vagy maximalizálni). Mivel a lineáris programozási feladat esetén, a megengedett megoldások halmazát konvex poliéder írja le, ezért szokás a (3.1) feladatot, lineáris egyenlőtlenségrendszert, megengedettségi feladatnak nevezni. Amennyiben, a lineáris programozási feladat esetén a lineáris célfüggvény minden együtthatója nulla, úgy a megengedettségi feladatot, mint speciális esetet kapjuk vissza.

Visszatérve a monoton szimplex algoritmus variánsának a lépésszám elemzéséhez, azt is kiemelhetjük, hogy a lineáris programozási feladatra, a *Dantzig-féle szimplex algoritmus* felfedezése óta sem ismert olyan pivot algoritmus, amelynek a lépésszám becslését (lépésszámára felsőkorlátot) az algoritmus elemzéséből nyernénk. A lineáris programozási feladat megoldására definiált pivot algoritmusok nagy többségéről megmutatták, hogy a *Klee-Minty feladat*on, vagy valamely azzal nagyon hasonló struktúrájú exponenciális ellenpéldán, egy adott bázisból indulva exponenciálisan sok bázis csere segítségével állítják elő az optimális megoldást. A pivot algoritmusok többségénél, a végesség bizonyításán kívül, ez az egyetlen elméleti jellegű támpont az algoritmus – elméleti – hatékonyságára, általános lineáris programozási feladat esetén.

Lineáris programozás esetén, a célfüggvénynek jelentős szerepe van abban, hogy a Klee-Minty feladaton a szimplex módszer az origóból indulva, a Klee-Minty kocka minden csúcsát meglátogatja. Ennek ellenére, nem kétséges az, hogy a 3.18. Lemmában definiált criss-cross módszer illetve ebben a részben bemutatásra kerülő monoton szimplex algoritmus variáns, a (3.1) lineáris egyenlőtlenségrendszer megoldása során exponenciális lépésszámú lehet. (Természetesen, matematikai szempontból az állítás akkor lenne – teljesen – korrekt, ha az említett algoritmusokra, a Klee-Minty feladathoz hasonlóan, megadható lenne egy jól leírható megengedettségi feladat, tetszőleges dimenzióban, amelyen az említett algoritmusok valamely adott bázisból indulva exponenciálisan sok iterációval jutnának el egy megenegedett bázisba. Annak ellenére, hogy ezekre az algoritmusokra ilyen példát nem ismerünk, a szakmai közmegegyezés szerint, ezek az algoritmusok nem lehetnek polinomiális komplexitásúak.)

A bemutatásra kerülő monoton szimplex algoritmus variáns, rendelkezik azzal a tulajdonsággal, hogy ha valamelyik változó a számítási eljárás során megengedetté vált, akkor ezt a megengedettséget mindvégig megőrzi, tehát a feladat megoldása – a változók megengedettségének az elérése – monoton módon történik.

Az MBU-szimplex algoritmusról – egy nem degeneráltsági feltétel mellett – kimutatjuk, hogy $m\Delta$ iterációra van szükségünk a feladat megoldásához. Tekintettel arra, hogy a nem degeneráltsági feltétel teljesülését nem lehet apriori bizonyítani, így minimál-index szabály segítségével bizonyítjuk hogy az MBU-szimplex algoritmus végességét.

Az MBU-szimplex algoritmus geometriai tulajdonságai, hasonlóak a criss-cross módszeréhez, abban az értelemben, hogy az algoritmus a megengedettségi feladat feltételei által alkotott metszésrendszer szomszédos metszéspontjain – nem megengedett bázisokon- - halad keresztül. Eltér azonban a criss-cross algoritmustól abban, hogy ha valamely iteráció során az aktuális bázis megoldás egy adott feltétel által meghatározott hipersík megengedett oldalára került, akkor a további iterációkban előállított bázis megoldások sem fognak ezen feltétel nem megengedett oldalára kerülni.

Az MBU-szimplex algoritmus általános bevezetőjét talán célszerű azzal befejezni, hogy az általános, nem nulla alsókorlátú, (maximális) hálózati folyam feladathoz, a most ismertetésre kerülő MBU-szimplex algoritmus egy egyszerű módosításával előálló új variánsával (Illés T. és Molnár-Szipai R., 2012), erősen polinom lépésben lehet, a hálózati folyam feladat egy tetszőleges – se nem primál, se nem duál megengedett – bázis megoldásból, primál megengedett bázis megoldást előállítani.

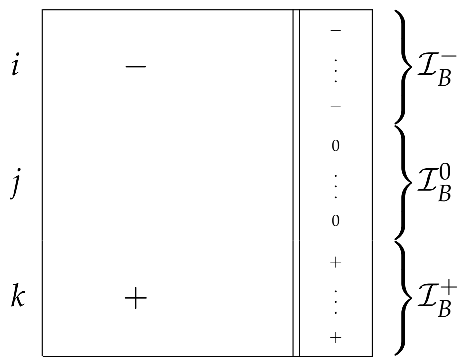
Egy rögzített B bázis esetén, az MBU-szimplex algoritmus tárgyalásakor a következő partíciókat fogjuk használni:

$$\mathcal{I}_B = \mathcal{I}_B^+ \cup \mathcal{I}_B^0 \cup \mathcal{I}_B^-,$$

ahol

$$\mathcal{I}_B^+ = \{i \in \mathcal{I}_B : \bar{x}_i > 0\}\,,\; \mathcal{I}_B^0 = \{i \in \mathcal{I}_B : \bar{x}_i = 0\} \quad \text{és} \quad \mathcal{I}_B^- = \{i \in \mathcal{I}_B : \bar{x}_i < 0\}\,.$$

Időnként, ha csupán a megengedettségét szeretnénk hangsúlyozni bizonyos bázisváltozóknak, akkor az $\mathcal{I}_B^{\oplus} = \mathcal{I}_B^+ \cup \mathcal{I}_B^0$ jelölést használjuk. A partíciónak megfelelő bázistábla a következő



3.1. ábra. A bázis partícionálása.

A legtöbb szakirodalomból ismert pivot módszer primál megengedett sorban csak pozitív elemen végez báziscserét (pl. primál szimplex), illetve primál nem megengedett sor esetén negatív elemen végez báziscserét (pl. duál szimplex) esetleg a kétféle stratégiát ötvözi (pl. criss-cross módszer). Ezt felismerve Fukuda és Terlaky

(1997), bevezették az úgynevezett elfogadható pivot fogalmát. Az MBU-szimplex algoritmus ennél általánosabb feltétellel definiált pivot választást is használni fog. A pozitív értékű változó sorában pozitív pivot elemen végzett pivotálás, felfogható az elfogadható báziscserék duál oldali pivotjának megengedettségi feladatokra vonatkozó megfelelőjeként. A degeneráltság kezelésekor a degenerált sorokban pozitív, illetve negatív elemen is végezhetünk báziscserét.

3.27. Definíció. Egy adott B bázis és T pivot tábla esetén a t_{ij} elemet *általánosított elfogadható* pivot elemnek nevezzük, ha

1. $i \in \mathcal{I}_B^-$ és $t_{ij} < 0$, vagy
2. $i \in \mathcal{I}_B^+$ és $t_{ij} > 0$, vagy
3. $i \in \mathcal{I}_B^0$ és $t_{ij} \neq 0$.

Algoritmusunkban, és a későbbiekben bevezetett részfeladatok megoldásakor általánosított elfogadható báziscseréket használunk. Az algoritmus megfogalmazásához szükségünk lesz a degeneráltság fogalmának a finomítására. Vezessük be $s \in \mathcal{I}_N$ esetén a

$$\mathcal{K}_s = \left\{ i \in \mathcal{I}_B^0 : t_{is} > 0 \right\}$$

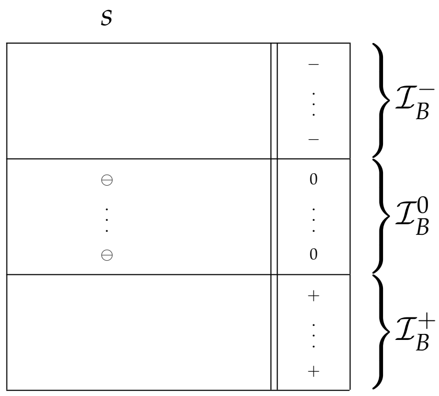
jelölést.

3.28. Definíció. Egy B bázist *nem degenerált*nak nevezünk, ha $\bar{\mathbf{x}}_B$ egyik komponense sem nulla. A B bázis *degenerált*, ha az $\bar{\mathbf{x}}_B$ bázismegoldásnak van nulla komponense.

A degeneráltság jelensége az aktuális bázistól is függ, oly módon, hogy az adott bázismegoldás előállítható a dimenziójánál kevesebb bázisbeli elem kombinációjaként. A későbbiekben a degeneráltság két fajtáját fogjuk megkülönböztetni.

3.29. Definíció. A B degenerált bázist *lokálisan gyengén degenerált*nak nevezzük az $s \in \mathcal{I}_N$ indexre vonatkozóan, ha $\mathcal{K}_s = \emptyset$, illetve *lokálisan erősen degenerált*nak nevezzük az s indexre vonatkozóan, ha $\mathcal{K}_s \neq \emptyset$.

Tegyük fel hogy egy adott B bázis esetén eldöntöttük, hogy a bázistábla s. oszlopában kívánunk nem degenerált sorban általánosított elfogadható báziscserét végezni olymódon, hogy a degenerált változók ne váljanak nem megengedetté. Figyeljük meg, hogy ez pontosan akkor lehetséges, ha $\mathcal{K}_s = \emptyset$. Egy ilyen, az s indexre nézve lokálisan gyengén degenerált táblát mutat a 3.2. ábra.



3.2. ábra. Lokálisan gyengén degenerált pivot tábla.

3.30. Definíció. Legyen adott egy B bázis és egy $r \in \mathcal{I}_B^-$ index. A megfelelő bázis tábla egy t_{ij} elemén végzett pivotálást x_r *növelő*nek nevezünk, ha

1. a t_{ij} elem általánosított elfogadható pivot elem,
2. $\mathcal{I}_B^\oplus \subseteq \mathcal{I}_{B'}^\oplus$ és
3. $\hat{x}_r > \bar{x}_r$ teljesül,

ahol $\bar{\mathbf{x}}$ az aktuális megoldást, B' az új bázist és $\hat{\mathbf{x}}$ az új bázismegoldást jelöli.

Az x_r növelő pivotálásra mutatunk egy egyszerű példát.

3.31. Példa. Tekintsük a következő rövid pivot táblát.

$$\begin{array}{c|cc||c|}
 & x_4 & x_5 & \\
\hline
x_1 & 0 & -1 & 0 \\
x_2 & \mathbf{1} & -1 & 1 \\
x_3 & -1 & 0 & -2 \\
\hline
\end{array}$$

A tábla egyetlen nem megengedett változójának a sorában egy negatív elem található, így a vezérváltozó az x_3, és az x_4 oszlopában végzünk pivotálást. A vezérváltozó sorában a pivotálás nem lenne növelő, mert az x_2 változó nem megengedetté válna, ellentmondva a definíció 2. feltételének. Mivel a tábla gyengén degenerált $(\mathcal{K}_4 = \emptyset)$, így van x_3 növelő báziscsere, azaz az x_2 távozik és az x_4 belép a bázisba.

	x_2	x_5	
x_1	0	−1	0
x_4	1	−1	1
x_3	1	−1	−1

Célunk x_r növelő pivotálásokat végrehajtó algoritmust megfogalmazni. Ismereteink szerint a szakirodalomból egyetlen x_r növelő pivotálásokat végző algoritmus ismert, Anstreicher és Terlaky (1994) általános, primál illetve duál megengedett megoldásból induló lineáris programozási feladatokra megfogalmazott primál illetve duál oldali monoton-szimplex algoritmusa.

Sajnos léteznek olyan bázistáblák, melyen nincs x_r növelő pivotálás, ahogyan azt a 3.3. ábra példája is mutatja. A probléma egy megengedett megoldása az $x_1 = x_2 = 0, x_3 = x_4 = 1$, még sincs az egyetlen negatív értékű sorhoz növelő pivotálás.

	x_3	x_4	
x_1	−1	0	−1
x_2	1	−1	0

3.3. ábra. Lokálisan erősen degenerált tábla, melyen nincsen x_r növelő pivotálás, ahol $r = 1$.

Ebben az esetben, az MBU-szimplex algoritmus, *degenerált pivot lépés*re kényszerül majd, az x_2 és x_3 változók kicserélésével.

Algoritmusunk a monotonitás elérése érdekében a következő szemléletes képet követi. Kiválaszt egy, az adott megoldásban nem megengedett változót, melyet az elkövetkező iterációk során megengedetté kíván tenni. Mindaddig, míg ezen változó megengedetté nem válik, ezt a változót *vezér változónak* fogjuk nevezni. Az x_r vezérváltozó megengedettségét monoton módon javítjuk x_r növelő báziscserék segítségével. Látni fogjuk, ha az algoritmus csupán nem degenerált, vagy lokálisan gyengén degenerált bázistáblákon halad, akkor ez mindig megtehető.

Az x_r növelő báziscsere megtalálásának érdekében a báziscserére kiválasztott oszlopban két hányadostesztet kell végezni, melyek értékének viszonyából állapítható meg hogy végezhető-e báziscsere a vezérváltozó sorában, vagy általánosabb x_r növelő báziscserére van-e szükség.

Az algoritmus leírásában a két hányadosteszt értékét Θ_1-gyel és Θ_2-vel jelöltük. A definíciójukból könnyen látható, hogy $\Theta_1 > 0$ és $\Theta_2 \geq 0$, továbbá ha a megfelelő bázis lokálisan nem erősen degenerált, akkor $\Theta_2 > 0$. Az algoritmus belső ciklusának célja a vezérváltozó megengedettségének előállítása.

Megmutatjuk, hogy az algoritmus kizárólag x_r növelő báziscseréket végez, ha az adott B bázis a kiválasztott nem bázis változóra nézve nem erősen degenerált. A következő állításban azt az esetet vizsgáljuk, amikor éppen a vezérváltozó távozik a bázisból.

MBU-szimplex algoritmus megengedettségi feladatokra

Bemenő adatok: $A \in \mathbb{R}^{m \times n}, \mathbf{b} \in \mathbb{R}^m$,
$B \in \mathbb{R}^{m \times m}$ az A reguláris részmátrixa,
$T := B^{-1}A, \bar{\mathbf{b}} := B^{-1}\mathbf{b}, \mathcal{I}_B^- := \{i \in \mathcal{J} \mid \bar{b}_i < 0\}$.

Begin
while $(\mathcal{I}_B^- \neq \emptyset)$ **do**
Legyen $r \in \mathcal{I}_B^-$ tetszőleges (vezér változó), $rDone := false$.
while $(rDone = false)$ **do**
$\mathcal{J}_r^- := \{j \in \mathcal{I}_N \mid t_{rj} < 0\}$.
if $(\mathcal{J}_r^- = \emptyset)$ **then**
nincs megengedett megoldás, **return**
endif
Legyen $s \in \mathcal{J}_r^-$ tetszőleges, $\mathcal{K}_s := \{i \in \mathcal{I}_B^0 \mid t_{is} > 0\}$.
if $(\mathcal{K}_s \neq \emptyset)$ **then**
$(T, l) = DegEljárás(T, \mathcal{I}_B^0, r)$.
if $(l \in \mathcal{I}_N)$ **then**
$s := l$.
else
nincs megengedett megoldás, **return**
endif
endif
$\Theta_1 := \frac{\bar{b}_r}{t_{rs}}$, és $\Theta_2 := \min\left\{\frac{\bar{b}_k}{t_{ks}} \mid k \in \mathcal{I}_B^{\oplus}, t_{ks} > 0\right\}$.
if $(\Theta_1 \leq \Theta_2)$ **then**
pivotálás a t_{rs} elemen, $rDone = true$.
else
$q := \arg_k \min\left\{\frac{\bar{b}_k}{t_{ks}} \mid k \in \mathcal{I}_B^+, t_{ks} > 0\right\}$, pivotálás a t_{qs} elemen.
endif
endwhile
$\mathcal{I}_B^- := \{i \mid \bar{b}_i < 0\}$.
endwhile
$\bar{\mathbf{x}}$ megengedett megoldás.
End.

3.32. Állítás. *Az MBU szimplex algoritmus egy adott lépésében az aktuális B bázis esetén legyen $r \in \mathcal{I}_B^-$ és $q \in \mathcal{I}_B^\oplus, t_{qs} > 0$. Tegyük fel, hogy*

$$\frac{\bar{b}_q}{t_{qs}} = \Theta_2 \geq \Theta_1 = \frac{\bar{b}_r}{t_{rs}} > 0,$$

ahol $\bar{b}_r < 0$, $t_{rs} < 0$, $\bar{b}_q \geq 0$ és $t_{qs} > 0$. Ekkor az algoritmus a t_{rs} elemen végez báziscserét. Jelölje a báziscsere utáni új bázist B'. Ekkor

$$\mathcal{I}_B^\oplus \cup \{s\} \subseteq \mathcal{I}_{B'}^\oplus$$

teljesül, és így

$$\left|\mathcal{I}_{B'}^\oplus\right| > \left|\mathcal{I}_B^\oplus\right|.$$

Bizonyítás. A t_{rs} elemen végzett báziscsere során az x_r változó távozik a bázisból, és így új értéke nulla, míg az x_s változó belép a bázisba. Jelölje az új bázishoz tartozó megoldást –és így az új bázistábla jobboldalát– $\mathbf{b}^+$. Bizonyításunk esetszétválasztáson alapszik.

a) Az s index esetén a megfelelő jobboldal $b_s^+ = \frac{\bar{b}_r}{t_{rs}} = \Theta_1 > 0$ vagyis a vezérváltozó helyére belépő változó megengedett lesz.

b) Az $i \in \mathcal{I}_B^+, i \neq s$ esetén az új jobboldalt a $b_i^+ = \bar{b}_i - \frac{t_{is}\bar{b}_r}{t_{rs}}$ képlettel számolható ki. Ha $t_{is} \leq 0$, akkor felhasználva, hogy $\bar{b}_i > 0$, $\bar{b}_r < 0$ és $t_{rs} < 0$ adódik, hogy $b_i^+ > 0$ mivel egy nemnegatív számot adunk a már amúgy is pozitív $\bar{b}_i$ jobboldalhoz. Egyébként, ha $t_{is} > 0$ akkor a $b_i^+ = t_{is}\left(\frac{\bar{b}_i}{t_{is}} - \frac{\bar{b}_r}{t_{rs}}\right)$ átalakítás alapján, mivel a vezérváltozó sorában történik a báziscsere, ezért a $\frac{\bar{b}_i}{t_{is}} \geq \Theta_2 \geq \Theta_1 = \frac{\bar{b}_r}{t_{rs}} > 0$ feltétel alapján $b_i^+ > 0$.

c) Legyen $i \in \mathcal{I}_B^0$. Az új jobboldal értéke $b_i^+ = \bar{b}_i - \frac{t_{is}\bar{b}_r}{t_{rs}}$. Mivel $\frac{\bar{b}_q}{t_{qs}} = \Theta_2 > 0$, ezért a $q \notin \mathcal{I}_B^0$, vagyis a $\mathcal{K}_s = \emptyset$ és így a $t_{is} \leq 0$. Figyelembe véve a $\bar{b}_i = 0$ feltételt is, a $b_i^+ = -\frac{t_{is}\bar{b}_r}{t_{rs}} = -t_{is}\Theta_1 \geq 0$ teljesül.

Mivel minden esetet számba vettünk, így megmutattuk hogy megengedett változó nem válik negatívvá a báziscsere folyamán, de az x_r korábban nem megengedett vezérváltozó megengedetté vált, így $\left|\mathcal{I}_{B'}^\oplus\right| > \left|\mathcal{I}_B^\oplus\right|$. Ezzel állításunkat beláttuk. ∎

A következő állítás azzal az esettel foglalkozik, amikor nem a vezérváltozó sorában választunk pivot pozíciót. Ha az aktuális bázisunk nem lokálisan erősen degenerált a vizsgált nem bázis változóra nézve, akkor az algoritmus által választott báziscsere továbbra is x_r növelő lesz.

3.33. Állítás. *Az algoritmus egy adott lépésében az aktuális B bázis esetén legyen* $r \in \mathcal{I}_B^-$ *és* $q \in \mathcal{I}_B^\oplus$, $t_{qs} > 0$. *Tegyük fel, hogy*

$$0 \leq \frac{\bar{b}_q}{t_{qs}} = \Theta_2 < \Theta_1 = \frac{\bar{b}_r}{t_{rs}},$$

ahol $\bar{b}_r < 0$, $t_{rs} < 0$, $\bar{b}_q \geq 0$ *és* $t_{qs} > 0$. *Ebben az esetben az algoritmus a* t_{qs} *elemen végez báziscserét. Jelölje a báziscsere utáni bázist* B'. *Ekkor* $\mathcal{I}_B^\oplus \backslash \{q\} \subseteq \mathcal{I}_{B'}^\oplus \backslash \{s\}$ *és* $0 > b_r^+ \geq \bar{b}_r$. *Amennyiben a bázis nem degenerált, vagy lokálisan gyengén degenerált az s indexre nézve, akkor* $b_r^+ > \bar{b}_r$.

Bizonyítás. Az előző állításban bevezetett jelöléseket és gondolatmenetet használva a hányadostesztekből adódik, hogy tetszőleges $i \in \mathcal{I}_B^\oplus$ index esetén $i \in \mathcal{I}_{B'}^\oplus$ teljesül $i \neq q$ esetén.

Továbbá, mivel $b_s^+ = \frac{\bar{b}_q}{t_{qs}} \geq 0$, így $s \in \mathcal{I}_{B'}^\oplus$ és

$$\mathcal{I}_B^\oplus \backslash \{q\} \subseteq \mathcal{I}_{B'}^\oplus \backslash \{s\}$$

biztosítja, hogy egy már megengedett változó ne váljon negatívvá. A vezérváltozó megváltozását $b_r^+ = \bar{b}_r - \frac{t_{rs}\bar{b}_q}{t_{qs}}$ formula adja meg, ahol $-\frac{t_{rs}\bar{b}_q}{t_{qs}} \geq 0$, hiszen $t_{rs} < 0$, $t_{qs} > 0$ és $\bar{b}_q \geq 0$. A $\Theta_2 < \Theta_1$ feltevés miatt $0 \geq \frac{t_{rs}\bar{b}_q}{t_{qs}} > \bar{b}_r$ adódik, és így

$$0 > b_r^+ = \bar{b}_r - \frac{t_{rs}\bar{b}_q}{t_{qs}}.$$

Amennyiben a bázis nem degenerált, vagy lokálisan gyengén degenerált az $s \in \mathcal{I}_N$ indexre, akkor definíció szerint $\Theta_2 > 0$ teljesül, tehát $\frac{\bar{b}_q}{t_{qs}} > 0$ és $-t_{rs}\frac{\bar{b}_q}{t_{qs}} > 0$, így

$$0 > b_r^+ = \bar{b}_r - t_{rs}\frac{\bar{b}_q}{t_{qs}} > \bar{b}_r$$

teljesül. Ezzel állításunkat beláttuk. ∎

Geometriailag a 3.33. Állítást úgy lehet értelmezni, hogy a báziscsere során a vezérváltozó által kijelölt feltételhez közelebbi megoldást kapunk.

A 3.32. és 3.33. Állításokat összefoglalva kapjuk az alábbi következményt.

3.34. Következmény. *Az MBU szimplex algoritmus megengedettségi feladatokra, ha lokálisan nem erősen degenerált bázisokon halad, akkor minden lépésben* x_r *növelő báziscserét hajt végre a vezérváltozóra nézve, és így véges.*

Bizonyítás. Mivel a lehetséges bázisok száma véges, így elég megmutatni hogy az algoritmus nem ciklizál, vagyis ugyanaz a bázis nem fordulhat elő kétszer. Mivel feltettük, hogy a bázisok, amelyeken az algoritmus végighalad nem lehetnek lokálisan erősen degeneráltak, ezért a 3.32. és 3.33. Állításokból következik, hogy az algoritmus minden lépésben x_r növelő báziscserét hajt végre a vezérváltozóra nézve, így minden lépésben vagy egy újabb változó válik megengedetté, vagy pedig a vezérváltozó értéke növekszik, így kétszer ugyanaz a bázis megoldás nem fordulhat elő. ∎

A 3.31. Példa folytatásaként bemutatunk egy a vezérváltozó sorában történő pivotálást is. Ezzel az MBU szimplex algoritmusunk mindkét, nem degenerált pivot fajtájára (x_r növelő pivotálásra) példát adunk.

3.35. Példa. Pivotálásra a vézérváltozó sorában kerül sor, mert $\Theta_1 = 1$ és $\Theta_2 = +\infty$.

	x_2	x_5	
x_1	0	-1	0
x_4	1	-1	1
x_3	1	$-\mathbf{1}$	-1

A tábla továbbra is degenerált, a báziscsere oszlopa egyértelmű, hiszen az x_3 vezérváltozó sorában csak egy negatív elem van. A gyengén degeneráltság miatt és a hányados teszt értelmében, a vezérváltozó sorában elvégezhető a báziscsere.

	x_2	x_3	
x_1	-1	-1	1
x_4	0	-1	2
x_5	-1	-1	1

A 3.32. és 3.33. Állítások és a 3.34. Következmény az MBU-szimplex algoritmus pivotálási szabályát jellemző eredények. Anstreicher és Terlaky cikkükben lineáris programozási feladatra definiált primál MBU-szimplex algoritmusukra hasonló eredményeket igazoltak.

A továbbiakban a vezérváltozó értékének a növekedésére alsó korlátot adunk, és így az algoritmus lépésszámára felső korlátot állítunk elő.

A rövid pivot tábla és a bázismegoldás definiciója alapján a rövid pivot tábla egy oszlopára és az aktuális bázismegoldásra teljesül, hogy $\mathbf{t}_s = B^{-1}\mathbf{a}_s$ és $\bar{\mathbf{b}} = B^{-1}\mathbf{b}$, vagyis a $\mathbf{t}_s$ és $\bar{\mathbf{b}}$ vektorok a $B\mathbf{u} = \mathbf{a}_s$ illetve $B\mathbf{v} = \mathbf{b}$ lineáris egyenletrendszerek egyértelmű megoldásai. A Cramer szabály alapján tetszőleges $i \in I_B$ index esetén

$$t_{is} = \frac{\det(B_{is})}{\det(B)} \quad \text{és} \quad \bar{\mathbf{b}}_i = \frac{\det(B_i)}{\det(B)},$$

ahol a $B_{is} \in \mathbb{R}^{m\times m}$ mátrixot úgy kapjuk, hogy a B bázis i.-ik oszlopát kicseréljük az $\mathbf{a}_s$ vektorra, és hasonlóan, a B_i mátrix a B i. oszlopának $\mathbf{b}$ vektorra cserélésével keletkezik. Egy olyan x_r növelő pivot során, mely nem a vezérváltozó sorában történik, a 3.33. állításban bizonyítottak alapján

$$b_r^+ = \bar{b}_r - \frac{t_{rs}\bar{b}_q}{t_{qs}} = \frac{\det(B_r)}{\det(B)} - \frac{\frac{\det(B_{rs})}{\det(B)}\frac{\det(B_q)}{\det(B)}}{\frac{\det(B_{qs})}{\det(B)}} = \frac{\det(B_r)}{\det(B)} - \frac{\det(B_{rs})\det(B_q)}{\det(B_{qs})\det(B)},$$

ahol

$$-\frac{\det(B_{rs})}{\det(B_{qs})}\frac{\det(B_q)}{\det(B)} > 0$$

teljesül a bázis lokálisan nem erősen degenerált volta miatt. Jelölje

$$\Delta_A := \min\left\{ -\frac{\det(B_{rs})\det(B_q)}{\det(B_{qs})\det(B)} : \begin{array}{l} B \text{ az A reguláris részmátrixa és} \\ \frac{\det(Brs)}{\det(B)} < 0,\ \frac{\det(B_q)}{\det(B)} > 0,\ \frac{\det(B_{qs})}{\det(B)} > 0 \end{array} \right\}$$

a vezérváltozó értékének minimális növekedését. Felhasználva hogy csak lokálisan nem erősen degenerált bázisokat vizsgálunk adódik, hogy $\Delta_A > 0$ véges, és

$$b_r^+ = \bar{b}_r - \frac{t_{rs}\bar{b}_q}{t_{qs}} = \frac{\det(B_r)}{\det(B)} - \frac{\det(B_{rs})\det(B_q)}{\det(B_{qs})\det(B)} \geq \frac{\det(B_r)}{\det(B)} + \Delta_A.$$

Összefoglalva, az x_r növelő pivot vagy megengedetté teszi a vezérváltozót, vagy annak értékét legalább Δ_A-val növeli. Habár a végesség bizonyításához ez már elegendő lenne, a lépésszám felső becsléséhez meg kell becsülnünk a vezérváltozók lehetséges legnagyobb abszolút értékét. Legyen

$$\Delta_{\max} := \max\left\{ -\frac{\det(B_r)}{\det(B)} : \begin{array}{l} sgn(\det(B_r)) = -sgn(\det(B)), \text{ és} \\ B \in \mathbb{R}^{m\times m} \text{ az } A \text{ egy reguláris részmátrixa} \end{array} \right\}$$

a lehetséges legnagyobb abszolútértékű jobboldal a Cramer szabály alapján. Ha a feladat nem triviális, és van olyan bázis mely esetén van negatív jobboldal, akkor $\Delta_{\max}$ pozitív és véges. Legyen $\Delta \in \mathbb{Z}$ és $\Delta := \left\lceil \frac{\Delta_{\max}}{\Delta_A} \right\rceil$.

A következő állításban felső korlátot adunk az algoritmus belső ciklusának maximális hosszára, vagyis azon iterációk számára, melyek során a vezérváltozó értéke növekszik ugyan, de nem válik megengedetté.

3.36. Állítás. *Tegyük fel, hogy az algoritmus lokálisan nem erősen degenerált bázisokon halad végig. Legyen $r \in I_B^-$ az aktuális vezérváltozó indexe. Ekkor az algoritmus legfeljebb Δ báziscserét végez mielőtt a vezérváltozó megengedetté válik, vagy legfeljebb Δ báziscsere alatt kimutatja hogy a megengedettségi feladatnak nincs megoldása.*

Bizonyítás. A definíciók alapján a vezérváltozó legkisebb értéke $-\Delta_{\max}$ lehet, melynek értéke minden iterációban legalább Δ_A-val növekszik, így legfeljebb Δ iterációt végezhet az algoritmus mielőtt a következő x_r növelő pivot a vezérváltozót megengedetté tenné, vagy előállít egy primál infizibilis táblát. ∎

Készen állunk az algoritmus lépésszámának becslésére.

3.37. Tétel. *Tekintsük az (3.1) megengedettségi feladatot, és tegyük fel hogy az algoritmus lokálisan nem erősen degenerált bázisokon halad végig. Ekkor az MBU-szimplex algoritmus legfeljebb $m\Delta$ báziscserét végez a feladat megoldása során.*

Bizonyítás. A 3.36. Állítás alapján az algoritmus legfeljebb Δ báziscserét végez egy-egy vezérváltozó megengedettségének beállítása előtt, vagy elér egy infizibilis táblát. Az algoritmus indulásakor a nemmegengedett változók maximális száma megegyezhet a sorok számával. A 3.32. és 3.33. Állítások alapján, ha egy változó megengedett egy iteráció során akkor az is marad, így az algoritmus nem ciklizálhat, és legfeljebb $m\Delta$ lépésben vagy megoldja a feladatot, vagy kimutatja, hogy a feladatnak nincs megoldása. ∎

Megmutattuk, hogy a nemdegenráltsági feltétel teljesülése esetén az algoritmus véges, és korlátot tudtuk adni a szükséges báziscserék maximális számára. Ez a felső korlát gyakran nagyon durva, így érdekes kérdés lenne olyan feladatosztályokat keresni, amelyek esetén a korlát jól számítható, vagyis könnyen meghatározhatók a Δ_A és $\Delta_{\max}$ számok. Egy nyilvánvaló módon felmerülő ilyen feladatosztály a teljesen unimoduláris mátrixok osztálya, azonban a legtöbb ilyen tulajdonságú feladat erősen degenerált, így a becslés csak a lokálisan nem erősen degenerált báziscserék számára ad egyszerű felső korlátot. Ezen felső korlátra a determinánsok meghatározásából $\mathbf{b} \in \mathbb{Z}^m$ esetén egyszerű számolással a $\mathbf{b}$ oszlopa szerinti kifejtést használva $\Delta \leq \|\mathbf{b}\|_1$ adódik.

Eddig azzal az esettel foglalkoztunk, amikor a $\mathcal{K}_s = \emptyset$ (gyengén degeneráltság) teljesült az MBU-szimplex algoritmus által meglátogatott összes táblán, figyelembe véve az $s \in \mathcal{I}_N$ index kiválasztását is. Nem ez az általános esete, hiszen a degeneráltság (erős degeneráltság), természetesen előforduló jelenség a lineáris egyenlrendszek és lineáris egyenlőtlenségrendszerek megoldása során.

Egy lehetséges degeneráltság elleni **DegEljárás**, a minimál index szabály felhasználásával a következő

(T, l) = **DegEljárás(**$T, \mathcal{I}_B^0, r$**)**

Bemenő adatok: $\mathcal{J}_r^- := \{j \in \mathcal{I}_N : t_{rj} < 0\}$

begin

  while $(\mathcal{J}_r^- \neq \emptyset)$ **do**

    Legyen $k \in \mathcal{J}_r^-$ a minimális index.

    Legyen $\mathcal{I}_k^+ = \{i \in \mathcal{I}_B^0 \mid t_{ik} > 0\})$

    if $(\mathcal{I}_k^+ = \emptyset)$ **then**

      A hívó tábla gyengén degenerált, **Return(**T, k**)**.

    endif

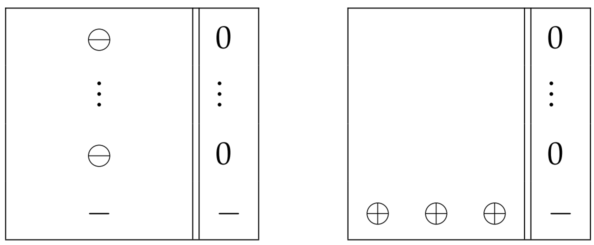
    Legyen $l \in \mathcal{I}_k^+$ a minimális index.

    Pivotálás a t_{lk} elemen.

  endwhile

  A hívó tábla nem megengedett, **Return(**$T, -1$**)**.

end



Gyengén degenerált   Nem megengedett

3.4. ábra. A degeneráció elleni eljárás leállási táblái a degenerált részfeladatra és a vezérváltozó sorára megszorítva.

A degeneráltsági eljárás nyilván a pivot táblának csupán azt a részét használja, amely a degenerált sorokból és a vezér változó sorából ($\mathcal{I}_B \cup \{r\}$) áll. Ennek következtében a degenerációs eljárás leállási táblái a következők: 1. gyengén degenerált tábla, 2. nem megengedett tábla. A degeneráltsági eljárás leállási tábláit a degenerált sorokra és a vezérváltozó sorára megszorítva a 3.4. ábra mutatja be.

Figyeljük meg, hogy a degeneráció elleni eljárás mindkét leállási táblája biztosítja az teljes algoritmus végességét, hiszen a teljes táblára nézve az egyik esetben növelő pivot lehetséges, míg a másik esetben nem megengedettség miatt az algoritmus véget ér. Így elegendő bizonyítani hogy a degeneráció elleni eljárás véges.

3.38. Tétel. *Ha a degenerációs eljárás a minimál indexes pivot elem választási szabályt alkalmazza akkor véges.*

Bizonyítás. A **DegEljárás** esetén a teljes feladat csupán a degenerált sorokból és a vezérváltozó sorából áll.

Tegyük fel indirekt hogy a degeneráció elleni eljárás nem véges. Mivel a bázisok száma véges, így ez csak úgy lehetséges, ha az eljárás ciklizál. Tekintsünk egy minimális méretű ciklizáló példát. Az ilyen tulajdonságú feladat esetén minden változó – a vezérváltozó kivételével – végtelen sokszor mozog.

Jelölje l a legnagyobb mozgó indexű változót és legyen a B' bázis olyan, amely esetén az x_l bázison kívüli változó. Mivel az algoritmus a lehetséges x_i változók közül a legkisebb indexűt választja ki, ez csak úgy lehetséges, hogy ha a B' bázishoz tartozó $t'^{(r)}$ vektorra $t'_{rl} < 0$ illetve $t'_{ri} \geq 0$, bármely $i \in \mathcal{I} - \{l\}$ esetén.

Tekintsük most azt a B'' bázist, mikor az x_l változó legközelebb kilép a bázisból. A belépő változó legyen az x_k. Mivel a minimál indexes szabály, a maximális indexű elemet választja ki kilépésre, ezért a $\mathbf{t}''_k$ vektor esetén $t''_{lk} > 0$, $t''_{rk} < 0$, $t''_{ik} \leq 0$, $i \in \mathcal{I}^0_{B''} \setminus \{l\}$ és $t''_{ik} = 0$, $i \in \mathcal{I}_{N''}$ teljesül.

Az elmondottak alapján, nyilvánvaló, hogy a skaláris szorzatot, a második bázis szerinti index halmazok segítségével kell felírni, mert $t'_{ri}\, t''_{ik} = 0$, bármely $i \in \mathcal{I}_{N''}$ esetén az $\mathbf{t}''_k$ vektor tulajdonságai miatt. Figyelembe véve az 1.48. Ortogonalitási tételt, $\mathbf{t}'^{(r)}$ és $\mathbf{t}''_k$ merőlegesek egymásra, azaz

$$0 = {\mathbf{t}'^{(r)}}^T \mathbf{t}''_k = \sum_{i \in \mathcal{I}^0_{B''} \setminus \{l\}} t'_{ri} t''_{ik} + t'_{rl}\, t''_{lk} + t'_{rr}\, t''_{rk} \leq t'_{rl}\, t''_{lk} + t'_{rr}\, t''_{rk}.$$

Figyelembe véve a $t'_{rl} < 0$ és $t''_{lk} > 0$, illetve $t'_{rr} = 1$ és $t''_{rk} < 0$ feltételeket

$$0 = {\mathbf{t}'^{(r)}}^T \mathbf{t}''_k \leq t'_{rl}\, t''_{lk} + t'_{rr}\, t''_{rk} < 0$$

adódik, ami ellentmond az ortogonalitási tételnek. ∎

Most már készen állunk arra, hogy tetszőleges estre igazoljuk az MBU-szimplex algoritmus végességét a **DegEljárás** végességének a felhasználásával.

3.39. Tétel. *Ha az MBU-szimplex algoritmus a minimál indexes pivot elem választási szabályt alkalmazza akkor véges.*

Bizonyítás. Nem erősen degenerált feladatok esetén a növelő báziscserék biztosítják hogy egy adott bázis ne térhessen vissza. Erősen degenerált pivot táblát a degeneráció elleni eljárás a 3.38. Tétel alapján véges lépésben gyengén degenerálttá konvertálja vagy kimutatja hogy a feladat nem megengedett anélkül, hogy a tábla jobboldalát megváltoztatná. ∎

Gyakorlati szempontból, az MBU-szimplex algoritmus használata során elegendő, akkor alkalmazni a minimál indexes pivot elem választási szabályt, amikor a **DegEljárás** elindul, hiszen azon kívül biztosított, hogy egyetlen korábbi bázis sem térhet vissza.

Az MBU-szimplex algoritmus használatát a következő példán illusztráljuk és azonnal össze is hasonlítjuk a criss-cross algoritmus egy variánsának iterációival.

3.40. Példa. Oldjuk meg a a következő megengedettségi feladatot a *criss-cross* illetve az *MBU–szimplex* algoritmusokkal

$$\begin{array}{rcl} x_1 + 2x_2 \qquad - x_4 + x_5 & \leq & 30 \\ x_1 \qquad + 2x_3 + x_4 + x_5 & = & 80 \\ 2x_2 \qquad\qquad + x_5 & \geq & 40 \\ x_1,\, x_2,\, x_3,\, x_4,\, x_5 & \geq & 0 \end{array}$$

Az induló bázistábla előállítása érdekében a következő átalakításokat végezzük el:

- A második sort osztottuk 2-vel, így az x_3 változó a második bázis változó lett.
- A harmadik sort szoroztuk -1-gyel annak érdekében, hogy kisebb egyenlő típusú egyenlőtlenséget kapjunk.
- Az első és harmadik egyenlőtlenségeket, az s_1 és s_3 eltérés változók bevezetésével egyenletekké alakítottuk át. Az s_1 és s_3 eltérés változók, természetesen előjelkötött változók lesznek, hiszen az egyenlőtlenség bal- és a jobboldala közötti eltérést méri. As s_1 és s_3 az első illetve harmadik bázisváltozó lesz. Mivel rövid bázis táblával dolgozunk ezért az oszlopaikkal (első illetve harmadik egységvektor) nem egészítettük ki a bázis táblát.

	x_1	x_2	x_3	x_4	x_5	b
s_1	1	2	0	-1	1	30
x_3	$\frac{1}{2}$	0	1	$\frac{1}{2}$	$\frac{1}{2}$	40
s_3	0	-2	0	0	-1	-40

Zöld illetve piros színekkel az MBU-szimplex illetve a criss-cross algoritmusnak megfelelő pivot pozíciókat jelöltük ki. Azonnal látható a két algoritmus közötti lényeges különbség: a criss-cross módszer mohó módon próbál pivotálni, egyetlen célt tartva szemelőtt, hogy az éppen nem megengedett feltételt megengedetté tegye. Ezzel szemben, az MBU-szimplex algoritmus, elsődlegesen azokat a feltételeket, amelyek már megnegedettek, szeretné megengedettként megtartani és ha lehetőség kínálkozik, akkor egy nem megengedett feltételt megengedetté tenni.

Először végezzük el a criss-cross algoritmus által kijelölt pivotot: figyelembe véve a minimál index szabályt, a pivot pozíció $t_{32} = -2$ és ekkor az s_3 távozik és az x_2 belép a bázisba. A következő bázis táblához jutunk.

	x_1	x_2	x_3	x_4	x_5	b
s_1	1	0	0	-1	0	-10
x_3	$\frac{1}{2}$	0	1	$\frac{1}{2}$	$\frac{1}{2}$	40
x_2	0	1	0	0	$\frac{1}{2}$	20

Az új bázis táblán, az első feltétel nem megengedetté vált. Alkalmazzuk a criss-cross algoritmust: egyetlen szóbajövő pivot pozíció van a $t_{14} = -1$. Ezen pivotálva s_1 távozik a bázisból és x_4 bekerül a bázisba, így a következő megnegdett bázis táblához jutunk:

	x_1	x_2	x_3	x_4	x_5	b
x_4	-1	0	0	1	0	10
x_3	1	0	1	0	$\frac{1}{2}$	35
x_2	0	1	0	0	$\frac{1}{2}$	20

A criss-cross algoritmus az s_1, x_3, s_3 bázisból, az s_1, x_3, x_2 bázison keresztül jutott el az x_4, x_3, x_2 megengedett bázishoz.

A megengedett bázis megoldás $x_1 = x_5 = 0, x_2 = 20, x_3 = 35$ és $x_4 = 10$. Mivel az s_1 és s_3 távoztak a bázisból, így nem bázis változóként $s_1 = s_3 = 0$ adódik, ami egyben azt is jelenti, hogy az első és a harmadik feltétel is egyenlőséggel teljesül.

Annak érdekében, hogy megvizsgáljuk milyen bázisokon keresztül jut el megengedett bázis megoldásig az MBU-szimplex algoritmus, térjünk vissza az induló bázis táblához.

Az MBU-szimplex algoritmus vezér változónak az s_3 (nem megengedett) változót jelöli ki. (Mivel csak egy nem megengedett változó volt, ezért nem kellett minimál index szabályt alkalmazni.) A vezér változó sorában két lehetséges pivot pozíció volt, ezért a minimál index szabály segítségével a t_{32} elemet jelöltük ki lehetséges pivot pozíciónak. Az algoritmusnak megfelelően ki kell számolnunk a Θ_1 értéket illetve a Θ_2 értéket a részleges hányados teszt alkalmazásával. Esetünkben

$$\Theta_1 = \frac{-40}{-2} = 20 \quad \text{és} \quad \Theta_2 = \frac{30}{2} = 15,$$

azaz $\Theta_2 < \Theta_1$ és ezért a $t_{12} = 2$ lesz a pivot pozíció, vagyis s_1 távozik a bázisból és helyére x_2 lép be. Mivel nem a vezér változó sorában történt a pivotálás, ezért a vezér változó értéke továbbra is nem megengedett lesz, de értéke nőtt és az s_3

változatlanul vezér változó lesz a következő bázis táblán.

$$\begin{array}{c|ccccc|c|}
 & x_1 & x_2 & x_3 & x_4 & x_5 & b \\ \hline
x_2 & \frac{1}{2} & 1 & 0 & -\frac{1}{2} & \frac{1}{2} & 15 \\
x_3 & \frac{1}{2} & 0 & 1 & \frac{1}{2} & \frac{1}{2} & 40 \\
s_3 & 1 & 0 & 0 & \mathbf{-1} & 0 & -10 \\ \hline
\end{array}$$

A harmadik egyenlet nem megengedett (a vezér változóhoz tartozó feltétel) és a sorában egyetlen szóbajövő pivot pozíció, a $t_{34} = -1$ található. Az MBU-szimplex algoritmus előírásának megfelelően kiszámítjuk a Θ_1 és Θ_2 értékeket

$$\Theta_1 = \frac{-10}{-1} = 10 \quad \text{és} \quad \Theta_2 = \frac{40}{\frac{1}{2}} = 80,$$

azaz $\Theta_1 < \Theta_2$ és így a $t_{34} = -1$ lesz a pivot pozíció. Elvégezve a pivotálást a harmadiknak megadott bázistáblához jutunk, igaz, olyan formában, amikor az első ás a harmadik feltételeknek megfelelő egyenletek helyet cseréltek.

Az MBU-szimplex algoritmus az s_1, x_3, s_3 bázisból, az x_2, x_3, s_3 bázison keresztül jutott el az x_2, x_3, x_4 megengedett bázishoz, tehát a második bázis a criss-cross és az MBU-szimplex algoritmusok esetén eltérő volt, annak elelnére a kiindulási bázis tábla és a megtalált megengedett bázis tábla azonos.

3.4. Kúpok

A 0-csúcsú, konvex, kúp fogalmát már a poliéderek tárgyalásakor bevezettük.

3.41. Definíció. A $\mathcal{C} \subseteq \mathbb{R}^n$ *poliedrikus kúp*, (vagy *metszet-kúp*), ha létezik egy $A \in \mathbb{R}^{m \times n}$ mátrix, amelyre

$$\mathcal{C} = \{\mathbf{x} \in \mathbb{R}^n \mid A\,\mathbf{x} \leq \mathbf{0}\}.$$

Az $A \in \mathbb{R}^{m \times n}$ mátrix oszlop vektorai segítségével, definiálhatunk egy másfajta kúpot is.

3.42. Definíció. Legyen $\mathcal{A} = \{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n\} \subseteq \mathbb{R}^m$ adott véges vektor rendszer és jelölje $\mathcal{J}$ a vektorok indexhalmazát. A

$$\mathcal{C}(\mathcal{A}) = \mathcal{C}(\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n) = \{\mathbf{b} \in \mathbb{R}^m \mid \mathbf{b} = \sum_{j=1}^{n} x_j \mathbf{a}_j,\ x_j \geq 0,\ \forall j \in \mathcal{J}\}$$

halmazt, *végesen generált kúp*nak nevezzük.

Mindkét definíció esetén könnyen megmutatható, hogy jogosan használjuk a kúp elnevezést, sőt az is könnyen belátható, hogy konvex, zárt kúpot kapunk mindkét esetben.

A végesen generált kúp elemei tehát, olyan $\mathbf{b} \in \mathbb{R}^m$ vektorok, amelyek előállnak az $\mathbf{a}_i$ vektorok nem negatív lineáris kombinációjaként. Nyilvánvaló, kapcsolat áll fenn a (3.1) lineáris egyenlőtlenségrendszer megoldhatósága és a $\mathbf{b} \in \mathcal{C}(\mathcal{A})$ között, amikor az $A \in \mathbb{R}^{m\times n}$ mátrix oszlop vektoraiból áll az $\mathcal{A}$ halmaz.

Az $A \in \mathbb{R}^{m\times n}$ mátrix oszlop vektoraiból álló $\mathcal{A}$ halmaz segítségével, egy újabb kúpot definiálhatunk.

3.43. Definíció. A $\mathcal{C}(\mathcal{A}) = \mathcal{C}(\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n) \subseteq \mathbb{R}^m$ végesen generált kúp *polárisán* a következő poliedrikus kúpot értjük

$$\mathcal{C}^*(\mathcal{A}) = C^*(\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n) = \{\mathbf{y} \in \mathbb{R}^m \mid \mathbf{y}^T\mathbf{a}_j \leq 0,\ \forall j \in \mathcal{J}\}.$$

Nyilván, úgy is felírhtjuk a végesen generált kúp polárisát, hogy

$$\mathcal{C}^*(\mathcal{A}) = \{\mathbf{y} \in \mathbb{R}^m \mid A^T\mathbf{y} \leq \mathbf{0}\},$$

és ebből az alakból látszik, hogy poliedrikus kúpról van szó.

Az $\mathcal{A} = \{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n\} \subseteq \mathbb{R}^m$ véges vektor rendszert egészítsük ki az $\mathbb{R}^m$ tér egység vektoraival, $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_m$ vektorokkal. Az egység vektorok index halmaza legyen $\mathcal{I}$. Majd az így előálló $n+m$ elemű véges vektorendszernek készítsük el az induló bázis tábláját, amelynél a bázis vektorok az egység vektorok és a bázison kívüli vektorok pedig az $\mathbf{a}_j$ vektorok. A szokásos módon cseréljünk ki annyi $\mathbf{e}_i$ vektort, $\mathbf{a}_j$ vektorral, amennyit csak lehet. Ekkor a következő bázis táblához jutunk.

	$\mathcal{J}$	$\mathcal{I}$
$\mathcal{J}_B$		
$\mathcal{I}_B$	0	

Mivel az $\mathbf{e}_i$ és $\mathbf{a}_j$ vektorok között a pivotálás leállt, ezért tudjuk, hogy $|\mathcal{J}_B|$ az $\mathcal{A} = \{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n\} \subseteq \mathbb{R}^m$ véges vektor rendszer rangja. Az $\{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n, \mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_m\}$ véges vektor rendszer összes olyan bázis tábláját, amely maximális számú $\mathbf{a}_j$ vektort tartalmaz a bázisban, egyszerűen *maximális* bázis tábláknak nevezzük. Az összes ilyen bázis táblát pivotálással elő állíthatjuk.

Ezek a maximális bázis táblák segítségével definiálhatjuk az $\mathbf{v}_r, \mathbf{u}_s \in \mathbb{R}^m$ vektorokat, a következő módon

$$v_{ri} = t_{ri}, \;\; i \in \mathcal{I}, \; r \in \mathcal{J}_B \qquad \text{és} \qquad u_{si} = t_{si}, \;\; i \in \mathcal{I}, \; s \in \mathcal{I}_B.$$

Definiáljuk továbbá az

$$\mathcal{U}_1 = \{\pm \mathbf{u}_s \mid s \in \mathcal{I}_B\} \qquad \text{és} \qquad \mathcal{U}_2 = \{-\mathbf{v}_r \mid r \in \mathcal{J}_B \;\text{ és } t_{ri} \geq 0, \; \forall i \in \mathcal{J}\}$$

véges, az $\mathbb{R}^m$ térbe tartozó vektor rendszereket.

3.44. Feladat. Bizonyítsa be, hogy

- az $\mathcal{U}_1$ vektor rendszer által generált kúp egy altér,
- az $\mathcal{U}_2$ vektor rendszer által generált kúp nem tartalmaz egyenest.

Figyelembe véve az $\mathbf{u}_s$ és $\mathbf{v}_r$ vektorok konstrukcióját, a bázisok egymásba transzformálhatóságát (1.38. Lemma) és a kompozíciós lemmát (1.49. Következmény) az alábbi egyenlőtlenségek adódnak

$$\mathbf{u}^T\mathbf{a}_j = 0, \;\; \forall \mathbf{u} \in \mathcal{U}_1, \, \forall j \in \mathcal{J} \qquad \text{és} \qquad \mathbf{v}^T\mathbf{a}_j \leq 0, \;\; \forall \mathbf{v} \in \mathcal{U}_2, \, \forall j \in \mathcal{J}.$$

Legyen

$$\mathcal{Y} = \begin{cases} \mathrm{U}_1 \cup \mathcal{U}_2, & \text{ha} \quad \mathrm{U}_1 \neq \emptyset \;\; \text{vagy} \;\; \mathcal{U}_2 \neq \emptyset \\ \quad 0, & \text{különben} \end{cases}$$

a maximális bázis táblák segítségével definiált véges, az $\mathbb{R}^m$ térbe tartozó vektor rendszer. Ilyen értelemben, az $\mathcal{A} \subset \mathbb{R}^m$ véges vektor rendszerhez rendeltük az $\mathcal{Y} \subset \mathbb{R}^m$ véges vektor rendszert.

Ennyi bevezető után készen állunk arra, hogy a Farkas-lemmára (3.19. Lemma) építve, három híres tételt, Weyl-tétel (1935), Minkowski-tétel (1896) és Farkas-tétel (1898, 1902) bebízonyítsuk.

Hermann Weyl tételének igazolásakor és publikálásakor nem ismerte Farkas Gyula munkásságát, viszont hivatkozik Minkowski-tételére és kifejti, hogy nem érti, Minkowski miért nem vette észre a tétele megfordíthatóságát. Nyilván, Minkowski tisztában volt azzal, hogy Farkas Gyula nem csak a tétele megfordíthatóságát, hanem, amint később látni fogjuk ennél, többet is igazolt. Furcsa, hogy Weyl nem ismerte Farkas Gyula német nyelvű 1902-es dolgozatát, hiszen ugyanabban a folyóiratban jelent meg, mint Minkowski tétele.

A modern szakirodalomban pedig sokan Minkowski-Farkas tételről beszélnek, megfeledkezve Weyl eredményéről. Szerintünk mindhárom állítás nagyon érdekes és szép tétel. Bizonyításunk nyilván eltér az eredeti Minkowski és Weyl bizonyításoktól, hiszen mi mindhárom híres tételt lényegében a Farkas-lemmára felépítve igazoljuk.

3.45. Tétel. *(Weyl-tétel, 1935.) Legyen $\mathcal{A} = \{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n\} \subset \mathbb{R}^m$ adott, véges vektor rendszer, ekkor létezik olyan $\mathcal{Z} \subset \mathbb{R}^m$ véges vektor rendszer, amelyre*

$$\mathcal{C}(\mathcal{A}) = \mathcal{C}^*(\mathcal{Z})$$

Bizonyítás. Belátjuk, hogy a $\mathcal{Z}$ véges vektor rendszernek megfelel az általunk bázis táblák segítségével megkonstruált $\mathcal{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_k\}$ véges vektor rendszer. A következő két állítást kell igazolnunk:

1. bármely $\mathbf{b} \in C(\mathcal{A})$ vektor esetén $\mathbf{b} \in C^*(\mathcal{Y})$ teljesül,
2. bármely $\mathbf{b} \in C^*(\mathcal{Y})$ vektor esetén $\mathbf{b} \in C(\mathcal{A})$ teljesül.

1. Tegyük fel, hogy $\mathbf{b} \in C(\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n)$, azaz az $A\,\mathbf{x} = \mathbf{b}, \;\; \mathbf{x} \geq \mathbf{0}$ rendszer megoldható. Ekkor az $\mathcal{U}_1$ és az $\mathcal{U}_2$ halmazok konstrukciója miatt, a kompozíciós tulajdonságot (1.49. Következmény) felhasználva, azt kapjuk, hogy az $\mathbf{y}_i^T \mathbf{a}_j \leq 0$, teljesül bármely $\mathbf{y}_i \in \mathcal{Y}$ és $\mathbf{a}_j \in \mathcal{A}$ vektorokra. Ekkor az

$$\mathbf{y}_i^T \mathbf{b} = \sum_{j=1}^{n} (\mathbf{y}_i^T \mathbf{a}_j) x_j \leq 0$$

adódik, bármely $\mathbf{y}_i \in \mathcal{Y}$ vektor esetén, tehát a $\mathbf{b} \in C^*(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_k)$. Ezzel beláttuk, hogy

$$C(\mathcal{A}) \subseteq C^*(\mathcal{Y}).$$

2. Belátjuk, hogy a $\mathbf{b} \notin C(\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n)$ feltevésből $\mathbf{b} \notin C^*(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_k)$ következik. Tetszőleges $\mathbf{b} \notin C(\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n)$ esetén, a végesen generált kúp definíciója alapján, az

$$A\mathbf{x} = \mathbf{b}, \;\; \mathbf{x} \geq 0$$

lineáris egyenlőtlenségrendszer nem oldható meg. Két eset lehetséges: a) a lineáris egyenletrendszernek sincsen megoldása, b) a lineáris egyenletrendszernek nincsen nem negatív megoldása.

a) Az $A\mathbf{x} = \mathbf{b}$ lineáris egyenletrendszer sem oldható meg. Alkalmazzuk a Farkas-Minty (2.4. Lemma) és Rouché-Kronecker-Capelli (2.5. Lemma) lemmákat, és nyilvánvaló lesz, hogy az $\mathcal{U}_1$ halmazba az összes olyan vektort beválasztottuk, amely szóba jöhet, a lineáris egyenletrendszer valamely ellentmondásos egyenlete esetén. (Ne felejtsük el, hogy az ellentmondásos egyenletről egy alkalmasan megválasztott bázis esetén válik nyilvánvalóvá, hogy nem lehet megoldása.)

Részletesebben kifejtve, az $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n, \mathbf{b}, \mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_m$ véges vektor rendszernek bármely maximális számú $\mathbf{a}_j$ bázis vektort tartalmazó (rövid) bázis táblája a következő alakú lesz

	$\bar{\mathcal{J}}_B$	b	$\bar{\mathcal{I}}_B$
$\mathcal{J}_B$			
$\mathcal{I}_B$	0	$* \; \vdots \; *$	

és valamely $i \in \mathcal{I}_B$ esetén $t_{ib} \neq 0$ teljesül, mivel a lineáris egyenletrendszer nem oldható meg. Ebben az esetben $\mathbf{u}_i, -\mathbf{u}_i \in \mathcal{U}_1$ vektorok közül, a kompozíciós tulajdonság (1.49. Következmény) alapján, valamelyik vektor skaláris szorzata, a $\mathbf{b}$ vektorral pozitív, vagyis $\mathbf{b} \notin C^*(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_k)$.

b) Az $A\mathbf{x} = \mathbf{b}$ lineáris egyenletrendszer megoldható, de nem létezik nem negatív megoldása, azaz létezik az alábbi struktúrájú pivot tábla

	$\bar{\mathcal{J}}_B$	b	$\bar{\mathcal{I}}_B$
$\mathcal{J}_B$	$\oplus \; \ldots \; \oplus$	$-$	
$\mathcal{I}_B$	0	$0 \; \vdots \; 0$	

azaz $r \in \mathcal{J}_B$ esetén $t_{rj} \geq 0, \quad \forall j \in \mathcal{J}$ és $t_{rb} < 0$ teljesül. Pontosan az ilyen sorokhoz rendeltük az $\mathcal{U}_2$ halmazba tartozó vektorokat, úgy, hogy $-\mathbf{v}_r^T \mathbf{a}_j \leq 0$ teljesüljön bármely $j \in \mathcal{J}$ esetén. Figyelembe véve a kompozíciós tulajdonságot (1.49. Következmény) és azt, hogy a megfelelő vektor ellentetjét vettük be az $\mathcal{U}_2$ halmazba, ezért $-\mathbf{v}_r^T \mathbf{b} > 0$ teljesül. Mivel $\mathcal{U}_2 \subseteq \mathcal{Y}$, ezért $\mathbf{b} \notin C^*(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_k)$. ■

Weyl-tétel bizonyításának a 2. pontját, miután megállapítottuk, hogy $\mathbf{b} \notin C(\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n)$, azaz a (3.1) lineáris egyenlőtlenségrendszer nem oldható meg, akkor természetesen a Farkas-lemma a (3.19. Lemma) alapján nyilvánvaló, hogy az alternatív rendszernek van $\mathbf{y} \in \mathbb{R}^m$ megoldása. Már csak azt kell meggondolni, hogy $\mathbf{y} \in \mathcal{Y}$, de ez a konstrukció miatt igaz, és így a kívánt eredményt kapjuk. Az előző részletes gondolatmenetet így lehetne lerövidíteni.

Nem időrendi sorrendben igazoljuk a végesen generált kúpokkal kapcsolatos három híres tételt, inkább arra törekszünk, hogy egymáshoz fűződő kapcsolatukat kidomborítsuk illetve olyan logikai sorrendbe állítsuk, amelyben a legrövidebb és legegyszerűbb bizonyításokhoz juthatunk. Most következzen Hermann Minkowski tétele.

3.46. Tétel. *(Minkowski-tétel, 1896.) Legyen $\mathcal{A} = \{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n\} \subset \mathbb{R}^m$ adott, véges vektor rendszer, ekkor létezik olyan $\mathcal{Z} \subset \mathbb{R}^m$ véges vektor rendszer, amelyre*

$$C^*(\mathcal{A}) = C(\mathcal{Z}).$$

Bizonyítás. Belátjuk, hogy $\mathcal{Z}$ véges vektor rendszernek választhatjuk az $\mathcal{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_k\}$ véges vektor rendszert, amelynek a konstrukcióját a Weyl-tétel (3.45. Tétel) előtt megadtuk. A következő két állítást kell igazolnunk:

1. $C^*(\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n) \subseteq C(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_k)$
2. $C(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_k) \subseteq C^*(\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n)$.

A második állítás igazolása pontosan ugyanúgy történik, mint a Weyl-tétel (3.45. Tétel) bizonyításában megfogalmazott első állításé, azaz legyen

$$\mathbf{z} \in C(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_k)$$

ekkor a

$$\mathbf{z} = \sum_{i=1}^{n} \mathbf{y}_i x_i, \;\; x_i \geq 0$$

rendszer megoldható. Az $\mathcal{U}_1$ és az $\mathcal{U}_2$ halmazok konstrukciója miatt, a kompozíciós tulajdonságot (1.49. Következmény) felhasználva, azt kapjuk, hogy az

$$\mathbf{a}_j^T \mathbf{y}_i \leq 0$$

teljesül, bármely i és j indexekre, tehát

$$\mathbf{a}_j^T \mathbf{z} = \sum_{i=1}^{n} (\mathbf{a}_j^T \mathbf{y}_i) x_i \leq 0,$$

bármely j esetén, ami pontosan azt jelenti, hogy

$$\mathbf{z} \in C^*(\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n).$$

Az első állítást a következő formában igazoljuk: tetszőleges

$$\mathbf{z} \notin C(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_k) \;\; \text{esetén} \;\; \mathbf{z} \notin C^*(\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n)$$

teljesül. A $\mathbf{z} \notin C(\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_k)$, azt jelenti, hogy a

$$\mathbf{z} = \sum_{t=1}^{k} \mathbf{y}_i x_i, \; x_i \geq 0$$

lineáris egyenlőtlenségrendszernek nem létezik megoldása, tehát a Farkas-lemma (3.19. Lemma) alapján létezik olyan $\mathbf{b} \in \mathbb{R}^m$ vektor, amelyre

$$\mathbf{b}^T \mathbf{y}_i \leq 0 \quad \text{teljesül, bármely } i \text{ indexre} \tag{3.14}$$

és

$$\mathbf{b}^T \mathbf{z} = 1. \tag{3.15}$$

A (3.14) miatt

$$\mathbf{b} \in C^*(\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_k),$$

vagyis

$$\mathbf{b} \in C(\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n) \tag{3.16}$$

adódik a Weyl-tétel alapján. Összegezve, ha inderekt módon feltételezzük, hogy

$$\mathbf{z} \in C^*(\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n)$$

akkor a végesen generált kúp polárisának a definíciója (3.42. Definíció) miatt

$$\mathbf{z}^T \mathbf{a}_j \leq 0$$

teljesül, bármely j index esetén. Figyelembe véve a (3.16) összefüggést, azt kapjuk, hogy

$$A\,\mathbf{x} = \mathbf{b}, \; \mathbf{x} \geq \mathbf{0}$$

lineáris egyenlőtlenségrendszer megoldható. Ezekután számoljuk ki a **b** és **z** vektorok skaláris szorzatát

$$\mathbf{z}^T \mathbf{b} = \sum_{j=1}^{n} (\mathbf{z}^T \mathbf{a}_j)\, x_j \leq 0$$

teljesül az indirekt feltételünk alapján, de ez ellentmond a (3.15) egyenletnek, tehát az indirekt feltevésünket el kell vetnünk, azaz

$$\mathbf{z} \notin \mathbf{C}^*(\mathbf{a_1}, \mathbf{a_2}, \dots, \mathbf{a_n})$$

adódik, teljessé téve a bizonyításunkat. ∎

Az előző tétel alábbi megfogalmazását is szokták Minkowski-tételnek nevezni.

3.47. Következmény. *Tetszőleges kúp pontosan akkor poliedrikus, ha végesen generált.*

Farkas Gyula, lemmája segítségével, a következő tételt igazolta és publikálta 1898-ban magyar nyelven, majd négy évvel később, egy új bizonyítással németül is.

3.48. Tétel. *(Farkas-tétel, 1898, 1902.) Tetszőleges $\mathcal{A} \subset \mathbb{R}^m$ vektor rendszer esetén $C(\mathcal{A})$ végesen generált kúp polárisának polárisa önmaga.*

Bizonyítás. A Weyl-tétel (3.45. Tétel) alapján tudjuk, hogy az $\mathcal{A}$ véges vektor rendszer által generált kúp előáll, mint az $\mathcal{Y}$ rendszer által generált kúp polárisa. Másrészt, a Minkowski-tétel (3.46. Tétel) miatt az $\mathcal{Y}$ által generált kúp megegyezik az $\mathcal{A}$ által generált kúp polárisával, vagyis $C(\mathcal{A})$ polárisának polárisa önmaga. ∎

Egy érdekes feladat előkészítéseként vezessük be a halmazok Minkowski-féle összegét.

3.49. Definíció. Legyenek az $\mathcal{A}, \mathcal{B} \subset \mathbb{R}^m$ nemüres halmazok. Az

$$\mathcal{A} + \mathcal{B} = \{\mathbf{a} + \mathbf{b} \mid \mathbf{a} \in \mathcal{A},\ \mathbf{b} \in \mathcal{B}\},$$

halmazt, a két halmaz *Minkowski-féle összeg*ének nevezzük.

A következő feladat azt mondja ki, hogy adott Euklideszi tér végesen generált kúpjainak a halmaza, a Minkowski-összegre, a szokásos metszet képzésre és a poláris kúp képzésére nézve algebrát alkot.

3.50. Feladat. Legyen $\mathcal{K} := \{C \subset \mathbb{R}^n \mid C \text{ végesen generált kúp }\}$. Bizonyítsa be, hogy bármely $C_1, C_2 \in \mathcal{K}$ esetén a következő állítások igazak

1. $C_1 + C_2 \in \mathcal{K}$,
2. $C_1 \bigcap C_2 \in \mathcal{K}$,
3. $C_1^* \in \mathcal{K}$,
4. $C_1^{**} = C_1$,
5. $(C_1 + C_2)^* = C_1^* \bigcap C_2^*$,
6. $(C_1 \bigcap C_2)^* = C_1^* + C_2^*$.

Az előző feladat megoldása során is előjön a végesen generált kúpok konvexitása. A következő, egyszerű feladat, azt mutatja be, hogy tetszőleges lineáris egyenlőtlenségrendszer is konvex halmazt definiál.

3.51. Feladat. Legyenek az $A \in \mathbb{R}^{k\times n}$, $E \in \mathbb{R}^{l\times n}$, $B \in \mathbb{R}^{k\times m}$, $\in R^{l\times m}$ mátrixok, és a $\mathbf{b} \in \mathbb{R}^k$, $\mathbf{c} \in \mathbb{R}^l$ vektorok. Bizonyítsa be, hogy

$$S = \{(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^{n+m} \mid A\mathbf{x} + B\mathbf{y} = \mathbf{b},\ E\mathbf{x} + F\mathbf{y} \geq \mathbf{c},\ \mathbf{x} \geq \mathbf{0}\}$$

konvex halmaz.

Ezt a fejezetet a Motzkin-tétellel vagy más néven a konvex poliéderek alaptételével, fejezzük be.

Korábban, a 3.14. Tételben kimondtuk és konstruktívan igazoltuk, hogy egy poliéder bármelyik megoldása előáll mint véges sok bázismegoldás konvex kombinációja és egy irány (nem negatív, homogén lineáris megoldás) összegeként. Ez igazából a Motzkin-tétel egyik iránya.

3.52. Tétel. *(Motzkin–tétel, 1936.) Legyen adott a $\mathcal{P} \subset \mathbb{R}^n$ halmaz. A $\mathcal{P}$ pontosan akkor poliéder, ha $\mathcal{P} = \mathcal{Q} + \mathcal{C}$, ahol $\mathcal{Q}$ egy politóp és $\mathcal{C}$ egy végesen generált kúp (vagyis poliedrikus kúp).*

Bizonyítás. **1.** Ha $\mathcal{P}$ poliéder, akkor felírható $\mathcal{Q} + \mathcal{C}$ alakban.
Legyen $\mathcal{P} := \{\mathbf{x} \in \mathbb{R}^n \mid A\mathbf{x} \leq \mathbf{b}\} \subset \mathbb{R}^n$ poliéder, ahol $A \in \mathbb{R}^{m\times n}$ és $\mathbf{b} \in \mathbb{R}^m$. Tekintsük a következő poliedrikus kúpot:

$$\mathcal{P}_1 = \{\, (\mathbf{x}, x_{n+1}) \in \mathbb{R}^{n+1} \ :\ x_{n+1} \in \mathbb{R}_{\oplus},\ A\mathbf{x} - x_{n+1}\mathbf{b} \leq \mathbf{0}\}.$$

Kölcsönösen megfeleltethetők az

$$\mathbf{x} \in \mathcal{P} \quad \text{pontok az} \quad \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix} \in \mathcal{P}_1$$

pontoknak. A $\mathcal{P}_1$ poliedrikus kúp, így a Minkowski-tétel alapján léteznek

$$\begin{pmatrix} \mathbf{y}^1 \\ y^1_{n+1} \end{pmatrix}, \begin{pmatrix} \mathbf{y}^2 \\ y^2_{n+1} \end{pmatrix}, \dots, \begin{pmatrix} \mathbf{y}^p \\ y^p_{n+1} \end{pmatrix}$$

vektorok, amelyekkel

$$\mathcal{P}_1 = \mathcal{C}\left(\begin{pmatrix} \mathbf{y}^1 \\ y^1_{n+1} \end{pmatrix}, \begin{pmatrix} \mathbf{y}^2 \\ y^2_{n+1} \end{pmatrix}, \dots, \begin{pmatrix} \mathbf{y}^p \\ y^p_{n+1} \end{pmatrix} \right)$$

a $\mathcal{P}_1$ előáll, mint végesen generált kúp.

Az általánosság korlátozása nélkül feltehetjük, hogy y^i_{n+1} egyenlő 0 vagy 1. Legyen

$$\mathcal{Q}_1 = conv\left\{ \begin{pmatrix} \mathbf{y}^i \\ y^i_{n+1} \end{pmatrix} : y^i_{n+1} = 1 \right\} \quad \text{és} \quad \mathcal{C}_1 = \mathcal{C}\left(\left\{ \begin{pmatrix} \mathbf{y}^i \\ y^i_{n+1} \end{pmatrix} : y^i_{n+1} = 0 \right\} \right),$$

ekkor $\{\mathbf{x} \in \mathcal{P}_1 |\ x_{n+1} = 1\} = \mathcal{Q}_1 + \mathcal{C}_1$ teljesül. Ezt $\mathbb{R}^n$ térre megszorítva kapjuk a kívánt előállítást.

2. Ha $\mathcal{P}$ felírható $\mathcal{P} = \mathcal{Q} + \mathcal{C}$ alakban, akkor poliéder.
Legyen $\mathcal{Q} = conv\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_r\}$ és $\mathcal{C} = \mathcal{C}(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_s)$. Legyen

$$\mathcal{P}_1 = \mathcal{C}\left(\left\{ \begin{pmatrix} \mathbf{x}_1 \\ 1 \end{pmatrix}, \begin{pmatrix} \mathbf{x}_2 \\ 1 \end{pmatrix}, \ldots, \begin{pmatrix} \mathbf{x}_r \\ 1 \end{pmatrix}, \begin{pmatrix} \mathbf{y}_1 \\ 0 \end{pmatrix}, \begin{pmatrix} \mathbf{y}_2 \\ 0 \end{pmatrix}, \ldots, \begin{pmatrix} \mathbf{y}_s \\ 0 \end{pmatrix} \right\} \right).$$

Ekkor a $\mathbf{z} \in \mathcal{P}$ pontosan akkor teljesül, ha $\begin{pmatrix} \mathbf{z} \\ 1 \end{pmatrix} \in \mathcal{P}_1$. A Weyl-tétel alapján, a végesen generált $\mathcal{P}_1 \subseteq \mathbb{R}^{n+1}$ kúp felírható, mint egy másik véges vektor rendszer által generált kúp polárisa. Legyenek a generáló vektorok $\bar{\mathbf{a}}^{(i)} = (\mathbf{a}^{(i)}, b_i)^T \in \mathbb{R}^{n+1}$ sor vektorok $i = 1, 2, \ldots m$ esetén és jelölje $\bar{\mathbf{z}} = \begin{pmatrix} \mathbf{z} \\ z_{n+1} \end{pmatrix}$ a $\mathcal{P}_1$ elemeit. A $\mathcal{P}_1$ kúp reprezentációja egy végesen generált kúp polárisaként a következő lesz

$$\mathcal{P}_1 = \{\bar{\mathbf{z}} \in \mathbb{R}^{n+1} : \bar{A}\,\bar{\mathbf{z}} \leq \mathbf{0}\}.$$

A lineáris egyenlőtlenségrendszert részletesen kiírva kapjuk, hogy

$$\bar{A}\,\bar{\mathbf{z}} = A\,\mathbf{z} + z_{n+1}\,\mathbf{b} \leq \mathbf{0}$$

Tehát a $\mathbf{z} \in \mathcal{P} \subset \mathbb{R}^n$ pontokat az egy-egyértelmű megfeleltetés miatt

$$A\,\mathbf{z} + 1 \cdot \mathbf{b} \leq \mathbf{0} \quad \text{jellemzi, azaz} \quad A\,\mathbf{z} \leq -\mathbf{b}.$$

Összefoglalva, a $\mathcal{P}$ halmazt a következő módon lehet megadni

$$\mathcal{P} = \{\mathbf{z} \in \mathbb{R}^n : A\,\mathbf{z} \leq -\mathbf{b}\},$$

tehát a $\mathcal{P} \subset \mathbb{R}^n$ halmaz, poliéder. ■

A fejezet lezárásaként említsük meg, hogy Motzkin újra felfedezte Fourier eliminációs módszerét, amelyet lineáris egyenlőtlenségrendszerek megoldására dolgozott ki és ma Fourier-Motzkin eliminációs módszerként tarja számon a szakirodalom. A Fourier-Motzkin eliminációs módszernek, ma már elméleti jelentősége sincsen, hiszen ebben a fejezetben bemutatott módszerek mind egyszerűbbek és mindnek jobb a komplexitása általános esetben is. Gyakorlati jelentősége a Fourier-Motzkin eliminációs módszernek, soha sem volt.

3.5. Végesen generált kúpok: illusztráció és kapcsolat a Farkas-lemmával

Tekintsük újra a végesen generált kúp definícióját és a Farkas-lemmát.

Legyen $\mathcal{A} = \{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n\} \subseteq \mathbb{R}^m$ adott véges vektor rendszer és jelölje $\mathcal{J}$ a vektorok indexhalmazát. A

$$\begin{aligned} \mathcal{C}(\mathcal{A}) &= \mathcal{C}(\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n) = \{\mathbf{b} \in \mathbb{R}^m \mid \mathbf{b} = \sum_{j=1}^{n} x_j \mathbf{a}_j,\ x_j \geq 0,\ \forall j \in \mathcal{J}\} \\ &= \{\mathbf{b} \in \mathbb{R}^m \mid A\,\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0} \ \text{ megoldható}\,\} \end{aligned}$$

halmazt, *végesen generált kúp*nak nevezzük.

Farkas lemma. Az alábbi két lineáris egyenlőtlenségrendszer közül pontosan az egyik oldható meg:

$$\left.\begin{aligned} A\mathbf{x} &= \mathbf{b} \\ \mathbf{x} &\geq \mathbf{0} \end{aligned}\right\} \qquad\qquad \left.\begin{aligned} \mathbf{y}^T A &\leq \mathbf{0} \\ \mathbf{y}^T \mathbf{b} &= 1 \end{aligned}\right\}$$

ahol az $A \in \mathbb{R}^{m \times n}$ mátrix oszlopvektorai az $\mathcal{A} = \{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n\} \subseteq \mathbb{R}^m$ adott véges vektor rendszer elemei és $\mathbf{b} \in \mathbb{R}^m$ egy tetszőleges vektor.

A Farkas-lemma egy egyszerű, szép és geometriai jellegű bizonyítása (vázlatosan) a következő:

1. Megmutatjuk, hogy a $\mathcal{C}(\mathcal{A})$ végesen generált kúp, tulajdonképpen, a nemnegatív ortáns lineáris transzformáltja, ahol a lineáris transzformáció mátrixa, az A mátrix.

2. Bármely $\mathbf{b} \in \mathbb{R}^m$ vektor esetén két lehetséges eset van: $\mathbf{b} \in \mathcal{C}(\mathcal{A})$ vagy $\mathbf{b} \notin \mathcal{C}(\mathcal{A})$.

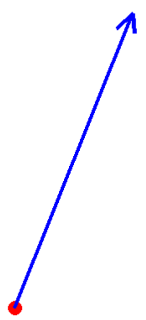
2.a. Ha $\mathbf{b} \in \mathcal{C}(\mathcal{A})$ akkor a Farkas-lemma első rendszerének van megoldása.

2.b. Ha $\mathbf{b} \notin \mathcal{C}(\mathcal{A})$ akkor alkalmaznunk kell a Minkowskitól származó szeparációs tételt és ekkor létezik egy $\mathbf{y} \in \mathbb{R}^m$ vektor, amelyik segítségével definiált hipersík elválasztja a $\mathbf{b}$ vektort és a $\mathcal{C}(\mathcal{A})$ konvex, zárt kúpot. Algebrailag kifejezve, ez pontosan azt jelenti, amit a Farkas-lemma második egyenlőtlenségrendszere fejez ki.

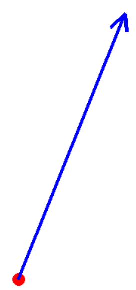
Ennek a bizonyításnak a szépsége a geomatriai jellegéből adódik. Ezt a geometriai jelleget illusztrálja a következő animáció. Az animáció során felhasznált adatok a következők:

$$A = \begin{pmatrix} 0 & 0 & 1 & 2 & 2 & 1 \\ 1 & 2 & 2 & 1 & 0 & 0 \\ 2 & 1 & 0 & 0 & 1 & 2 \end{pmatrix}, \quad \mathbf{b}_1 = \begin{pmatrix} 2 \\ 2 \\ 2 \end{pmatrix}, \quad \mathbf{b}_2 = \begin{pmatrix} 2 \\ -2 \\ -2 \end{pmatrix}.$$

Az animáció elindításához kattintson az animáció számára fenntartott területre (a vektor környékére).



3.5. ábra. A bizonyítás **2.a.** esete az A mátrixszal és $\mathbf{b}_1$ vektorral.



3.6. ábra. A bizonyítás **2.b.** esete az A mátrixszal és $\mathbf{b}_2$ vektorral.

4. fejezet

Lineáris programozás dualitás elmélete

Lineáris programozás (lineáris optimalizálás), az operációkutatás alapvető fontosságú témaköre, amelynek az alap feladata lineáris egyenlőtlenségrendszer esetén, lineáris célfüggvény minimalizálása. Geometriai szempontból egy lineáris függvény szélsőértékére vagyunk kiváncsiak, egy konvex poliéder esetén.

Az általánosság korlátozása nélkül feltehető, hogy a lineáris egyenlőtlenségrendszer a (3.1) formában adott. Ebben az esetben a lineáris programozási feladatot

$$\min\ \mathbf{c}^T\mathbf{x} \quad \text{feltéve, hogy} \quad A\,\mathbf{x} = \mathbf{b}, \quad \mathbf{x} \geq \mathbf{0}, \tag{4.1}$$

alakban fogalmazhatjuk meg, ahol $A \in \mathbb{R}^{m\times n}$, $\mathbf{c}, \mathbf{x} \in \mathbb{R}^n$ és $\mathbf{b} \in \mathbb{R}^m$.

Többben foglalkoztak annak a tudománytörténeti kérdésnek a vizsgálatával, hogy mikor és ki írta fel az első lineáris optimalizálási feladatot. A válaszok közül két érdekeset emelnék ki. Az első, francia hadmérnökök a XVIII. században, erőd építési feladatok kapcsán foglalkoztak az optimális anyag szállítás kérdésével. A másik érdekes lineáris programozási feladatra vezető probléma megfogalmazása az un. pénzváltási feladta volt, amelyet a XII. századi Itáliában jegyeztek le pénzváltók.

A modern lineáris programozás születéséhez többen is hozzájárultak. Farkas Gyula nevét, és eredményeit már korábban említettük. A Farkas-lemmán kívül a legismertebb eredmény a lineáris programozás megoldására szolgáló szimplex-algoritmus, amelyet George B. Dantzig fedezett fel.

Az operációkutatás területén elért eredményekért számos közgadasági Nobel-díjat osztottak ki. Ezek között is előkelő helyen szerepel a lineáris programozás közgazdasági alkalmazásaiért 1975-ben kiadott Nobel-díj, amelyet Kantorovich és Koopmans kapták megosztva.

A (4.1) feladatban megfogalmazott lineáris optimalizálási feladat egyszerűsége ellenére számos érdekes kérdés fogalmazható meg. Ezek közül a feltételek megoldhatóságával kapcsolatos kérdéseket az előző fejezetben vizsgáltuk meg és (pivot) algoritmusokat fogalmaztunk meg a megengedettségi feladat megoldására. Felmerül a kérdés, hogy a lineáris programozási feladat megoldására használhatók lesznek-e korábbi, a megengedettségi feladat megoldására megfogalmazott algoritmusok vagy esetleg tovább fejleszthetők leszenek az optimalizálási feladat megoldására is. Megmutatjuk, hogy mind a criss-cross algoritmusnak, mind pedig az MBU-szimplex algoritmusnak létezik lineáris programozás feladat megoldására alkalmas változata is.

Természetes módon merül fel a pivot algoritmusok (elméleti) hatékonyságának kérdése is, amelyet ebben a jegyzetben csak részben érintünk majd. Az első fontos témakör, amelyet megfogalmazunk és megvizsgálunk az un. dualitás kérdése. A kérdés röviden úgy fogalmazható meg, hogyha szeretnék egy lineáris célfüggvényt optimalizálni egy poliéderen akkor van-e véges minimuma ? Ha van véges minimuma a lineáris programozási feladatnak, akkor annak mi az oka ? Egyik magyarázat lehet az, hogy a poliéder, politop vagyis korlátos és zárt halmaz. Azt is látni fogjuk, hogy nem mindég ez a magyarázata az optimum végességének.

4.1. Lineáris programozás duál feladata

A (4.1) feladatot fogjuk primál lineáris programozási feladatnak nevezni. Könnyen megmutatható, hogy tetszőleges lineáris programozási feladat, a (4.1) feladat formájára hozható, hasonlóan, mint a lineáris egyenlőtlenségrendszerek esetén, amelyek a (3.1) alakú feladatra transzformálhatók.

4.1. Definíció. A lineáris programozási *primál* és *duál* feladatok legyenek a következő alakban adottak

$$\left.\begin{array}{rcl} \min & \mathbf{c}^T\mathbf{x} & \\ A\mathbf{x} & = & \mathbf{b} \\ \mathbf{x} & \geq & \mathbf{0} \end{array}\right\} \ (P) \qquad\qquad \left.\begin{array}{rcl} \max & \mathbf{b}^T\mathbf{y} & \\ A^T\mathbf{y} & \leq & \mathbf{c} \end{array}\right\} \ (D)$$

ahol $A \in \mathbb{R}^{m\times n}$, $\mathbf{c}, \mathbf{x} \in \mathbb{R}^n$ és $\mathbf{y}, \mathbf{b} \in \mathbb{R}^m$. Az általánosság korlátozása nélkül feltehető, hogy $rang(A) = m$. Legyen a *primál* illetve a *duál megengedett megoldás*ok halmaza rendre

$$\mathcal{P} := \{\mathbf{x} \in \mathbb{R}^n_{\oplus} \mid A\mathbf{x} = \mathbf{b}\} \qquad \text{és} \qquad \mathcal{D} := \{\mathbf{y} \in \mathbb{R}^m \mid A^T\mathbf{y} \leq \mathbf{c}\}.$$

A $\mathcal{P}$ és $\mathcal{D}$ halmazok nyilván poliéderek és így a primál- illetve duál lineáris programozási feladat egy n- illetve m-változós lineáris feltételes, lineáris célfüggvényes optimalizálási feladat.

A duál feladatot, szokás a következő módon átalakítani az $\mathbf{s} \in \mathbb{R}^n$ *eltérés vektor* (eltérés változók) bevezetésével

$$\left.\begin{array}{rrcl} \max & \mathbf{b}^T\mathbf{y} & & \\ A^T\mathbf{y} & +\,\mathbf{s} & = & \mathbf{c} \\ & \mathbf{s} & \geq & \mathbf{0} \end{array}\right\} \quad (D),$$

ekkor a megengedett megoldás halmazt az alábbi módon adhatjuk meg

$$\mathcal{D} := \{(\mathbf{y},\mathbf{s}) \in \mathbb{R}^{m+n} \mid A^T\mathbf{y} + \mathbf{s} = \mathbf{c}, \ \ \mathbf{s} \geq \mathbf{0}\}.$$

Az eltérés vektor, az eredeti döntési változó segítségével, kifejezhető

$$\mathbf{s} = \mathbf{c} - A^T\mathbf{y} \quad \text{és} \quad \mathbf{s} \geq \mathbf{0},$$

ahol $\mathbf{y} \in \mathbb{R}^m$.

Most pedig megfogalmazhatjuk az első eredményt a lineáris programozási feladatpárral kapcsolatban.

4.2. Állítás. *(Gyenge dualitás tétel.) Legyenek adottak a (P) és (D) feladatok. Ekkor bármely $\mathbf{x} \in \mathcal{P}$ és $\mathbf{y} \in \mathcal{D}$ esetén $\mathbf{c}^T\mathbf{x} \geq \mathbf{b}^T\mathbf{y}$ és egyenlőség pontosan akkor áll fenn, ha $\mathbf{x}^T\mathbf{s} = 0$, ahol $\mathbf{s} = \mathbf{c} - A^T\mathbf{y}$.*

Bizonyítás. Először belátjuk az egyenlőtlenséget, feltételezve, hogy adott egy tetszőleges $\mathbf{x} \in \mathcal{P}$ primál illetve $\mathbf{y} \in \mathcal{D}$ duál megengedett megoldás. Ekkor a duál feladat eredeti formáját használjuk és az n darab lineáris egyenlőtlenség mindegyikét, a megfelelő indexű, nem negatív x_i változóval megszorozzuk. Az egyenlőtlenség iránya a nem negativitás miatt nem változik meg. A következő lépésben ezeket az egyenlőtlenségeket összeadjuk és mátrix alakban az alábbi módon írhatjuk le

$$\mathbf{c}^T\mathbf{x} \geq (\mathbf{y}^T A)\,\mathbf{x} = \mathbf{y}^T(A\,\mathbf{x}) = \mathbf{y}^T\mathbf{b}.$$

Az első egyenlőség következik a mátrix szorzás tranzitivitásából, míg a második egyenlőség megfelel az $\mathbf{x}$ primál megengedettségének.

Egyenlőség esetén, a célfüggvények különbségét számoljuk ki. Felhasználva a primál megengedettséget

$$0 = \mathbf{c}^T\mathbf{x} - \mathbf{y}^T\mathbf{b} = \mathbf{c}^T\mathbf{x} - \mathbf{y}^T(A\,\mathbf{x}) = (\mathbf{c} - \mathbf{y}^T A)^T\,\mathbf{x} = \mathbf{s}^T\mathbf{x}, \tag{4.2}$$

azt kapjuk, hogy a nemnegatív $\mathbf{x}$ és $\mathbf{s}$ vektorok merőlegesek egymásra. ■

A (4.2) feltételt a lineáris programozás *komplementaritási feltétel*ének nevezzük. Figyelembe véve azt, hogy az $\mathbf{x}$ primál megengedett megoldás és az $\mathbf{s}$ duál eltérés változó, előjel kötött vektorok, a lineáris komplementaritási feltételt a következő formában írhatjuk fel

$$\mathbf{0} = \mathbf{x}\,\mathbf{s} = (x_1\, s_1, x_2\, s_2, \ldots, x_n\, s_n)^T, \tag{4.3}$$

ahol $\mathbf{x}\,\mathbf{s}$ az $\mathbf{x}$ és az $\mathbf{s}$ vektorok *Hadamard-szorzat*át (koordinátánkénti szorzatát) jelöli. Ezek alapján a komplementaritási feltétel a következő formát nyeri

$$x_i\, s_i = 0, \qquad i = 1, 2, \ldots, n.$$

Bevezethetjük a *primál*

$$\mathcal{P}^* := \{\mathbf{x}^* \in \mathcal{P} \,:\, \mathbf{c}^T\mathbf{x}^* \leq \mathbf{c}^T\mathbf{x},\ \ \forall \mathbf{x} \in \mathcal{P}\}$$

illetve a *duál*

$$\mathcal{D}^* := \{\mathbf{y}^* \in \mathcal{D} \,:\, \mathbf{b}^T\mathbf{y}^* \geq \mathbf{b}^T\mathbf{y},\ \ \forall \mathbf{y} \in \mathcal{D}\}.$$

*optimális megoldások halmaz*át. Az optimális megoldások jellemzésére szolgál a következő állítás, amelyet a gyenge dualitás tételből egyszerűen levezethetünk.

4.3. Állítás. *(Gyenge equilibrium tétel) Legyenek adottak a (P) és (D) feladatok. Legyen továbbá $\bar{\mathbf{x}} \in \mathcal{P}$ és $\bar{\mathbf{y}} \in \mathcal{D}$, amelyek esetén $\mathbf{c}^T\bar{\mathbf{x}} = \mathbf{b}^T\bar{\mathbf{y}}$ ekkor $\bar{\mathbf{x}} \in \mathcal{P}^*$ és $\bar{\mathbf{y}} \in \mathcal{D}^*$.*

Bizonyítás. Legyen $\mathbf{y} \in \mathcal{D}$ tetszőleges és alkalmazzuk a gyenge dualitás tételt. Ekkor $\mathbf{b}^T\bar{\mathbf{y}} = \mathbf{c}^T\bar{\mathbf{x}} \geq \mathbf{b}^T\mathbf{y}$, azaz $\bar{\mathbf{y}} \in \mathcal{D}^*$.

Tetszőleges $\mathbf{x} \in \mathcal{P}$ megoldásból kiindulva, alkalmazva a gyenge dualitás tételt az $\mathbf{x}$ és $\bar{\mathbf{y}} \in \mathcal{D}$ vektorokra, valamint felhasználva az állítás feltételét, – az előzőhöz hasonló módon – kapjuk az $\bar{\mathbf{x}} \in \mathcal{P}^*$ összefüggést. ∎

A gyenge dualitás- és gyenge equilibrium tételek következményeit foglaljuk össze a következő feladatban, amelyeknek a bizonyításait az olvasóra bízzuk.

4.4. Feladat. Legyenek adottak a (P) és (D) feladatok.

1. Bármely $\mathbf{x} \in \mathcal{P}$ esetén a (P) feladat célfüggvényértéke, $\mathbf{c}^T\mathbf{x}$, felsőkorlát a (D) feladat optimum értékére.

2. Bármely $\mathbf{y} \in \mathcal{D}$ esetén a (D) feladat célfüggvényértéke, $\mathbf{b}^T\mathbf{y}$, alsókorlát a (P) feladat optimum értékére.

3. Ha $\mathcal{P} \neq \emptyset$ és a (P) feladat célfüggvénye alulról nem korlátos a $\mathcal{P}$ poliéderen, akkor a $\mathcal{D} = \emptyset$.

4. Ha $\mathcal{D} \neq \emptyset$ és a (D) feladat célfüggvénye felülről nem korlátos a $\mathcal{D}$ poliéderen, akkor a $\mathcal{P} = \emptyset$.

5. Ha $\mathcal{D} = \emptyset$ és $\mathcal{P} \neq \emptyset$, akkor a (P) feladat célfüggvénye alulról nem korlátos a $\mathcal{P}$ poliéderen.

6. Ha $\mathcal{D} \neq \emptyset$ és $\mathcal{P} = \emptyset$, akkor a (D) feladat célfüggvénye felülről nem korlátos a $\mathcal{D}$ poliéderen.

A primál lineáris programozási feladathoz, bevezethetjük a *(primál) pivot táblát* az alábbi módon

A	$\mathbf{b}$
$\mathbf{c}^T$	$*$

Könnyen látható, hogy a tábla felső része megfelel a (3.1) megengedettségi feladat (lineáris egyenlőtlenségrendszer) alap feladatához tartozó pivot táblának. Az alsó rész pedig a célfüggvényt reprezentálja, mint egy lineáris egyenletrendszert. A b oszlop és a c sor kereszteződésében álló '$*$' azt reprezentálja, hogy még nem számoltunk ki megoldását a megengedettségi feladat, így a célfüggvény helyettesíti értékét sem ismerhetjük.

4.5. Definíció. Az $A \in \mathbb{R}^{m \times n}$ mátrix $m \times m$-es nem szinguláris, A_B részmátrixát *bázis*nak nevezzük.

Az

$$A\mathbf{x} = \mathbf{b}$$

lineáris egyenletrendszert, az A_B bázis segítségével az alábbi alakban írhatjuk fel

$$A_B\mathbf{x}_B + A_N\mathbf{x}_N = \mathbf{b},$$

ahol az $\mathbf{x}$ vektort a bázis és nem bázis változók szerint partícionáltuk és ugyanezt tettük az A mátrixszal is. Ekkor a lineáris egyenletrendszer *(általános) megoldás*a a következő vektor lesz

$$\mathbf{x}_N = \mathbf{u} \in \mathbb{R}^{n-m}, \quad \text{és} \quad \mathbf{x}_B = A_B^{-1}\mathbf{b} - A_B^{-1}A_N\,\mathbf{u} \in \mathbb{R}^m.$$

Az $\mathbf{x}_B$ a *bázis változó*k vektora, míg az $\mathbf{x}_N$ a *nem bázis változó*ké.

Vezessük be a következő definíciókat:

4.6. Definíció. Legyen adott a (P) és (D) primál-duál lineáris programozási feladatpár.

1. Az $\mathbf{x}_B = A_B^{-1}\mathbf{b},\ \mathbf{x}_N = \mathbf{0}$ megoldást az $A\mathbf{x} = \mathbf{b}$ lineáris egyenletrendszer *bázis megoldás*ának nevezzük.

2. Ha az $\mathbf{x}_B = A_B^{-1}\mathbf{b} \geq \mathbf{0}$ akkor az $(\mathbf{x}_B, \mathbf{x}_N)$ megoldást *(primál) megengedett bázis megoldás*nak és az A_B mátrixot pedig *primál megengedett bázis*nak nevezzük.

3. Az $\mathbf{y} = \mathbf{c}_B^T A_B^{-1} \in \mathbb{R}^m$ vektort *(duál) bázis megoldás*nak nevezzük.

4. Ha a $\mathbf{c}_B^T A_B^{-1} A_N \leq \mathbf{c}_N^T$ teljesül akkor az A_B bázis *duál megengedett bázis*.

Az A_B bázis segítségével az eltérés változót a következő módon is kifejezhetjük

$$\mathbf{s} = \mathbf{c} - \mathbf{y}^T A = \mathbf{c} - \mathbf{c}_B^T A_B^{-1} A$$

tehát a duál megengedettség feltétele az $\mathbf{s} \geq \mathbf{0}$, azaz egy adott A_B bázis esetén

$$\mathbf{s} = (\mathbf{s}_B, \mathbf{s}_N) = (\mathbf{c}_B - \mathbf{c}_B^T A_B^{-1} A_B, \mathbf{c}_N - \mathbf{c}_B^T A_B^{-1} A_N) = (\mathbf{0}, \mathbf{c}_N - \mathbf{c}_B^T A_B^{-1} A_N)$$

a duál megengedettség feltétele az

$$\mathbf{s}_N = \mathbf{c}_N - \mathbf{c}_B^T A_B^{-1} A_N \geq \mathbf{0}, \quad \text{azaz} \quad \mathbf{c}_B^T A_B^{-1} A_N \leq \mathbf{c}_N^T,$$

ahogyan azt az előző definícióban megfogalmaztuk

Észrevehetjük, hogy adott bázis esetén a duál bázis megoldást pont úgy választjuk meg, hogy

$$\mathbf{c}^T\mathbf{x} = \mathbf{c}_B^T\mathbf{x}_B + \mathbf{c}_N^T\mathbf{x}_N = \mathbf{c}_B^T A_B^{-1}\mathbf{b} = \mathbf{b}^T\mathbf{y},$$

teljesüljön, azaz bázis megoldás esetén a primál és duál feladatok célfüggvényértékei megegyeznek. Azt is megmutathatjuk, hogy bázis megoldás esetén a megoldások – a megengedettségtől függetlenül – komplementárisak, ugyanis

$$\mathbf{x}^T\mathbf{s} = \mathbf{x}_B^T\mathbf{s}_B + \mathbf{x}_N^T\mathbf{s}_N = 0$$

figyelembe véve az $\mathbf{x}_N = \mathbf{0}$ és $\mathbf{s}_B = \mathbf{0}$ feltételeket.

A feladatunkat pivotálást használó algoritmusokkal szeretnénk megoldani, ehhez jól használható az A_B bázishoz tartozó teljes illetve rövid pivot tábla:

$A_B^{-1}A$	$A_B^{-1}\mathbf{b}$
$\mathbf{c}^T - \mathbf{c}_B^T A_B^{-1} A$	$-\mathbf{c}_B A_B^{-1}\mathbf{b}$

$A_B^{-1}A_N$	$A_B^{-1}\mathbf{b}$
$\mathbf{c}_N^T - \mathbf{c}_B^T A_B^{-1} A_N$	$-\mathbf{c}_B A_B^{-1}\mathbf{b}$

ahol az elvégzett transzformációkat a következő módon értelmezhetjük:

1. Előállítottuk a lineáris egyenletrendszernek az A_B bázishoz tartozó megoldását.

2. Elimináltuk a célfüggvény sorában a bázis változók célfüggvény együtthatóit. Ezt úgy tettük meg, hogy az x_i bázisváltozó egyenletét megszoroztuk c_i célfüggvény együtthatóval majd kivontuk a célfüggvény sorból.

Tehát az előző (teljes) pivot táblát, az eredeti táblából elemi sor transzformációkkal állítottuk elő. Vezessük be a következő jelöléseket

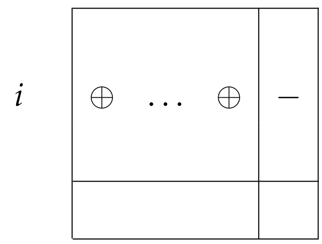
$$T = A_B^{-1}A, \quad \bar{\mathbf{b}} = A_B^{-1}\mathbf{b}, \quad \bar{\mathbf{c}} = \mathbf{c} - \mathbf{c}_B A_B^{-1}A, \quad \zeta = -\mathbf{c}_B A_B^{-1}\mathbf{b} \tag{4.4}$$

Az aktuális nem feltétlenül megengedett $\mathbf{x}_B$ bázismegoldás, megegyezik a táblából kiolvasható $\bar{\mathbf{b}}$ vektorral. Az aktuális nem feltétlenül megengedett bázis megoldáshoz tartozó célfüggvény értéket a $\zeta = -\mathbf{c}_B A_B^{-1}\mathbf{b}$ jelöli. A T mátrix (i, j) indexekhez tartozó eleme t_{ij}. Az A_B bázishoz tartozó bázis változók indexeinek a halmazát jelölje I_B, míg a nem bázis változókét I_N.

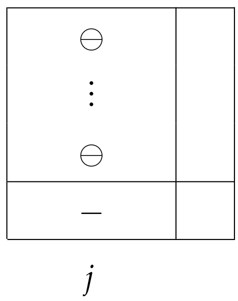
Báziscserét a következő módon terjeszthetjük ki lineáris programozási feladatra:

- elemi sor transzformációkkal egységvektort képezünk a becserélendő vektor oszlopában,
- a bázisba belépő változó, aktuális célfüggvény együtthatóját is elimináljuk megfelelő elemi sor transzformáció segítségével, azaz a célfüggvény sor megfelelő elemét is kinullázzuk,
- az utolsó oszlop értéke (a célfüggvény érték) is módosul az elemi sor tarnszformáció következtében.

Megfogalmazunk három állítást: a primál- illetve duál nem megengedettségi kritériumot és az optimalitási kritériumot.



Primál nem megengedettségi kritérium



Duál nem megengedettségi kritérium

Az előző pivot táblákat már láttuk az előjeles Farkas-Minty lemma következményeként megfogalmazva a 3.22. és 3.23. Feladatokban. A hozzájuk tartozó *nem megengedettségi kritériumok*at a következő feladatban fogalmazzuk meg.

4.7. Feladat. Legyen adott a (P) és (D) lineáris programozási feladatpár.

1. Ha a A_B bázis esetén valamely $\bar{b}_i < 0$, $i \in I_B$ és $t_{ij} \geq 0$ bármely $j \in I_N$ fennáll akkor nem létezik megengedett megoldása a (P) feladatnak.

2. Ha a A_B bázis esetén $\bar{c}_j < 0$, $j \in I_N$ és $t_{ij} \leq 0$, bármely $i \in I_B$ fennáll akkor nem létezik megengedett megoldása a (D) feladatnak.

A (P) és (D) lineáris programozási feladatpárhoz tartozó optimális bázis táblát az alábbiaknak megfelelően adhatjuk meg

$$\begin{array}{|ccc|c|}
\hline
 & & & \oplus \\
 & & & \vdots \\
 & & & \oplus \\
\hline
\oplus & \cdots & \oplus & \\
\hline
\end{array}$$

és az optimalitási kritériumot az alábbi feladatban mondjuk ki.

4.8. Feladat. Legyen adott a (P) és (D) lineáris programozási feladatpár. Ha az A_B primál és duál megengedett bázis akkor az $\mathbf{x} = (\mathbf{x}_B, \mathbf{x}_N) = (A_B^{-1}\mathbf{b}, \mathbf{0})$ optimális megoldása a (P) feladatnak és $\mathbf{y}^T = \mathbf{c}_B^T A_B^{-1}$ optimális megoldása a (D) feladatnak.

Továbbá,

$$\mathbf{c}^T\mathbf{x} = \mathbf{b}^T\mathbf{y}$$

teljesül.

A 4.7. és 4.8. Feladatok bizonyítását az olvasóra bízzuk.

Az optimalitási kritériumot kifejező bázis tábla jelentését a következő mófon fogalmazhatjuk meg: Az aktuális bázis megoldás,

1. primál megengedett bázis megoldás;
2. duál megengedett bázis megoldás;
3. és a megoldások célfüggvény értékei egyenlőek.

Az 1-3. tulajdonságok pontosan a 4.3. Állítás alapján biztosítják a megoldások optimalitását, azaz $\mathbf{x} \in \mathcal{P}^*$ és $\mathbf{y} \in \mathcal{D}^*$. A 3. tulajdonság a 4.2. Állítás következtében a célfüggvények egyenlőségéből levezethető a (4.2) komplementaritási feltétel.

Összefoglalva, a (P) és (D) lineáris programozási feladatpár megoldásainak optimalitásához szükséges és elegendő

- *primál megengedettség*, $\mathbf{x} \in \mathcal{P}$, azaz $A\,\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0}$,
- *duál megengedettség*, $(\mathbf{y}, \mathbf{s}) \in \mathcal{D}$, azaz $\mathbf{y}^T A + \mathbf{s} = \mathbf{c},\ \mathbf{s} \geq \mathbf{0}$,
- *komplementaritás*, azaz $\mathbf{x}^T \mathbf{s} = 0$.

Természetes módon merül fel két kérdés:

1. Hogyan oldhatjuk meg a (P) és (D) lineáris programozási feladatpárokat ?
2. Bármelyik (P) és (D) lineáris programozási feladatpárnak van megoldása ?

Ezekre a kérdésekre keressük a választ a következő részben.

4.2. Criss-cross algoritmus

Lineáris programozási feladat megoldására szolgáló pivot algoritmusok közül egyértelműen a legegyszerűbb a **criss-cross algoritmus**. A criss-cross algoritmus ötlete S. Ziontstól (1969) származik, aki azzal kérdéssel foglalkozott, hogy *megoldható-e az általános lineáris programozási feladat egy fázisban* vagy szükség van a két fázisú szimplex módszerre. Két fázisú szimplex módszer első fázisának a célja (primál) megengedett bázis előállítása, amelyről a szimplex módszer elindítható.

Zionts kutatásait nem koronázta siker, habár megfogalmazta a criss-cross algoritmus prototipusát, de nem sikerült algoritmusának a végességét – általános esetben – bizonyítani. (Ma az az általános vélekedés, hogy a Zionts-féle criss-cross algoritmus nem véges, habár ezt senki se mutatta meg.)

Ziontsot érdeklő kérdést, Terlaky Tamás válaszolta meg 1985-ben, megfogalmazva és minimál index szabály segítségével igazolva, eljárásának a végességét. A Terlaky-féle criss-cross algoritmus, ma már a klasszikus pivot algoritmusok közül, a szimplex módszer mellett, az egyik legismertebb.

Farkas Gyula alapvető fontosságú lemmájának egy bázis táblás formájára, a Farkas-Minty előjeles lemmára (3.18. Lemma), a criss-cross algoritmus lineáris egyenlőtlenségrendszerek megoldására kidolgozott változatával adtunk konstruktív bizonyítást. Ma már a criss-cross módszer egyértelműen egy algoritmus családot takar, amelynek egyik klasszikus algoritmusát a Terlaky-féle, lineáris programozási feladat megoldására szolgáló criss-cross algoritmust megfogalmazzuk és minimál indexes szabály segítségével a végességét igazoljuk.

Legyen adott a (P) és (D) lineáris programozási feladatpár és tegyük fel, hogy ismert egy tetszőleges A_B bázisa. A feladat eredeti adatainak és az aktuális A_B bázis segítségével kiszámolhatóak az aktuális bázistábla adatai a (4.4) képletek segítségével. Ebből kiolvasható az aktuális primál

$$\mathbf{x} = (\mathbf{x}_B, \mathbf{x}_N) = (A_B^{-1}\mathbf{b}, \mathbf{0}) = (\bar{\mathbf{b}}, \mathbf{0})$$

illetve duál bázis megoldás

$$\mathbf{y} = \mathbf{c}_B^T A_B^{-1}, \text{ és } \mathbf{s} = \bar{\mathbf{c}} = \mathbf{c} - \mathbf{c}_B^T A_B^{-1} A$$

ahol

$$\mathbf{s} = (\mathbf{s}_B, \mathbf{s}_N) = (\bar{\mathbf{c}}_B, \bar{\mathbf{c}}_N) = (\mathbf{0}, \mathbf{c}_N - \mathbf{c}_B^T A_B^{-1} A_N).$$

Figyelembe véve, hogy tetszőleges bázis esetén a primál $A\,\mathbf{x} = \mathbf{b}$ feltétel és a duál $\mathbf{y}^T A + \mathbf{s} = \mathbf{c}$ feltétel teljesül, az optimalitási kritérium teljesüléséhez az kell, hogy $\mathbf{x} \geq \mathbf{0}$ és $\mathbf{s} \geq \mathbf{0}$ előjelkötések teljesüljenek. A bázis megoldások struktúrájából láttuk, hogy $\mathbf{x}_N = \mathbf{0}$ és $\mathbf{s}_B = \mathbf{0}$ adódik, így elegendő olyan bázist találni, amelyek esetén

$$\bar{\mathbf{b}} = \mathbf{x}_B \geq \mathbf{0} \quad \text{és} \quad \bar{\mathbf{c}}_N = \mathbf{s}_N \geq \mathbf{0}$$

teljesül. A Terlaky-féle criss-cross algoritmus a lineáris programozási (P) és (D) feladatok megoldása során csupán arra törekszik – teljesen figyelmen kívül hagyva a célfüggvény értékét és annak a változását, – hogy mohó módon találjon egy olyan A_B bázist, amely

1. primál és duál megengedett bázis, vagy
2. igazolja, azt, hogy nem létezik ilyen bázis, azaz vagy előállít egy primál nem megengedett bázist, vagy előállít egy duál nem megengedett bázist.

Fogalmazzuk meg a **minimál indexes criss-cross algoritmus**t a lineáris programozási (P) és (D) feladatok megoldására.

Bemenő adatok: $A \in \mathbb{R}^{m\times n}$, $\mathbf{b} \in \mathbb{R}^m$, $\mathbf{c} \in \mathbb{R}^n$. Legyen adott egy A_B bázis és a bázishoz tartozó tábla adatai T, $\bar{\mathbf{b}}$, $\bar{\mathbf{c}}$, ζ, valamint az index halmazok $\mathcal{I}, \mathcal{I}_B, \mathcal{I}_N$.

1. Ha $\bar{\mathbf{b}} \geq \mathbf{0}$ és $\bar{\mathbf{c}} \geq \mathbf{0}$ teljesül, akkor az *optimalitási kritérium* alapján optimális megoldásnál vagyunk, különben menjünk a 2. lépésre.
2. Legyen $\mathcal{I}_B^- = \{i \in \mathcal{I}_B : \bar{b}_i < 0\}$ és $\mathcal{I}_N^- = \{j \in \mathcal{I}_N : \bar{c}_j < 0\}$. Nyilván $\mathcal{I}^- = \mathcal{I}_B^- \cup \mathcal{I}_N^- \neq \emptyset$ és legyen $r = \min\limits_{k\in\mathcal{I}^-} k$ és menjünk a 3. lépésre.
3. Két eset lehetséges:

   3. a. Duál iteráció: $r \in \mathcal{I}_B^-$ ekkor határozzuk meg a
   $$\mathcal{J}_r^- = \{j \in \mathcal{I}_N : t_{rj} < 0\}$$
   index halmazt. Ha $\mathcal{J}_r^- = \emptyset$ akkor a *primál nem megengedettségi kritérium* alapján
   $$\mathcal{P} = \emptyset.$$
   Ellenkező esetben legyen $s = \min\limits_{k\in\mathcal{J}_r^-} k$ és menjünk a 4. lépésre.

3. b. Primál iteráció: $r \in \mathcal{I}_N^-$ ekkor határozzuk meg a

$$\mathcal{J}_r^- = \{j \in \mathcal{I}_B : t_{rj} > 0\}$$

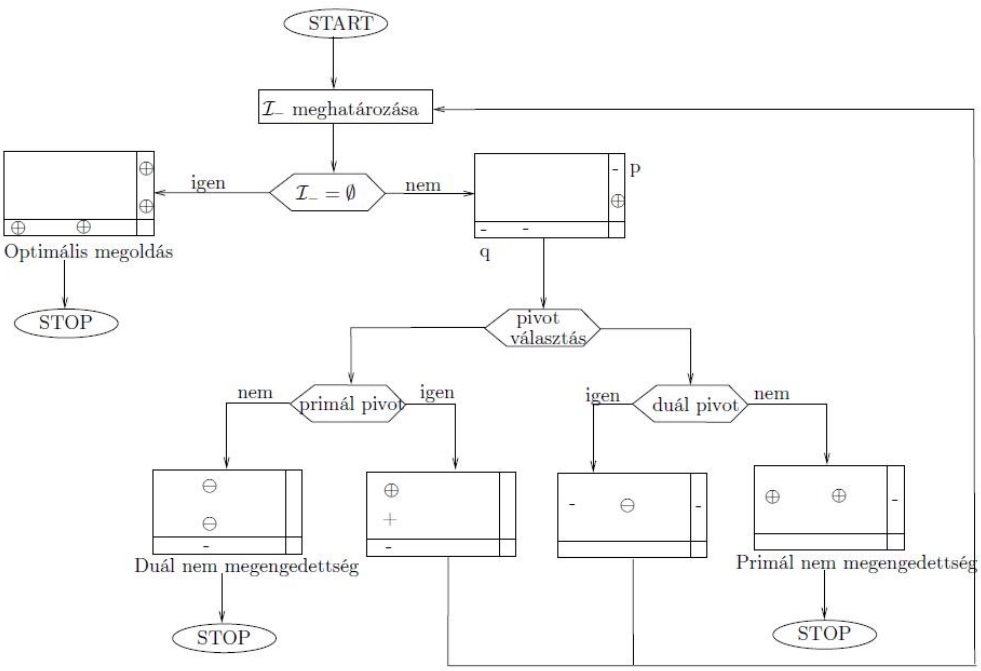
index halmazt. Ha $\mathcal{J}_r^- = \emptyset$ akkor a *duál nem megengedettségi kritérium* alapján

$$\mathcal{D} = \emptyset.$$

Ellenkező esetben legyen $s = \min\limits_{k \in \mathcal{J}_r^-} k$ és menjünk a 4. lépésre.

4. Pivotálás: duál iteráció esetén (3. a.), pivotáljunk a t_{rs} elemen, míg primál iteráció esetén (3. b.), pivotáljunk a t_{sr} elemen. Határozzuk meg az $\mathcal{I}_B$ és $\mathcal{I}_N$ index halmazokat és menjünk az 1. lépésre.

Tehát a lineáris programozási (P) és (D) feladatok megoldására megfogalmazott criss-cross algoritmus valóban a lehetséges eredmények (primál- vagy duál nem megengedettségi kritérium illetve az optimalitási kritérium) valamelyikével áll meg.



4.1. ábra. Criss-cross algoritmus.

Mostmár csak az a kérdés, hogy véges-e az algoritmus, azaz véges sok iteráció alatt eléri-e valamelyik leállási táblát. Figyelembe véve, hogy a (P) feladatnak véges sok bázisa van, ezért a criss-cross algoritmus csak úgy lehetne nem véges, hogy valamelyik bázis tábla, az iterációk során, végtelen sokszor előfordul. Tekintettel arra, hogy az algoritmus determinisztikus és a bázis táblák egymásba transzformálhatók, egy bázis csak akkor térhetne végtelen sokszor vissza, ha legalább két bázis végtelen sokszor előfordul, hiszen az a változó, amelyik az egyik bázis esetén távozik, valamely későbbi másik bázis esetén be kell, hogy lépjen a bázisba, annak érdekében, hogy az elsőnek megfigyelt bázis visszatérhessen. Ezt a jelenséget *ciklizálás*nak nevezzük, vagyis a ciklizálás esetén egy olyan jelenséggel állunk szembe, hogy valamely $B_1, B_2, \ldots, B_k$ bázis oly módon ismétlődik, hogy a $B_{k+1} = B_1$ bázissal és mivel determinisztikus az algoritmus, ezért egymásután, újra és újra végig járja ugyanazokat a bázisokat.

Mielőtt bebizonyítanánk a lineáris programozási (P) és (D) feladatok megoldására megfogalmazott criss-cross algoritmus végességét, foglaljuk össze az algoritmus pszeudokódját:

Criss-cross algoritmus lineáris programozási feladatokra

Bemenő adatok:

egy (P) feladat rövid (bázis) táblája, nem megengedett változók $\mathcal{I}_-$ indexhalmaza (azon i indexek halmaza, amelyekre $\bar{b}_i < 0$ vagy $\bar{c}_i < 0$)

Begin
  while $\mathcal{I}_- \neq \emptyset$ **do**
    begin
      $p := \min\{i \in I_B : \bar{b}_i < 0\}$; $p := \min\{j \in I_N : \bar{c}_j < 0\}$; $r := \min\{p, q\}$;
      if $r = p$
        then
          if $\mathbf{t}^{(p)} \geq \mathbf{0}$ **then** *STOP:* a $P = \emptyset$
                    else legyen $q := \min\{j \in I_N \ : \ t_{pj} < 0\}$;
          endif
        else (i.e. $r = q$)
          if $\mathbf{t}_q \leq \mathbf{0}$ **then** *STOP:* a $D = \emptyset$
                    else legyen $p := \min\{i \in I_B : t_{iq} > 0\}$;
          endif
      endif
      pivotálás a (p, q) pozíción; $I_B := I_B \cup \{q\} \setminus \{p\}$;
      határozzuk meg az $\mathcal{I}_-$ halmazt;
    end
    optimális megoldásnál vagyunk;
  endwhile
end.

Ennyi előkészület után készen állunk bizonyítani a minimál indexes criss-cross algoritmus végességét a (P) és (D) lineáris programozási feladatokra.

4.9. Tétel. *Legyenek adottak a (P) és (D) lineáris programozási feladatok. A minimál indexes criss-cross algoritmus véges sok iterációban megoldja a (P) és (D) lineáris programozási feladatokat és leáll a következő leállási táblák valamelyikével:*

- *egy primál nem megengedettségi tábla, vagy*
- *egy duál nem megengedettségi tábla, vagy*
- *egy optimális tábla.*

Bizonyítás. A minimál indexes criss-cross algoritmus megfogalmazásából világos, hogy az algoritmus megállási kritériumai pontosan megegyeznek a tételben felsorolt leállási táblákkal, azaz az algoritmus leállásakor vagy azt mutatja meg, hogy nem létezik primál megengedett megoldás, vagy azt mutatja meg, hogy nem létezik duál megnegedett megoldás, vagy pedig előállít egy optimális, primál-duál megoldás párt.

A kérdés most már csak az, hogy véges sok lépésben leáll-e a minimál indexes criss-cross algoritmus ?

Indirekt módon tegyük fel, hogy a minimál indexes criss-cross algoritmus nem véges. Ez azt jelenti, hogy létezik olyan példa, amelyen ciklizál, hiszen a lehetséges bázisok száma véges. A minimál indexes criss-cross algoritmus ciklizálási ellenpéldái közül vegyünk egy olyat, amelyet *minimális ciklizálási ellenpéldá*nak nevezünk, abban az értelemben, hogy minden változó a ciklus során belép a bázisba, majd onnan távozik.

Bármely ciklizálási ellenpéldából készíthetünk, minimális ciklizálási ellenpéldát, úgy, hogy azoknak a változóknak megfelelő vektorokat, amelyek a ciklus során vagy végig bázis változók voltak, vagy végig nem bázis változók, azaz azokat, amelyek a ciklus során nem mozogtak, a hozzájuk tartozó oszlop vektorkkal együtt töröljük a véges vektor rendszerünkből (az A mátrix oszlopai közül).

A példa minimalitása miatt, a ciklus során minden változó belép, majd (később) távozik a bázisból. Vizsgáljuk azokat az állapotokat, amikor a legnagyobb indexű változó, az n indexű változó belép a B' bázisba, illetve amikor távozik onnan (B'' bázis). Az alábbi négy eset lehetséges:

(a) primál iterációnál kerül be az B' bázisba és primál iterációnál távozik az B'' bázisból;

(b) primál iterációnál kerül be az B' bázisba és duál iterációnál távozik az B'' bázisból;

(c) duál iterációnál kerül be az B' bázisba és primál iterációnál távozik az B'' bázisból;

(d) duál iterációnál kerül be az B' bázisba és duál iterációnál távozik az B'' bázisból.

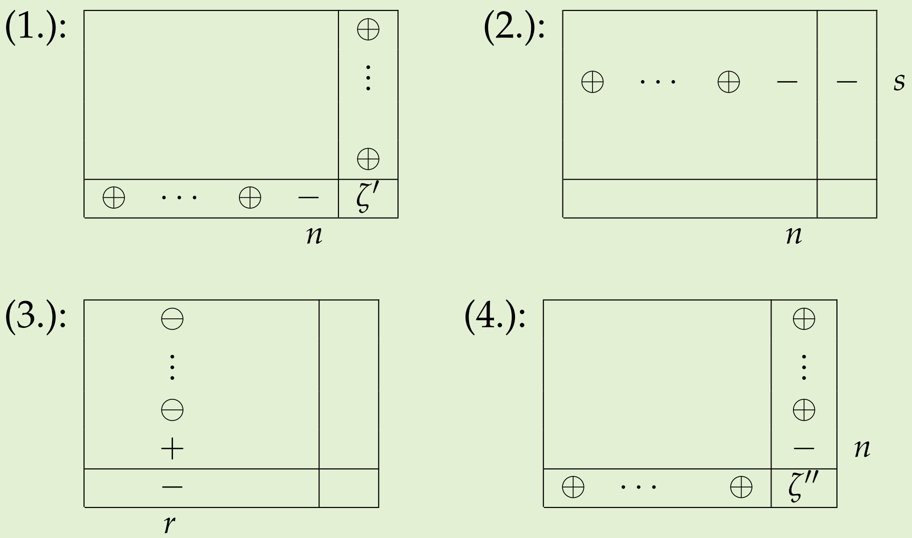
A következő ábrán láthatjuk az n indexű változó lehetséges mozgásait figyelembe véve a minimál index szabályt, azaz azt, hogy ha a minimál index szabály a legnagyobb indexű változót választja ki, akkor minden más változó adatának az előjele megfelelő kell, hogy legyen már. Ezeket az eseteket az un. *majdnem leállási táblák* írják le, azaz

(1.) primál iterációnál belép a bázisba,

(2.) duál iterációnál belép a bázisba,

(3.) primál iterációnál távozik a bázisból, és

(4.) duál iterációnál távozik a bázisból.



Az (1.) – (4.) táblákon látható előjelstruktúra alapján az (a) eset vizsgálata az (1.) és (3.), a (b) eseté az (1.) és (4.), a (c) eseté a (2.) és (3.), míg a (d) eseté a (2.) és (4.) táblák vizsgálatát jelenti.

A bizonyítás menete a következő lesz: az (1.) – (4.) táblákról $\mathbf{t}'$ sor (oszlop) illetve $\mathbf{t}''$ oszlop (sor) vektorokat olvasunk ki, az (a) – (d) lehetséges eseteknek megfelelően. Megmutatjuk, hogy a $\mathbf{t}'$ illetve $\mathbf{t}''$ vektorok együttes előfordulása ellentmond az ortogonalitási tételnek. Ebből pedig az következik, hogy az (a) – (d) esetek egyike sem fordulhat elő, azaz a criss-cross algoritmus nem ciklizálhat, tehát véges, ha pedig véges, akkor valamelyik leállási táblájával ér véget az algoritmus.

Most pedig térjünk rá az (a) – (d) esetek tárgyalására. Az ortogonalitási tételt a bázis tábla $(n+1)$ darab $(m+1)$ dimenziós vektoraira alkalmazzuk. Ehhez az alábbi struktúrájú vektorokat fogunk kiolvasni a táblákból:

$$\mathbf{t} = \begin{array}{|c|c|c|c|c|}\hline I_B \setminus \{n\} & I_N \setminus \{n\} & n & b & c \\ \hline\end{array}$$

A vektorok a B' bázishoz tartozó táblák esetén a következők:
(1.) tábla:

$$\mathbf{t}'^{(c)} = \begin{array}{|c|c|c|c|c|}\hline 0 \;\cdots\; 0 & \oplus \;\cdots\; \oplus & - & \zeta' & 1 \\ \hline\end{array}$$

$$\mathbf{t}'_b = \begin{array}{|c|c|c|c|c|}\hline \oplus \;\cdots\; \oplus & 0 \;\cdots\; 0 & 0 & -1 & \zeta' \\ \hline\end{array}$$

(2.) tábla:

$$\mathbf{t}'^{(s)} = \begin{array}{|c|c|c|c|c|}\hline 0 \;\cdots\; 0 \;\; \underset{s}{1} \;\; 0 \;\cdots\; 0 & \oplus \;\cdots\; \oplus & - & - & 0 \\ \hline\end{array}$$

A B'' bázishoz tartozó táblák esetén a kiolvasott vektorok a következők:
(3.) tábla:

$$\mathbf{t}''_r = \begin{array}{|c|c|c|c|c|}\hline \ominus \;\cdots\; \ominus & 0 \;\cdots\; 0 \;\; \underset{r}{-1} \;\; 0 \;\cdots\; 0 & + & 0 & - \\ \hline\end{array}$$

(4.) tábla:

$$\mathbf{t}''_b = \begin{array}{|c|c|c|c|c|}\hline \oplus \;\cdots\; \oplus & 0 \;\cdots\; 0 & - & -1 & \zeta'' \\ \hline\end{array}$$

$$\mathbf{t}''^{(c)} = \begin{array}{|c|c|c|c|c|}\hline 0 \;\cdots\; 0 & \oplus \;\cdots\; \oplus & 0 & \zeta'' & 1 \\ \hline\end{array}$$

Természetesen ne felejtsük el, hogy $\mathcal{I}_{B'} \neq \mathcal{I}_{B''}$ tehát óvatosan kell elemezzük az eseteket.

Az (a) esetben, ha az (1.) és a (3.) bázis tábla előfordulhat, akkor a $\mathbf{t}'^{(c)}$ és $\mathbf{t}''_r$ vektorok ortogonálisak. Számoljuk ki a két vektor skaláris szorzatát, figyelembe véve, hogy $t'_{ci} \geq 0$ és $t''_{ir} \leq 0$, bármely $i \in \mathcal{I} \setminus \{n\}$ esetén, valamint azt, hogy $t''_{br} = 0$ adódik az első egyenlőtlenség

$$0 = (\mathbf{t}''_r)^T \mathbf{t}'^{(c)} = \sum_{i \in \mathcal{I} \setminus \{n\}} t'_{ci}\, t''_{ir} + t'_{cn}\, t''_{nr} + t'_{cc}\, t''_{cr} + t'_{cb}\, t''_{br} \leq t'_{cn}\, t''_{nr} + t'_{cc}\, t''_{cr} < 0.$$

A második egyenlőtlenség teljesül, mert $t'_{cc} = 1$, $t''_{cr} < 0$, $t'_{cn} < 0$, $t''_{nr} > 0$ adódik az (1.) és (3.) tábla struktúrájából, ellentmondva az ortogonalitási tételnek (1.48. Tétel), tehát az (a) eset nem fordulhat elő.

A (c) esetben, ha az (2.) és a (3.) bázis tábla előfordulhat, akkor a $\mathbf{t}'^{(s)}$ és $\mathbf{t}''_r$ vektorok ortogonálisak. Számoljuk ki a két vektor skaláris szorzatát, figyelembe véve, hogy $t'_{si} \geq 0$ és $t''_{ir} \leq 0$, bármely $i \in \mathcal{I} \setminus \{n\}$ esetén, valamint azt, hogy $t'_{sc} = 0$ és $t''_{br} = 0$ adódik az első egyenlőtlenség

$$0 = (\mathbf{t}''_r)^T \mathbf{t}'^{(s)} = \sum_{i \in \mathcal{I} \setminus \{n\}} t'_{si}\, t''_{ir} + t'_{sn}\, t''_{nr} + t'_{sc}\, t''_{cr} + t'_{sb}\, t''_{br} \leq t'_{sn}\, t''_{nr} < 0.$$

A második egyenlőtlenség teljesül, mert $t'_{sn} < 0$ és $t''_{nr} > 0$ adódik a (2.) és (3.) tábla struktúrájából, ellentmondva az ortogonalitási tételnek (1.48. Tétel), tehát a (c) eset nem fordulhat elő.

A (d) esetben, ha az (2.) és a (4.) bázis tábla előfordulhat, akkor a $\mathbf{t}'^{(s)}$ és $\mathbf{t}''_b$ vektorok ortogonálisak. Számoljuk ki a két vektor skaláris szorzatát, figyelembe véve, hogy $t'_{si} \geq 0$ és $t''_{ib} \geq 0$, bármely $i \in \mathcal{I} \setminus \{n\}$ esetén, valamint azt, hogy $t'_{sc} = 0$, adódik az első egyenlőtlenség

$$0 = (\mathbf{t}''_b)^T \mathbf{t}'^{(s)} = \sum_{i \in \mathcal{I} \setminus \{n\}} t'_{si}\, t''_{ib} + t'_{sn}\, t''_{nb} + t'_{sc}\, t''_{cb} + t'_{sb}\, t''_{bb} \geq t'_{sn}\, t''_{nb} + t'_{sb}\, t''_{bb} > 0.$$

A második egyenlőtlenség teljesül, mert $t'_{sn} < 0$, $t''_{nb} < 0$, $t'_{sb} < 0$ és $t''_{bb} = -1$ adódik a (2.) és (3.) tábla struktúrájából, ellentmondva az ortogonalitási tételnek (1.48. Tétel), tehát a (c) eset nem fordulhat elő.

A (b) esetben, ha az (1.) és a (4.) bázis tábla előfordulhat, akkor a $\mathbf{t}'^{(c)}$ és $\mathbf{t}''_b$ vektorok illetve a $\mathbf{t}''^{(c)}$ és $\mathbf{t}'_b$ vektorok ortogonálisak. Ez az eset az előzőekhez képest bonyolultabb. Számoljuk ki a két-két pár vektor skaláris szorzatát, figyelembe véve, hogy $t'_{ci} \geq 0$, $t''_{ci} \geq 0$, $t'_{ib} \geq 0$ és $t''_{ib} \geq 0$, bármely $i \in \mathcal{I} \setminus \{n\}$ esetén, valamint azt, hogy $t'_{cc} = t''_{cc} = 1$, $t'_{bb} = t''_{bb} = -1$, továbbá $t'_{cb} = \zeta'$ és $t''_{cb} = \zeta''$ adódik az első egyenlőtlenség mindkét skaláris szorzat esetén

$$(\mathbf{t}''_b)^T \mathbf{t}'^{(c)} = \sum_{i \in \mathcal{I} \setminus \{n\}} t'_{ci}\, t''_{ib} + t'_{cn}\, t''_{nb} + t'_{cc}\, t''_{cb} + t'_{cb}\, t''_{bb} \geq t'_{cn}\, t''_{nb} + \zeta'' - \zeta'$$

illetve

$$(\mathbf{t}'_b)^T \mathbf{t}''^{(c)} = \sum_{i \in \mathcal{I} \setminus \{n\}} t''_{ci}\, t'_{ib} + t''_{cn}\, t'_{nb} + t''_{cc}\, t'_{cb} + t''_{cb}\, t'_{bb} \geq \zeta' - \zeta'',$$

mivel $t'_{nb} = t''_{cn} = 0$. A két skaláris szorzat külön-külön is nulla kell, hogy legyen, így az összegük is nyilván nullának kell lennie, azaz

$$0 = (\mathbf{t}''_b)^T \mathbf{t}'^{(c)} + (\mathbf{t}'_b)^T \mathbf{t}''^{(c)} \geq t'_{cn}\, t''_{nb} + \zeta'' - \zeta' + \zeta' - \zeta'' = t'_{cn}\, t''_{nb} > 0.$$

teljesül, mert $t''_{nb} < 0$, $t'_{cn} < 0$ miatt a skaláris szorzatok összege szigorúan pozitív, ellentmondva annak, hogy a két skaláris szorzat külön-külön nulla az ortogonalitási tétel (1.48. Tétel) miatt, tehát a (b) eset sem fordulhat elő.

Mivel az (a) – (d) esetek egyike sem fordulhat elő, ezért a minimál indexes criss-cross algoritmus nem ciklizálhat. ■

Végezetül illusztráljuk egy lineáris programozási feladat megoldásával a minimál indexes criss-cross algoritmus működését.

4.10. Példa. Legyen adott a következő lineáris programozási (primál) feladat

$$\min \quad -x_1 - 2x_2 - 3x_3 - 2x_4 - 3x_5$$

$$\begin{array}{rrrrrrrcr}
x_1 & +2x_2 & & +\ x_4 & +\ x_5 & +x_6 & & = & 110 \\
x_1 & & +2x_3 & +\ x_4 & +\ x_5 & & & = & 80 \\
 & 3x_2 & +\ x_3 & +\ x_4 & +\ x_5 & & & = & 70 \\
 & -2x_2 & & & -\ x_5 & & +x_7 & = & -\ 40
\end{array}$$

ahol $x_1, x_2, \ldots, x_7 \geq 0$. Ekkor a duál feladatot az alábbi formában állíthatjuk elő:

$$\max \quad 110y_1 + 80y_2 + 70y_3 - 40y_4$$

$$\begin{array}{rrrrcr}
y_1 & +\ y_2 & & & \leq & -1 \\
2y_1 & & +\ 3y_3 & -\ 2y_4 & \leq & -2 \\
 & 2y_2 & +\ y_3 & & \leq & -3 \\
y_1 & +\ y_2 & +\ y_3 & & \leq & -2 \\
y_1 & +\ y_2 & +\ y_3 & -\ y_4 & \leq & -3 \\
y_1 & & & & \leq & 0 \\
 & & & y_4 & \leq & 0
\end{array}$$

Az adott (P) feladathoz tartozó pivot tábla

1	2	0	1	1	1	0	110
1	0	2	1	1	0	0	80
0	3	1	1	1	0	0	70
0	-2	0	0	-1	0	1	- 40
-1	-2	-3	-2	-3	0	0	

Válasszuk I_B-t $\{1, 4, 6, 7\}$-nek és I_N-t $\{2, 3, 5\}$-nek. Ekkor kiszámolható az ezen bázishoz tartozó rövid bázis tábla

	x_2	x_3	x_5	$\bar{b}$
x_1	-3	1	0	10
x_4	3	1	1	70
x_6	2	-2	0	30
x_7	-2	0	-1	-40
	1	0	-1	150

$$\begin{aligned} \bar{\mathbf{b}}^T &= (10,\ 0,\ 0,\ 70,\ 0,\ 30,\ -40) \\ \mathbf{y}^T &= \mathbf{c}_B^T A_B^{-1} = (0,\ -1,\ -1,\ 0) \\ \bar{\mathbf{c}}^T &= \mathbf{c}^T - \mathbf{c}_B^T A_B^{-1} A = (0,\ 1,\ 0,\ 0,\ -1,\ 0,\ 0) \end{aligned}$$

Ekkor $x_7 = \bar{b}_4 = -40 < 0$ és $\bar{c}_5 = -1 < 0$, azaz az x_7 primál és az x_5 duál nem megengedett változó. A minimál index szabály alapján x_5 belép a bázisba és x_4 távozik a bázisból. Ekkor a következő pivot táblához jutunk

	x_2	x_3	x_4	$\bar{b}$
x_1	-3	1	0	10
x_5	3	1	1	70
x_6	2	-2	0	30
x_7	1	1	1	30
	4	1	1	220

Figyelembe véve, hogy az aktuális pivot tábla esetén az $\mathcal{I}_- = \emptyset$, optimális táblához jutottunk,

$$\mathbf{x}^T = (10,\ 0,\ 0,\ 0,\ 70,\ 30,\ 30) \in \mathcal{P}^*, \quad \mathbf{y}^T = (0,\ -1,\ -2,\ 0) \in \mathcal{D}^*$$

illetve $\bar{\mathbf{c}}^T = (0,\ 4,\ 1,\ 1,\ 0,\ 0,\ 0)$.

A célfüggvényérték, $\mathbf{c}_B^T B^{-1}\mathbf{b}$, amely helyett a táblában a $\zeta = -\mathbf{c}_B^T B^{-1}\mathbf{b}$ érték található a tábla bal alsó sarkában. Tehát az optimális célfüggvényérték -220.

A tábla optimalitását az $\mathbf{x} \geq \mathbf{0}$ és $\mathbf{s} = \bar{\mathbf{c}} \geq \mathbf{0}$ igazolják.

4.3. Erős dualitástétel

Az erős dualitás tétel a lineáris programozás alapvető tétele, amely kapcsolatot teremt a primál és duál feladatok között. A dualitás tételnek és következményeinek, a duál változóknak közgazdasági jelentésük van, amelyek magyarázatával számos cikk és könyv foglalkozik. A dualitás tételnek ma már számos bizonyítása ismert. A fontosabb bizonyítási módok a következők:

- Farkas-lemma felhasználásával igazoljuk az erős dualitás tételt. (Egyébként az erős dualitás tétel ekvivalens a Farkas-lemmával.)
- A konvex halmazok szeparációs tételének a felhasználásával lehet igazolni az erős dualitás tételt. (A szeparációs tétellel a Farkas-lemma is igazolható.)
- Konstruktív módon igazoljuk az erős dualitás tételt valamely pivot algoritmus végességének a felhasználásával. Ha a bizonyítást a szimplex módszer végességre építjük fel, akkor néhány geometriai jellegű tételre (pl. Motzkin-tétel) is szükségünk van az igazoláskor.

Mi a harmadik módszert alkalmazzuk, de a szimplex algoritmus helyett, amelyet még mindég nem tárgyaltunk, inkább a minimál indexes criss-cross algoritmust használjuk fel, amelynek a végességét az ortogonalitási tétel segítségével, az előző részben láttuk be.

4.11. Tétel. *(Erős dualitástétel.) Legyen adott a (P) és (D) lineáris programozási feladatpár. Ha a $\mathcal{P} \neq \emptyset$ és a $\mathcal{D} \neq \emptyset$ akkor létezik $\bar{\mathbf{x}} \in \mathcal{P}$ és $\bar{\mathbf{y}} \in \mathcal{D}$, amelyekre $\mathbf{c}^T\bar{\mathbf{x}} = \mathbf{b}^T\bar{\mathbf{y}}$, tehát az $\bar{\mathbf{x}}$ és $\bar{\mathbf{y}}$ megoldások, primál- és duál optimális megoldások.*

Bizonyítás. Alkalmazzuk a minimál indexes criss-cross algoritmust a (P) feladatra. A 4.9. Tétel alapján a minimál indexes criss-cross algoritmus véges és a lehetséges három leállási táblájának valamelyikével leáll. A tétel feltételei, $\mathcal{P} \neq \emptyset$ és a $\mathcal{D} \neq \emptyset$, kizárják annak a lehetőségét, hogy nem megengedett táblával álljon le a minimál indexes criss-cross algoritmus. Tehát a tétel feltételeit kielégítő lineáris programozási feladatok esetén a minimál indexes criss-cross algoritmus véges sok lépésben a következő tulajdonságú bázis megoldást állít elő $\bar{\mathbf{x}} \in \mathcal{P}$ és $\bar{\mathbf{y}} \in \mathcal{D}$, ahol

$$\bar{\mathbf{x}} = (\bar{\mathbf{x}}_B, \bar{\mathbf{x}}_N) = (A_B^{-1}\,\mathbf{b}, \mathbf{0}) \quad \text{és} \quad \bar{\mathbf{s}} = \bar{\mathbf{c}} = (\bar{\mathbf{c}}_B, \bar{\mathbf{c}}_N) = (\mathbf{0}, \mathbf{c}_N - \mathbf{c}_B^T\,A_B^{-1}\,A_N).$$

Továbbá $\bar{\mathbf{y}} = \mathbf{c}_B^T\,A_B^{-1} \in \mathcal{D}$, mert $\bar{\mathbf{s}} \geq \mathbf{0}$. Az előállított primál- és duál megengedett bázis megoldásról megmutatjuk, hogy a célfüggvényértékeik egyenlőek és így a gyenge equilibrium tétel (4.3. Állítás) miatt $\bar{\mathbf{x}} \in \mathcal{P}^*$ és $\bar{\mathbf{y}} \in \mathcal{D}^*$, azaz a megoldások optimálisak.

Tehát csak azt kell belátnunk, hogy a primál- és duál célfüggvényértékek azonosak, azaz

$$\mathbf{c}^T\bar{\mathbf{x}} = \mathbf{c}_B^T\,\bar{\mathbf{x}}_B + \mathbf{c}_N^T\,\bar{\mathbf{x}}_N = \mathbf{c}_B^T\,A_B^{-1}\,\mathbf{b} + \mathbf{c}_N^T\,\mathbf{0} = \mathbf{c}_B^T\,A_B^{-1}\,\mathbf{b} = \bar{\mathbf{y}}^T\mathbf{b}.$$

■

A lineáris programozás erős dualitás tételének több formája is ismeretes. Kimondjuk a következő két alakját és az olvasóra bízzuk az állítások igazolását.

4.12. Feladat. Legyenek adottak a (P) és (D) lineáris programozási feladatok.

1. Ha a (P) feladatnak létezik optimális megoldása akkor a (D) feladatnak is létezik és az optimum értékek megegyeznek.
2. Ha a $\mathcal{D}^* \neq \emptyset$ akkor a $\mathcal{P}^* \neq \emptyset$ és az optimum értékei a két feladatnak, megegyeznek.

A Farkas-lemmához hasonlóan a dualitástétel is kimondható általánosabb alakban. Legyen adott a primál- és duál lineáris programozási feladat a következő alalkban.

$$\left.\begin{array}{c} \max\{\mathbf{c}_1^T\mathbf{x}_1 + \mathbf{c}_2^T\mathbf{x}_2\} \\ A_{11}\,\mathbf{x}_1 + A_{12}\,\mathbf{x}_2 = \mathbf{b}_1 \\ A_{21}\,\mathbf{x}_1 + A_{22}\,\mathbf{x}_2 \leq \mathbf{b}_2 \\ \mathbf{x}_2 \geq \mathbf{0} \end{array}\right\} \quad (P_a) \qquad\qquad \left.\begin{array}{c} \min\{\mathbf{b}_1^T\mathbf{y}_1 + \mathbf{b}_2^T\mathbf{y}_2\} \\ \mathbf{y}_1^T A_{11} + \mathbf{y}_2^T A_{21} = \mathbf{c}_1 \\ \mathbf{y}_1^T A_{12} + \mathbf{y}_2^T A_{22} \geq \mathbf{c}_2 \\ \mathbf{y}_2 \leq \mathbf{0} \end{array}\right\} \quad (D_a)$$

Tegyük fel, hogy a feladatban megadott mátrixokkal és vektorokkal a kijelölt műveletek elvégezhetők. Jelölje a megengedett megoldás halmazokat rendre $\mathcal{P}_a$ és $\mathcal{D}_a$ illetve az optimális megoldás halmazokat $\mathcal{P}_a^*$ és $\mathcal{D}_a^*$. Ekkor kimondható és igazolható, a gyenge és erős dualitás tételek is, az általános alakú (P_a) és (D_a) lineáris programozási feladatpárokra.

4.13. Feladat. Legyenek adottak a (P_a) és (D_a) lineáris programozási feladatok.

1. (*Gyenge dualitás tétel.*) Ekkor bármely $(\mathbf{x}_1, \mathbf{x}_2) \in \mathcal{P}_a$ és $(\mathbf{y}_1, \mathbf{y}_2) \in \mathcal{D}_a$ esetén
$$\mathbf{b}_1^T\mathbf{y}_1 + \mathbf{b}_2^T\mathbf{y}_2 \geq \mathbf{c}_1^T\mathbf{x}_1 + \mathbf{c}_2^T\mathbf{x}_2.$$
2. (*Erős dualitás tétel.*) Ha a $\mathcal{P}_a \neq \emptyset$ és a $\mathcal{D}_a \neq \emptyset$ akkor létezik $(\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2) \in \mathcal{P}_a$ és $(\bar{\mathbf{y}}_1, \bar{\mathbf{y}}_2) \in \mathcal{D}_a$, amelyekre
$$\mathbf{c}_1^T\bar{\mathbf{x}}_1 + \mathbf{c}_2^T\bar{\mathbf{x}}_2 = \mathbf{b}_1^T\bar{\mathbf{y}}_1 + \mathbf{b}_2^T\bar{\mathbf{y}}_2,$$
tehát az $(\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2)$ és $(\bar{\mathbf{y}}_1, \bar{\mathbf{y}}_2)$ megoldások, primál- és duál optimális megoldások.

5. fejezet

Lineáris programozás pivot algoritmusai

Lineáris programozási feladat megoldására Dantzig 1947-ben vezette be a szimplex módszert. Dantzig szimplex módszerének szüksége van megengedett, induló bázis megoldásra, ezért eleinte csak a következő alakú lineáris programozási feladatokra működött az algoritmusa:

$$\left.\begin{array}{lcc} \max & \mathbf{c}^T\mathbf{x} & \\ A\mathbf{x} & \leq & \mathbf{b} \\ \mathbf{x} & \geq & \mathbf{0} \end{array}\right\} \quad (P_k) \qquad\qquad \left.\begin{array}{lcc} \min & \mathbf{b}^T\mathbf{y} & \\ A^T\mathbf{y} & \geq & \mathbf{c} \\ \mathbf{y} & \geq & \mathbf{0} \end{array}\right\} \quad (D_k)$$

ahol $A \in \mathbb{R}^{m\times n}$, $\mathbf{c}, \mathbf{x} \in \mathbb{R}^n$ és $\mathbf{y}, \mathbf{b} \in \mathbb{R}^m$. Továbbá tegyük fel, hogy $\mathbf{b} \geq \mathbf{0}$.

A (P_k) és (D_k) lineáris programozási feladatpár esetén, a lineáris egyenlőtlenségrendszer feltételeket az un. *eltérés változó*k bevezetésével egyenlőséges feltételekké tudjuk transzformálni, így a (P_k) feladat egy megnegedett induló bázisát könnyen előtudjuk állítani. Legyenek az $\mathbf{s} \in \mathbb{R}^m_\oplus$ és a $\mathbf{z} \in \mathbb{R}^n_\oplus$ a (P_k) és (D_k) feladatok eltérés vektorai. Ekkor a feladatokat az alábbi, módon írhatjuk fel

$$\left.\begin{array}{lcccc} \max & \mathbf{c}^T\mathbf{x} & +\,\mathbf{0}^T\mathbf{s} & & \\ & A\mathbf{x} & + \quad \mathbf{s} & = \mathbf{b} \\ & \mathbf{x}, & \mathbf{s} & \geq \mathbf{0} \end{array}\right\} \quad (P'_k) \qquad \left.\begin{array}{lcccc} \min & \mathbf{b}^T\mathbf{y} & +\,\mathbf{0}^T\mathbf{z} & & \\ & A^T\mathbf{y} & - \quad \mathbf{z} & = \mathbf{c} \\ & \mathbf{y}, & \mathbf{z} & \geq \mathbf{0} \end{array}\right\} \quad (D'_k)$$

A (P'_k) primál lineáris programozási feladatból azonnal kiolvasható egy induló bázis megoldás

$$\mathbf{x} = \mathbf{0}, \quad \text{és} \quad \mathbf{s} = \mathbf{b} \geq \mathbf{0},$$

ahol a bázis változók az $\mathbf{s}$ vektorba összefoglalt eltérés változók, míg a nem bázis változók, az eredeti döntési változók, amelyek az $\mathbf{x}$ vektorban szerepelnek.

Az induló, megengedett bázis megoldás (és induló megengedett bázis) létezése a feladat struktúrájából és a feltevésekből ($\mathbf{b} \geq \mathbf{0}$) adódik.

5.1. Definíció. A (P) primál lineáris programozási feladatot *kanonikus feladat*nak nevezzük, ha az $A\,\mathbf{x} = \mathbf{b}$ lineáris egyenletrendszer mátrixa tartalmaz egy $(m \times m)$-es egység mátrixot és ha az aktuális jobb oldal vektor $\mathbf{b}$ nem negatív.

Könnyen belátható, hogy a (P_k') feladat, kanonikus lineáris programozási feladat, azaz a (P_k) feladat, eltérés változók bevezetésével, kanonikus feladattá transzformálható. Kanonikus lineáris programozási feladat esetén egyszerűen megadható egy megengedett induló bázis és a hozzá tartozó megengedett, bázis megoldás.

A Dantzig-féle primál szimplex algoritmus *kizárólag* kanonikus lineáris programozási feladatot képes megoldani.

A (P_k) és (D_k) lineáris programozási feladatpár esetén, könnyen megfogalmazhatjuk a gyenge dualitás tételt.

5.2. Feladat. (Gyenge dualitás tétel.) Legyenek adottak a (P_k) és (D_k) feladatok. Ekkor bármely $\mathbf{x} \in \mathcal{P}_k$ és $\mathbf{y} \in \mathcal{D}_k$ esetén $\mathbf{c}^T\mathbf{x} \leq \mathbf{b}^T\mathbf{y}$ és egyenlőség pontosan akkor áll fenn, ha $\mathbf{x}^T\mathbf{z} + \mathbf{y}^T\mathbf{s} = 0$, ahol $\mathbf{s} = \mathbf{b} - A\mathbf{x}$ és $\mathbf{z} = A^T\mathbf{y} - \mathbf{c}$.

Az állítás bizonyítását az olvasóra bízzuk.

Figyelembe véve, hogy a (P_k) maximalizálási feladat, a (P_k) és (D_k) lineáris programozási feladatpár esetén, az optimális bázis tábla előjel struktúrája a következő lesz

$$
\begin{array}{|ccc|c|}
\hline
 & & & \oplus \\
 & & & \vdots \\
 & & & \oplus \\
\hline
\ominus & \cdots & \ominus & \\
\hline
\end{array}
$$

és a duál nem megengedettségi kritériumon kívül ez lesz a másik megállási feltétele a Dantzig-féle primál szimplex algoritmusnak.

5.1. Primál szimplex algoritmus

A primál szimplex algoritmus elindításához, ahogyan azt már említettük, szükség van egy primál megengedett bázisra, ezért a primál szimplex algoritmus kanonikus lineáris programozási feladatról (pl. a (P_k) feladat) indítható el. A primál szimplex algoritmus, primál megengedett bázisról, primál megengedett bázisra lép és közben,

– maximalizálási feladat esetén, – igyekszik növelni a célfüggvény értékét. (Minimalizálási feladat esetén, megengedett bázisokon haladva, természetesen minden iterációban igyekszik csökkenteni a célfüggvény értékét.)

Ha az adott lineáris programozási feladat nem kanonikus feladat, akkor induló megengedett bázis megoldás előállításához un. *két-fázisú szimplex módszer*t kell alkalmazni, amelyet később mutatunk be.

Tekintsük a (P) lineáris programozási feladatot, (4.1), alakban és tegyük fel, hogy ismert egy megengedett bázis, A_B a hozzá tartozó $\mathcal{I}_B$ és $\mathcal{I}_N$ index halmazokkal, $\bar{\mathbf{b}}$ és $\bar{\mathbf{c}}$ transzformált jobboldali illetve célfüggvény (*redukált költség*) vektorokkal. A tábla belsejét $T = A_B^{-1}\,A$ jelöli.

Legyen $x_s,\ s \notin \mathcal{I}_B : \bar{c}_s < 0$ ekkor az A_B bázis, nem lehet duál megengedett (nem optimális). Legyen $\mathcal{I}_B = \{i_1, i_2, \dots, i_m\}$ a bázis indexek halmaza és

$$\zeta = z(\bar{\mathbf{x}}) = \mathbf{c}_B^T \bar{\mathbf{x}}_B = \mathbf{c}_B^T A_B^{-1} \mathbf{b}$$

a bázis megoldás célfüggvényértéke. A teljes pivot tábla $\mathcal{I}_B \times (\mathcal{I}_B \cup \{s, b\})$ részét az alábbi lineáris egyenletrendszer fejezi ki

$$\begin{array}{lllllll}
x_{i_1} & & & & + t_{1s}\, x_s & = & \bar{b}_1 \\
& x_{i_2} & & & + t_{2s}\, x_s & = & \bar{b}_2 \\
& & \ddots & & \vdots & \vdots & \vdots \\
& & & x_{i_m} & + t_{ms}\, x_s & = & \bar{b}_m \\
\zeta & & & & + \bar{c}_s\, x_s & = & z(\bar{\mathbf{x}})
\end{array}$$

ahol az utolsó sor a célfüggvény sort reprezentálja. Mivel a feladatunk minimalizálási feladat, ezért a negatív redukált költségű x_s változót szeretnénk bevonni a bázisba, pozitív λ értéken, mert ekkor a célfüggvény aktuális értéke, $\bar{c}_s\ \lambda$ negatív számmal csökkenne. Végezzük el a kívánt transzformációt, azaz legyen $x_s = \lambda$ és fejezzük ki a bázis változók értékét a λ függvényében. Ekkor a következő lineáris egyenletrendszerhez jutunk

$$\begin{array}{lll}
x_s & = & \lambda \\
x_{i_1} & = & \bar{b}_1 - t_{1s}\lambda \\
x_{i_2} & = & \bar{b}_2 - t_{2s}\lambda \\
\vdots & \vdots & \quad\vdots \\
x_{i_m} & = & \bar{b}_m - t_{ms}\lambda
\end{array}$$

A lineáris egyenletrendszer mátrixának oszlopai affin független vektorok és így

$$\Delta = \Delta(\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_m, \mathbf{t}_s),$$

egy szimplex. Erről, a bázis cseréhez rendelt szimplexről, kapta a módszer a nevét.

Vizsgáljuk meg azt az esetet, amikor az x_s változót bevonjuk a bázisba és az értékét $\lambda \geq 0$ növeljük és bármely $j \notin \mathcal{I}_B \cup \{s\}$ esetén $x_j = 0$ teljesül, azaz a többi nem bázis változó értéke változatlanul nulla marad. (Ezért nem kellett azokat feltüntetni az első $(m \times (m+1))$-es lineáris egyenletrendszer esetén.)

Amikor az x_s változó felveszi a $\lambda \geq 0$ értéket, akkor az előző lineáris egyenletrendszer segítségével fejezhetjük ki a bázis változók értékét a λ érték függvényében. Azt kell biztosítanunk, hogy $x_{i_k} \geq 0$ legyen, ahol $k = 1, 2, \ldots, m$. Két esetünk lehet:

1. $t_{ks} \leq 0$: bármely $\lambda > 0$ esetén az $x_{i_k} = \bar{b}_k - t_{ks}\,\lambda \geq 0$ teljesül;

2. $t_{ks} > 0$: $x_{i_k} \geq 0$ pontosan akkor teljesül, ha $\lambda \leq \frac{\bar{b}_k}{t_{ks}}$.

Az első esetben, mivel a t_{ks} nem pozitív, így bármely pozitív λ esetén, $x_{i_k} \geq 0$ lesz. Ezzel szemben, ha t_{ks} pozitív, akkor a $\lambda \geq 0$ értéket a $\frac{\bar{b}_k}{t_{ks}}$ törttel (hányadossal) lehet felülről korlátozni. Ezt az elemzést kiterjeszthetjük az x_s változó összes együtthatójára, vagyis a pivot tábla teljes $\mathbf{t}_s$ oszlopára. Így jutunk el a szimplex módszer legfontosabb építőkövéhez, a *hányados-teszt*hez, amely biztosítja, hogy iterációról–iterációra primál megnegdetett bázis megoldásokon keresztül, – a poliéder csúcsain, – haladjon a primál szimplex algoritmus.

Vezessük be a $\mathcal{J}_s = \{i \in \mathcal{I}_B \,:\, t_{is} > 0\}$ index halmazt. Két eset lehetséges

1. ha $\mathcal{J}_s \neq \emptyset$ akkor a

$$\lambda = \min_{j \in \mathcal{J}_s} \frac{\bar{b}_j}{t_{js}} \qquad \text{és} \qquad \mathcal{J}_\lambda = \left\{ j \in \mathcal{J}_s \,:\, \lambda = \frac{\bar{b}_j}{t_{js}} \right\}$$

   és bármely $r \in \mathcal{J}_\lambda$ esetén (r, s) megfelelő pivot pozíció és

$$z(\mathbf{x}(\lambda)) = \zeta + \bar{c}_s\,\lambda = \zeta + \bar{c}_s\,\frac{\bar{b}_r}{t_{rs}} \leq \zeta = z(\bar{\mathbf{x}}),$$

2. ha $\mathcal{J}_s = \emptyset$ akkor a λ értékre nincsen korlát és figyelembe véve a célfüggvénynek az A_B bázishoz tartozó alakját, azt kapjuk, hogy

$$\lim_{\lambda \to +\infty} z(\mathbf{x}(\lambda)) = \lim_{\lambda \to +\infty} (\zeta + \bar{c}_s\,\lambda) = \zeta + \lim_{\lambda \to +\infty} \bar{c}_s\,\lambda = -\infty,$$

   mivel $\bar{c}_s < 0$ és $\lambda > 0$.

Ha valamely $j \in \mathcal{I}_B$ esetén $\bar{b}_j = 0$ akkor $\lambda = 0$, azaz $z(\mathbf{x}(\lambda)) = z(\bar{\mathbf{x}})$. Ekkor *(primál) degenerált megoldás*t kapunk és degenerált iterációról beszélünk, mert a célfüggvény érték változatlan maradt.

A $\bar{b}_j = 0$, valamely $j \in \mathcal{I}_B$ esetén, pedig azt jelenti, hogy a $\mathbf{b}$ vektor az $\mathbb{R}^m$ tér valamelyik valódi alterében fekszik. Ez a geometriai jelenség áll a degenerált megoldás hátterében és okozza azt, hogy bizonyos esetekben, – ha a pivot algoritmust nem eléggé körültekintően fogalmaztuk meg, – *ciklizálás* léphet fel. A ciklizálás jelenségével később részletesebben foglalkozunk még.

Visszatérve az előző vizsgálat két esetéhez, észrevehetjük, hogy az első esetben a hányados–tesztet alkalmaztuk, míg a második esetben, azonosítottunk egy olyan irányt, amelyik mentén a célfüggvény korlátozás nélkül csökkenthető volt. A gyenge dualitás tétel egyik következménye alapján, a primál – minimalizálási – célfüggvény alulról nem korlátos, ez pontosan akkor fordulhat elő, ha a $\mathcal{D} = \emptyset$, azaz teljesül a duál nem megengedettségi kritérium. Az eddigi elemzést a következő állítás foglalja össze.

5.3. Feladat. Legyen adott a (P) és (D) lineáris programozási feladat és x_s, $s \notin \mathcal{I}_B : \bar{c}_s < 0$, ahol $\mathcal{I}_B$ az aktuális bázis változók index halmaza. Vezessük be a $\mathcal{J}_s = \{i \in \mathcal{I}_B : t_{is} > 0\}$ index halmazt. Ekkor a következő állítások ekvivalensek

1. $\mathcal{J}_s = \emptyset$,
2. a (P) feladat célfüggvénye alulról nem korlátos,
3. $\mathcal{D} = \emptyset$.

Foglaljuk össze a *primál szimplex algoritmus*t, amelyet Dantzig, 1947-ben fogalmazott meg.

Legyen adott a (P) (minimalizálási) lineáris programozási feladat és egy hozzá tartozó primál megengedett A_B bázis, a rövid bázis táblával együtt.

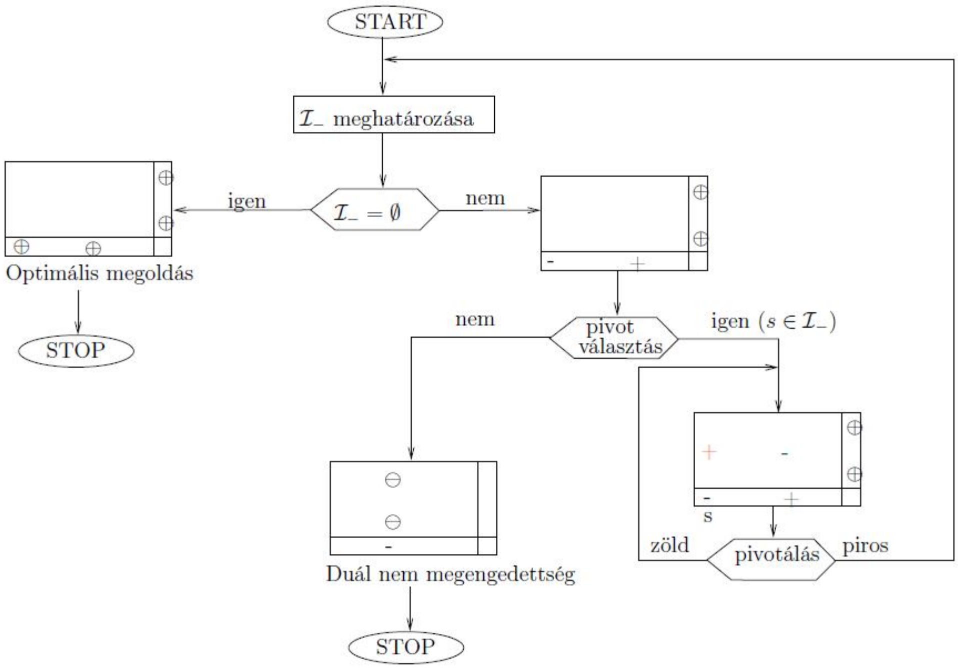
1. Legyen $\mathcal{I}_- := \{i \in \mathcal{I}_N \,|\, \bar{c}_i < 0\}$. Ha $\mathcal{I}_- = \emptyset$, akkor készen vagyunk, mert az A_B duál megengedett bázis is egyben. (Teljesülnek az optimalitási kritérium feltételei, azaz előállítottunk egy optimális megoldást, optimális bázis táblát.) Különben menjünk a 2. lépésre.

2. Legyen $q \in \mathcal{I}_-$ tetszőleges és határozzuk meg a $\mathcal{J}_q := \{i \in \mathcal{I}_B : t_{iq} > 0\}$ index halmazt. Ha $\mathcal{J}_q = \emptyset$ akkor készen vagyunk, mert teljesül a duál nem megengedettségi kritérium, $\mathcal{D} = \emptyset$. (Ebben az esetben nincsen optimális megoldása a feladatnak.) Különben menjünk a 3. lépésre.

3. Ha $\mathcal{J}_q \neq \emptyset$ akkor alkalmazzuk a hányados-tesztet:

$$\vartheta := \min\{\frac{b_i}{t_{iq}} : i \in \mathcal{J}_q\}$$

   és legyen $p \in \mathcal{I}_B$ tetszőleges, amelyre $\frac{b_p}{t_{pq}} = \vartheta$. Pivotáljunk t_{pq}-n és térjünk vissza az 1. lépéshez.



5.1. ábra. Szimplex algoritmus.

Adjuk meg a szimplex algoritmus folyamatábráját, amelyben a döntési helyzeteket pivot táblákon magyarázzuk.

Majd illusztráljuk a működését egy példán.

5.4. Példa. Induljunk ki a következő rövid bázistáblából:

	x_2	x_3	x_4	x_5	x_8	x_9	b
x_7	-4	1	-2	4	-2	6	4
x_6	-3	3	1	4	-4	11	12
x_1	1	-2	2	1	0	-2	1
	22	-6	9	-18	13	-35	-20

A szimplex módszerrel a duál nem megengedett változók közül bármelyik kiválasztható.

1. Amennyiben az x_3 változót választjuk belépőnek akkor a hányados teszt alapján

$$\frac{x_7}{t_{12}} = \frac{\bar{b}_1}{t_{12}} = \frac{4}{1} = 4 \quad \text{és} \quad \frac{x_6}{t_{22}} = \frac{\bar{b}_2}{t_{22}} = \frac{12}{3} = 4$$

adódik, azaz x_7 illetve x_6 is választható lenne távozónak.

2. Az x_5 változót választva belépő változónak akkor a hányados teszt alapján

$$\frac{x_7}{t_{14}} = \frac{\bar{b}_1}{t_{14}} = \frac{4}{4} = 1, \quad \frac{x_6}{t_{24}} = \frac{\bar{b}_2}{t_{24}} = \frac{12}{4} = 3 \quad \text{és} \quad \frac{x_1}{t_{34}} = \frac{\bar{b}_3}{t_{34}} = \frac{1}{1} = 1$$

két változó közül x_7 és x_1 választhatunk belépő változót.

3. Belépőnek választhatjuk az x_9 változót is. Ennek a változónak az oszlopára alkalmazva a hányados tesztet

$$\frac{x_7}{t_{16}} = \frac{\bar{b}_1}{t_{16}} = \frac{4}{6} = \frac{2}{3} \quad \text{és} \quad \frac{x_6}{t_{26}} = \frac{\bar{b}_2}{t_{26}} = \frac{12}{11}$$

a távozó változót, x_7, egyértelműen választhatjuk ki.

Válasszuk az első pivot pozíciónak t_{13}-at. Ekkor az alábbi (rövid) pivottáblát kapjuk:

	x_2	x_7	x_4	x_5	x_8	x_9	b
x_3	-4	1	-2	4	-2	6	4
x_6	9	-3	7	-8	2	-7	0
x_1	-7	2	-2	9	-4	10	9
	-2	6	-3	6	1	1	4

A többi pivot lépést a pivot pozícióval, a bázis változók bázisban elfoglalt sorrendjével, a primál megengedett megoldással és a célfüggvény értékével jellemezzük, ahelyett, hogy a teljes pivot táblákat közölnénk.

#	bázis vál.	bázis megoldás	célfv. érték	piv. poz.	megjegyegyzés
1.	x_7, x_6, x_1	$(1,0,0,0,0,12,4,0,0)$	$\zeta_1 = 20$	t_{13}	
2.	x_3, x_6, x_1	$(9,0,4,0,0,0,0,0,0)$	$\zeta_2 = -4$	t_{26}	degenerált
3.	x_3, x_2, x_1	$(9,0,4,0,0,0,0,0,0)$	$\zeta_3 = -4$	t_{42}	degenerált
4.	x_3, x_4, x_1	$(9,0,4,0,0,0,0,0,0)$	$\zeta_4 = -4$	t_{93}	
5.	x_9, x_4, x_1	$(1,0,0,1,0,0,0,0,1)$	$\zeta_5 = -6$		optimális

Végezetül, adjuk meg a szimplex algoritmus pszeudokódját:

Primál szimplex algoritmus (Dantzig, 1947)

Bemenő adatok:
a (P) feladat A_B primál megengedett bázisához tartozó T_B (rövid) bázis tábla.

Begin
$\mathcal{I}_- := \{i \in \mathcal{I}_N \mid \bar{c}_i < 0\}$;
while $\mathcal{I}_- \neq \emptyset$ **do**
begin
legyen $q \in \mathcal{I}_-$ tetszőleges;
if $\mathbf{t}_q \leq \mathbf{0}$ **then** *STOP:* a $\mathcal{D} = \emptyset$ (*duál nem megengedettségi kritérium*)
else
begin
legyen $\vartheta := \min\{\frac{x_i}{t_{iq}} : i \in \mathcal{I}_B \text{ és } t_{iq} > 0\}$; (*hányados teszt*)
legyen $p \in \mathcal{I}_B$ tetszőleges, amelyre $\frac{x_p}{t_{pq}} = \vartheta$;
end
endif
pivotálás: $\mathcal{I}_B := \mathcal{I}_B \cup \{q\} \setminus \{p\}$;
az $\mathcal{I}_-$ index halmaz meghatározása;
end
endwhile
$\mathcal{I}_- = \emptyset$ akkor optimális megoldásnál vagyunk; (*optimalitási kritérium*)
end.

5.2. Módszerek a ciklizálás elkerülésére

A primál szimplex algoritmusban – láthattuk az előző példán is – több helyen is volt választási szabadságunk. Előfordulhat olyan példa, amely megoldásakor a pivot pozíciók rossz megválasztása esetén, végtelen ciklusba kerülünk, azaz a bázisoknak egy véges sorozata, végtelen sokszor visszatér. Ezekben az esetekben, amikor az algoritmusunk ciklizál, nem tudjuk eldönteni, hogy az adott lineáris programozási feladatnak van-e megoldása vagy sincs. Tekintsük az alábbi, A. W. Tuckertől származó példát:

5.5. Példa. (A. W. Tucker) Legyen adott az alábbi rövid bázis tábla. Tekintettel arra, hogy az x_1 és x_2 bázis változók 0 értéket vesznek fel, az adott bázis tábla (primál) degenerált és az adott lineáris programozási feladat is degenerált.

1.	x_1	x_2	x_3	x_4	x_5	x_6	x_7	b
x_1	1	0	0	-2	-9	1	9	0
x_2	0	1	0	1/3	**1**	-1/3	-2	0
x_3	0	0	1	1	1	1	1	1
c	0	0	0	-2	-3	1	12	0

Az első bázis táblán, a pivot oszlopot, a legnegatívabb redukált költségű változó segítségével választjuk ki, így az x_5 lép be a bázisba és a hányados–teszt felhasználásával az x_2 változót választottuk ki távozónak. A pivot pozíció a t_{25} lesz és degenerált pivotra kerül sor, mert a $\bar{b}_2 = 0$. A degenerált pivot miatt a jobb oldali vektor értékei nem változtak meg, de a bázis és nem bázis vektorok halmaza megváltozott. A két különböző bázis, mégis ugyanazt a primál megengedett megoldást adja, vagyis annak ellenére, hogy a bázis megváltozott, a poliéderen kijelölt extremális pont, ugyanaz maradt. Így, természetesen, a célfüggvény értéke sem változott meg.

2.	x_1	x_2	x_3	x_4	x_5	x_6	x_7	b
x_1	1	9	0	**1**	0	-2	-9	0
x_5	0	1	0	1/3	1	-1/3	-2	0
x_3	0	-1	1	2/3	0	4/3	3	1
c	0	3	0	-1	0	0	6	0

A 2. primál megengedett bázis táblán, már csak egy duál nem megengedett változó található, az x_4, hiszen $\bar{c}_4 = -1$, ezért a pivot oszlop kiválasztása egyértelmű. A bázisba belépő változó az x_4 lesz és a hányados-teszt alkalmazásával választunk távozó változót. A hányados-teszt alkalmazásával 0, 0, és $\frac{3}{2}$ adódik, így a lehetséges távozó bázis változók, az x_1 és az x_5. Figyelembe véve a pivot pozíciók értékét: $t_{14} = 1$ és $t_{24} = \frac{1}{3}$ racionális döntésnek tűnik a t_{14} pozíciót választani pivot pozíciónak, különösen azért, mert a t_{14} értéke 1. Elvégezve a pivotálást az x_4 változó belép a bázisba a távozó x_1 változó helyére. Tekintettel arra, hogy degenerált pivotra került sor a jobboldal vektor nem változik és így a megoldás sem.

3.	x_1	x_2	x_3	x_4	x_5	x_6	x_7	b
x_4	1	9	0	1	0	-2	-9	0
x_5	-1/3	-2	0	0	1	1/3	**1**	0
x_3	-2/3	-7	1	0	0	8/3	9	1
c	1	12	0	0	0	-2	-3	0

A 3. bázis táblán két duál nem megengedett változó van, az x_6 és az x_7. Mivel az x_7 redukált költsége negatívabb, mint az x_6 változóé, ezért az x_7 változó legyen a bázisba belépő változó. A hányados-teszt ebben az esetben, a 7. oszlopra alkalmazva egyértelműen kijelöli a pivot pozíciót, amelyik a t_{27} lesz, azaz a második bázis változó távouik, ez pedig az x_5. Ismét degenerált pivotra került sor.

4.	x_1	x_2	x_3	x_4	x_5	x_6	x_7	b
x_4	-2	-9	0	1	9	**1**	0	0
x_7	-1/3	-2	0	0	1	1/3	1	0
x_3	7/3	11	1	0	-9	-1/3	0	1
c	0	6	0	0	3	-1	0	0

A 4. primál megengedett bázis táblán egyértelmű a pivot oszlop kiválasztása, mivel egyetlen nem bázis változó redukált költsége negatív, azaz az x_6 változó lép be a bázisba. A hányados-teszt két lehetséges pivot pozíciót jelöl ki, a t_{16} és t_{26} elemeket. Az eddigi döntéseinkkel összhangban, a törtekkel való számolás minimalizálásának az érdekében a $t_{16} = 1$ elemet választjuk pivot elemnek és ebben az esetben a bázisban az első pozíciót elfoglaló x_4 változó távozik. Ez alkalommal is degenerált pivotra került sor.

5.	x_1	x_2	x_3	x_4	x_5	x_6	x_7	b
x_6	-2	-9	0	1	9	1	0	0
x_7	1/3	**1**	0	-1/3	-2	0	1	0
x_3	5/3	8	1	1/3	-6	0	0	1
c	-2	-3	0	1	12	0	0	0

Az 5. primál megengedett bázis táblán ismét két lehetséges belépő változó van, hiszen az x_1 és x_2 változók redukált költségei rendre $\bar{c}_1 = -2$ és $\bar{c}_2 = -3$. Ismét mohó módon és lokálisan döntünk, azaz a negatívabb árnyékárú változót, x_2 jelöljük ki belépő változónak, majd oszlopára alkalmazzuk a hányados-tesztet. A hányados-teszt egyértelműen jelöli ki a pivot pozíciót, $t_{22} = 1$ és így a bázis második változója, x_7 távozik a bázisból. Az x_2 és x_7 változók közötti bázis csere is egy degenerált pivottal valósult meg.

6.	x_1	x_2	x_3	x_4	x_5	x_6	x_7	b
x_6	**1**	0	0	-2	-9	1	9	0
x_2	1/3	1	0	-1/3	-2	0	1	0
x_3	-1	0	1	3	10	0	-8	1
c	-1	0	0	0	6	0	3	0

A 6. primál megengedett bázis táblán – ismét – egyértelmű a pivot oszlop kiválasztása, mivel egyetlen nem bázis változó redukált költsége negatív, azaz az x_1 változó lép be a bázisba. A hányados-teszt két lehetséges pivot pozíciót jelöl ki, a t_{16} és t_{26} elemeket. Az eddigi döntéseinkkel összhangban, a törtekkel való számolás minimalizálásának az érdekében a $t_{16} = 1$ elemet választjuk pivot elemnek és ebben az esetben a bázisban az első pozíciót elfoglaló x_6 változó távozik. Ez alkalommal is degenerált pivotra került sor.

Ebben az esetben visszakapjuk az 1. primál megengedett bázis táblát.

A következőképpen alakultak a bázis index halmazok:

$\mathcal{I}_{B_1} = \{1,2,3\}$, $\mathcal{I}_{B_2} = \{1,3,5\}$, $\mathcal{I}_{B_3} = \{3,4,5\}$, $\mathcal{I}_{B_4} = \{3,4,7\}$, $\mathcal{I}_{B_5} = \{3,6,7\}$, $\mathcal{I}_{B_6} = \{2,3,6\}$, és $\mathcal{I}_{B_7} = \{1,2,3\} = \mathcal{I}_{B_1}$.

Tehát a B_1, B_2, B_3, B_4, B_5, B_6, B_1 bázis sorozat ciklust alkot, vagyis a szimplex módszer a degenerált feladat esetén ciklizálhat, így nem feltétlenül véges.

Az 5.5. Példán bemutatott jelenséget szeretnénk megvizsgálni, ezért tekintsük a 4.1. Definícióban megadott (P), primál lineáris programozási feladatot és tegyük fel, hogy a feladat, kanonikus feladat, azaz ismert egy A_B primál megengedett bázisa.

5.6. Definíció. Legyen adott a (P) lineáris programozási feladat és az A_B primál megengedett bázisa. Az A_B primál megengedett bázis, pontosan akkor *nem degenerált bázis*, ha $x_B = A_B^{-1}\,\mathbf{b} = \bar{\mathbf{b}} > \mathbf{0}$ teljesül.

Egy bázist *degenerált*nak nevezünk, ha létezik $i \in \mathcal{I}_B : \bar{b}_i = 0$. Észrevehetjük, hogy Tucker feladata esetén, a ciklizálásra, degenerált lineáris programozás feladat esetén került sor.

A (P) lineáris programozási feladatot *teljesen (primál) nem degenerált*nak nevezzük, ha a feladat összes bázisa nem degenerált.

5.7. Feladat. Legyen adott a (P) lineáris programozási feladat. Bizonyítsa be, hogy a (P) pontosan akkor teljesen (primál) nem degenerált lineáris programozási feladat, ha bármely $\mathbf{x} \in \mathcal{P}$ megoldás esetén legalább m nem nulla eleme van.

Legyen adott a (P) lineáris programozási feladat. Ha degenerált a (P) feladat, akkor a jobb oldalát módosíthatjuk, úgy, hogy

$$b_i(\varepsilon) = b_i + \varepsilon^i,$$

ahol $\varepsilon > 0$.

5.8. Feladat. Legyen adott az $A \in \mathbb{R}^{m\times n}$ mátrix és a $\mathbf{b} \in \mathbb{R}^m$ vektor. Tegyük fel, hogy a (3.1) lineáris egyenlőtlenségrendszer degenerált, ekkor létezik $\varepsilon_1 > 0$ úgy, hogy bármely $0 < \varepsilon < \varepsilon_1$ esetén az

$$A\,\mathbf{x} = \mathbf{b}(\varepsilon), \quad \mathbf{x} \geq \mathbf{0}$$

lineáris egyenlőtlenségrendszer nem degenerált.

Az előző feladat mutatja, hogyha a (P) feladat degenerált, akkor a jobb oldal perturbálásával – elvileg – elérhető lenne az, hogy a módosított jobb oldal mellett a feladat teljesen (primál) nem degenerált legyen.

Erre a perturbálási módszerre hivatkozva, a lineáris programozási feladatot megoldó professzionális optimalizálási szoftverek gyártói azt állítják, hogy a gyakorlati feladatok esetén, amelyek alapvetően primál (és duál) degenerált feladatok, szoftvereik mégsem ciklizálnak. A gyakorlati tapasztalat azt mutatja, hogy professzionális optimalizálási szoftverekkel történő számolásoknál, ritkán találkoznak ciklizálással.

Elméletileg persze kérdés, az, hogy hogyan kell meghatározni az ε_1 értéket. Gyakorlatban viszont az a gond, hogyha $\varepsilon_1 = 10^{-8}$ akkor $\varepsilon_1^{10} = 10^{-80}$, azaz $m \geq 10$ esetén, a $\mathbf{b}$ vektor 9. koordinátájának a perturbációja már nem is adható meg lebegőpontos számábrázolással, vagyis a gyakorlatban az említett perturbáció nem is fordulhat elő. Tehát ez a perturbációs modell nem ad magyarázatot arra a (tapasztalati) állításra, hogy a professzionális szoftverek nem (vagy inkább ritkán) ciklizálnak gyakorlati feladatok esetén.

A szimplex algoritmus ciklizálásának a kérdése, az 1950-es évek egyik sokat kutatott témaköre volt. Az elmélet szempontjából, un. *ciklizálást megakadályozó* pivotálási szabályokat kerestek. Az 1950-es években ilyen szabályokat nem sikerült felfedezniük, ehelyett Dantzig, Orden és Wolfe, a lexikografikus szimplex módszer megfogalmazásával adtak meg egy olyan szimplex algoritmus variánst, amely esetén a végességet sikerült igazolniuk degenerált feladtok esetén is. A lexikografikus szimplex módszer végességének a bizonyítása bonyolult és olyan technikákat használ (lexikografikus sorrend, lexikografikus tábla), amelyek idegenek a lineáris programozási feladatok témakörében illetve gyakorlati megvalósításuk is nagyon nehézkes.

Bland 1977-ben közölte, híressé vált, nagyon egyszerű ciklizálás ellenes szabályát, a minimál index szabályt, amelynek segítségével, egyszerű bizonyítást adott a szimplex módszer végességére. A szimplex módszernek a következő egyszerű módosítására kényszerült:

1. Ha a redukált költségek között több negatív van, akkor az legyen a pivot oszlop, amelyikhez tartozó változó indexe a legkisebb.

2. Ha a hányados-teszt több lehetséges távozó változót jelölne ki, akkor azok közül a legkisebb indexű változót kell távozónak választani.

Korábban a criss-cross és az MBU-szimplex algoritmusok megengedettségi feladatra definiált változatainak a végességét is a Bland-féle minimál indexes szabály alkalmazásával bizonyítottuk be. A 4. fejezetben a lineáris programozási feladat megoldására megfogalmazott criss-cross algoritmus végességét is a minimál index szabály használata biztosította.

A mi felépítésünk mellett, Bland 1977-es híres eredménye, egy egyszerű feladattá válik. (Fogalmazzuk meg a majdnem leállási táblákat, amelyeket a minimál index szabály logikája definiál. Ciklizálás feltételezése esetén, ezeknek a majdnem leállási tábláknak egyidejűleg elő kell fordulniuk az algoritmus által meglátogatott bázis táblák között, ellentmondva az ortogonalitási tételnek.)

5.9. Feladat. Legyen adott a (P) lineáris programozási feladat. Bizonyítsa be, hogy a Dantzig-féle (primál) szimplex algoritmus, ha a pivot pozíció meghatározásakor, a Bland-féle minimál index szabályt alkalmazzuk, akkor degenerált feladat esetén is véges.

A ciklizálást megakadályozó pivotálási szabályok vizsgálatának újabb korszakát S. Zhang indította el 1999-es dolgozatával, amelyben két olyan szabályról, a LIFO és a MOSV szabályokról, – amelyeket kombinatorikus optimalizálási algoritmusokban előszeretettel használtak, – mutatta meg, hogy a criss-cross algoritmus esetén, mint ciklizálást megakadályozó pivotálási szabályok, biztosítják a végességét. Zhang bizonyításai a kelleténél bonyolultabbak voltak. Illés és Mészáros az 1.48. Tétel felhasználásával és a majdnem leállási táblák definiálásával igazolták a megengedettségi- és a lineáris programozási feladatokra megfogalmazott criss-cross algoritmusok végességét felhasználva a LIFO és MOSV index választási szabályokat, sőt a 3.18. Lemma és a 4.11. Tétel bizonyításai a jegyzetünkben közöltekhez nagyon hasonlóak.

Vezessük be a LIFO és a MOSV index választási szabályokat. Legyen adott a (P) lineáris programozási feladat és jelölje $\mathcal{I}$ a változók indexeinek a halmazát. Tekintsük a következő leképezéseket: $\mathbf{s}_r : \mathcal{I} \mapsto \mathbb{N}_0$, és legyen $\mathbf{s}_0^T = (0, 0, ..., 0)$ vektor. Az $\mathbf{s}$ vektort számláló vektornak hívjuk és ez követi, adminisztrálja a pivot pozíciókat a következő módon:

Leggyakrabban választott változó (MOSV):

$$s_r(i) = \begin{cases} s_{r-1}(i) + 1, & \text{ha az } i. \text{ változó mozog az } r. \text{ iterációban} \\ s_{r-1}(i), & \text{különben.} \end{cases}$$

A $\mathbf{s}_{r-1}$ vektorban tartjuk nyilván, hogy az r. iterációig, melyik változó, hányszor változtatott bázis állapotot, azaz hányszor lépett be illetve távozott a bázisból.

LIFO:

$$s_r(i) = \begin{cases} r, & \text{ha az } i. \text{ változó mozog az } r. \text{ iterációban} \\ s_{r-1}(i), & \text{különben.} \end{cases}$$

A $\mathbf{s}_{r-1}$ vektorban tartjuk nyilván, hogy az r. iterációig, melyik változó, mikor változtatott bázis állapotot utoljára.

Az $\mathbf{s}_{r-1}$ számláló vektor segítségével definiálhatjuk a ciklizálást megakadályozó index választási szabályokat:

- **LIFO (Last In First Out).** Belépésnél, az r. iterációban, válasszuk azt az x_k változót a lehetségesek közül, amelyik utoljára került ki a bázisból, azaz

  $$\mathbf{s}_{r-1}(k) \geq \mathbf{s}_{r-1}(i), \text{ teljesül } \forall\, i \in \mathcal{I}_N.$$

  Kilépéskor válasszuk azt az x_l változót a lehetségesek közül, amelyik utoljára került be a bázisba, azaz

  $$\mathbf{s}_{r-1}(l) \geq \mathbf{s}_{r-1}(i), \text{ teljesül } \forall\, i \in \mathcal{I}_B.$$

- **Leggyakrabban választott változó (MOSV).** Mindig a leggyakrabban választott változót választjuk, azaz az r. iterációban belépőnek válasszuk azt az x_k változót a lehetségesek közül, amelyik esetén

  $$\mathbf{s}_{r-1}(k) \geq \mathbf{s}_{r-1}(i), \text{ teljesül } \forall\, i \in \mathcal{I}_N.$$

  A kilépő x_l változót, a lehetségesek közül, úgy választjuk ki, hogy

  $$\mathbf{s}_{r-1}(l) \geq \mathbf{s}_{r-1}(i), \text{ teljesül } \forall\, i \in \mathcal{I}_B.$$

Időnként, például az iterációs sorozat kezdetén, előfordulhat az az eset, hogy sem a LIFO, sem pedig a MOSV nem jelöli ki egyértelműen a belépő illetve a távozó változókat. Ebben az esetben a maximális értékű számlálóval rendelkező változók közül tetszőlegesen választhatunk. Ez a választási szabadság – látszólag – az eredeti Dantzig-féle szimplex algoritmus pivot pozíció megválasztásának a szabadágához hasonló, és ezért ciklizálást eredményezhet. Ez a szabadság miatt, pivot algoritmusok végesség bizonyítása a minimál indexes szabálynál kicsit bonyolultabb gondolatmenetet kíván.

A következő állítások bizonyításai megtalálhatók Csizmadia Zsolt doktori (PhD) disszertációjában.

5.10. Feladat. Legyen adott a (P) lineáris programozási feladat. Bizonyítsa be, hogy a Dantzig-féle (primál) szimplex algoritmus degenerált feladatok esetén is véges, ha a pivot pozíció meghatározásakor, a

1. LIFO,
2. MOSV

ciklizálást megakadályozó pivotálási szabályt használja.

Az előzőhöz hasonló állítás fogalmazható meg a lineáris programozási feladat megoldására definiált Terlaky-féle criss-cross algoritmus esetén is, azzal, hogy a minimál indexes szabályt lecseréljük a LIFO illetve a MOSV ciklizálást megakadályozó szabályra.

5.11. Feladat. Legyen adott a (P) lineáris programozási feladat. Bizonyítsa be, hogy a criss-cross algoritmus degenerált feladatok esetén is véges, ha a pivot pozíció meghatározásakor, a

1. LIFO,
2. MOSV

ciklizálást megakadályozó pivotálási szabályt használja.

Tekintettel arra, hogy a minimál indexes szabály használata, numerikus feladatok esetén, a kötöttsége miatt, nem kecsegtet nagyon nagy sikerrel, érdekes kérdés a kevésbé kötött, ciklizálás megakadályozására szolgáló szabályok gyakorlati feladatok megoldásakor való viselkedésének a vizsgálata. A LIFO és MOSV szabályok gyakorlati használhatóságát, (primál) szimplex– és (primál) MBU–szimplex algoritmusok esetén Illés Tibor és Nagy Adrienn vizsgálták.

Elméleti szempontból érdekes, hogy hány ciklizálás ellenes szabály lehet. Csizmadia, Illés és Nagy, megadták a Bland-féle minimál index, a LIFO és MOSV ciklizálás ellenes szabályok közös általánosítását, az un. *s-monoton index választási szabályok*at és megmutatták, hogy ez a ciklizálás ellenes szabály osztály, végtelen sok elemet tartalmaz.

Igazolták továbbá, hogy a szimplex–, MBU–szimplex– és criss-cross algoritmusok mindegyike véges, tetszőleges s-monoton index választási szabály esetén, egységesítve a pivot algoritmusok végesség bizonyításait.

5.3. Kétfázisú szimplex algoritmus

Említettük, hogy a primál szimplex algoritmus elindításához szükségünk van egy kezdeti primál megengedett bázisra. Ha kezdetben nem rendelkezünk ilyennel, akkor használhatjuk a kétfázisú szimplex algoritmust, amelynek első fázisa egy primál megengedett bázist keres.

Legyen adott a következő lineáris programozási feladat:

$$\left.\begin{array}{rcl} \min & \mathbf{c}^T\mathbf{x} & \\ A\mathbf{x} & = & \mathbf{b} \\ \mathbf{x} & \geq & \mathbf{0} \end{array}\right\} \quad (P)$$

ahol $A \in \mathbb{R}^{m\times n}$, $\mathbf{c}, \mathbf{x} \in \mathbb{R}^n$ és $\mathbf{y}, \mathbf{b} \in \mathbb{R}^m$. Az általánosság korlátozása nélkül feltehető, hogy $rang(A) = m$ és $\mathbf{b} \geq \mathbf{0}$. Ha az A mátrix tartalmaz egy $m \times m$-es egységmátrixot részmátrixként, akkor a (P) feladatot *kanonikus feladat*nak nevezzük és kiolvashatunk egy induló, megengedett bázis megoldást és a hozzá tartozó $\mathcal{I}_B$ és $\mathcal{I}_N$ index halmazokat.

Ha a (P) feladat nem kanonikus, akkor *mesterséges változó*k, $\mathbf{u} \in \mathbb{R}^m$ bevezetésével kanonikus feladattá transzformálhatjuk:

$$\left.\begin{array}{rc} \min & \mathbf{0}^T\mathbf{x} + \mathbf{e}^T\mathbf{u} \\ & A\mathbf{x} + I\mathbf{u} = \mathbf{b} \\ & \mathbf{x},\, \mathbf{u} \geq \mathbf{0} \end{array}\right\} \quad (P_1)$$

Mivel a (P_1) kanonikus feladat, van induló megengedett bázis megoldása és a célfüggvénye alulról korlátos, ezért megoldható szimplex módszerrel. A mesterséges változók egy primál megengedett bázisát alkotják a (P_1) feladatnak, mivel

$$(\mathbf{x}, \mathbf{u}) = (\mathbf{0}, \mathbf{b}) \geq (\mathbf{0}, \mathbf{0}).$$

Ekkor az eredeti (P) lineáris programozási feladatnak nem kapjuk meg egy megoldását, kivéve, ha $\mathbf{b} = \mathbf{0}$. Továbbá, nyilvánvaló, hogy a (P_1) feladat célfüggvény értéke, minden megoldásra nem negatív lesz.

Könnyen igazolható a következő állítás.

5.12. Feladat. Legyen adottak a (P) és (P_1) lineáris programozási feladatok. A $P \neq \emptyset$ pontosan akkor igaz, ha a (P_1) feladat optimumértéke nulla.

Ezek után megfogalmazhatjuk a kétfázisú szimplex módszert, általános lineáris programozási feladat megoldására.

Kétfázisú szimplex módszer:

- Vizsgáljuk meg a (P) feladatot. Ha kanonikus a feladat, akkor oldjuk meg a szimplex algoritmussal. **STOP**

- *Első fázis:* Különben írjuk fel a (P_1) feladatot. A (P_1) kanonikus feladat. Oldjuk meg a primál szimplex módszerrel.

  Ha a (P_1) feladat optimumértéke pozitív akkor nincsen megengedett megoldása a (P) feladatnak. **Primál nem megengedettség. STOP**

  Különben menjünk a második fázisra.

- *Második fázis:* Mivel a (P_1) feladat optimumértéke nulla, akkor megkaptuk a (P) feladatnak egy megenegedett bázisát (Kanonikus alakra transzformáltuk a feladatot).

  Oldjuk meg a (P) feladatot primál szimplex módszerrel, az első fázisban előállított bázisról indulva. **STOP**

5.4. MBU-szimplex algoritmus

Az MBU-szimplex algoritmust Anstreicher és Terlaky fogalmazták meg 1991-ben. A 3. fejezetben az MBU-szimplex algoritmusnak egy egyszerűsített duál változatát használtuk fel arra, hogy tetszőleges bázis megoldásból indulva megoldjuk a lineáris megengedettségi feladatot (lineáris egyenlőtlenségrendszert), vagy kimutassuk azt, hogy nincsen megoldása. Azt az MBU-szimplex algoritmust részletesen elemeztük. Most bemutatjuk az eredeti, Anstreicher és Terlaky által megfogalmazott (primál) MBU-szimplex algoritmust.

MBU–szimplex algoritmus (Anstreicher–Terlaky, 1991): Legyen adott a (P) lineáris programozási feladata és egy primál megengedett bázisa, A_B, a hozzá tartozó bázis táblával. Jelölje $\bar{\mathbf{c}}$ a redukált költségeket.

1. Ha az A_B bázis duál megengedett is, akkor készen vagyunk: találtunk egy **optimális megoldást. STOP**

   Különben valamelyik ciklizálás ellenes pivotálási szabály segítségével (minimál index, LIFO vagy MOSV), válasszuk ki az x_s duál nem megengedett változót (azaz $\bar{c}_s < 0$) és legyen x_s az un. *vezér változó*.

2. Ha az x_s változó oszlopában, minden elem nem pozitív, akkor $D = \emptyset$. **Duál nem megengedettségi kritérium. STOP**

   Különben a hányados-teszt alkalmazásával határozzuk meg a távozó változót $x_r,\ r \in \mathcal{I}_B$. (Amennyiben a hányados több helyen is felvétetett, akkor a ciklizálás elkerülésének az érdekében, használjunk ciklizálás ellenes pivotálási szabályt.)

3. Vezessük be a következő index halmazt

$$\mathcal{J}_r = \{i \in \mathcal{I} \mid \bar{c}_i \geq 0 \;\text{ és }\; t_{ri} < 0\}.$$

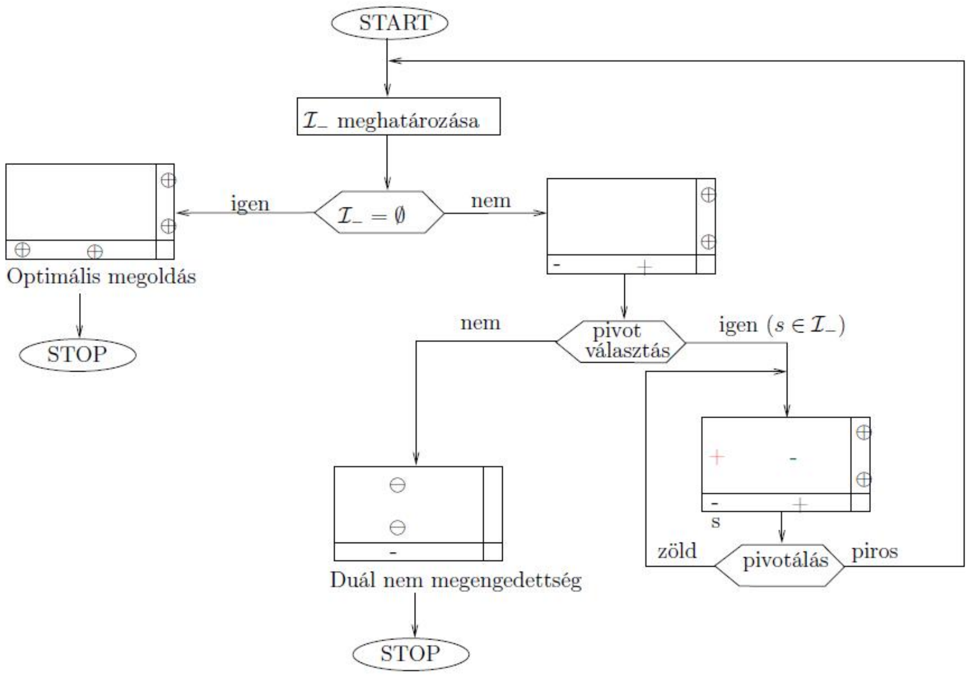
Számítsuk ki a Θ_1 és Θ_2 paraméterek értékét

$$\Theta_1 := \frac{|\bar{c}_s|}{t_{rs}} \qquad \text{és} \qquad \Theta_2 := \min_{i \in \mathcal{J}_r} \frac{\bar{c}_i}{|t_{ri}|} = \frac{\bar{c}_q}{|t_{rq}|}.$$

Ha a Θ_2 paraméter értékének a kiszámításakor a minimum több helyen is felvétetik, akkor alkalmazzuk a ciklizálás ellenes pivotálási szabályt, a $q \in \mathcal{I}_N \cap \mathcal{J}_r$ index kiválasztásakor.

4. Ha $\Theta_1 \leq \Theta_2$, akkor pivotáljunk t_{rs} elemen, és térjünk vissza az **1.** lépéshez (új fázis kezdődik).

5. Ha $\Theta_1 > \Theta_2$, akkor pivotáljunk t_{rq} elemen, ahol $q \in \mathcal{I}_N \cap \mathcal{J}_r$. Térjünk vissza a **2.** lépéshez.



5.2. ábra. MBU-szimplex algoritmus.

Legyen adott a (P) lineáris programozási feladat egy induló primál megengedett bázissal. Jelölje a szokásos módon $\mathcal{I}_B$ a bázis változók, míg $\mathcal{I}_N$ a bázison kívüli változók index halmazait. Legyen

$$\mathcal{I}_N^{\oplus} := \{j \in \mathcal{I}_N : \bar{c}_j \geq 0\} \qquad \text{és} \qquad \mathcal{I}_N^{-} := \{j \in \mathcal{I}_N : \bar{c}_j \geq 0\},$$

a megengedett illetve nem megengedett duál változók index halmaza.

A szokásos módon megadjuk az MBU-szimplex algoritmus pszeudokódját is.

MBU-szimplex algoritmus lineáris programozási feladatokra

Bemenő adatok: a *kanonikus* (P) feladat A_B *primál megengedett* bázisához tartozó T_B (rövid) bázis tábla.

Begin
$\mathcal{I}_- := \{i \in \mathcal{I}_N \,|\, \bar{c}_i < 0\}$;
while $\mathcal{I}_- \neq \emptyset$ **do**
vezető változó: legyen $s \in \mathcal{I}_-$ tetszőleges;
while $s \in \mathcal{I}_-$ **do**
legyen $\mathcal{K}_s = \{i \in \mathcal{I}_B : t_{is} > 0\}$
if $\mathcal{K}_s = \emptyset$ **then** *STOP:* a $\mathcal{D} = \emptyset$ (*duál nem megengedettségi kritérium*)
else
legyen $\vartheta := \min\{\frac{x_i}{t_{is}} : i \in \mathcal{K}_s\}$; (*hányados-teszt*)
legyen $r \in \mathcal{I}_B$ tetszőleges, amelyre $\frac{x_r}{t_{rs}} = \vartheta$ és $\Theta_1 := \frac{|\bar{c}_s|}{t_{rs}}$;
legyen $\mathcal{J}_r = \{i \in \mathcal{I} \mid \bar{c}_i \geq 0$ és $t_{ri} < 0\}$;
if $\mathcal{J}_r = \emptyset$ **then** $\Theta_2 := +\infty$
else $\Theta_2 := \min_{i \in \mathcal{J}_r} \frac{\bar{c}_i}{|t_{ri}|}$ és $q \in \mathcal{I}_N : \Theta_2 = \frac{\bar{c}_q}{|t_{rq}|}$
endif
if $\Theta_1 \leq \Theta_2$ **then** pivotálás a t_{rs} elemen
else pivotálás a t_{rq} elemen
endif
határozzuk meg az $\mathcal{I}_-$ halmazt;
endif
endwhile
endwhile
$\mathcal{I}_- = \emptyset$ akkor optimális megoldásnál vagyunk; (*optimalitási kritérium*)
end.

Az MBU-szimplex algoritmus elemzéséhez, végességének a bizonyításához megfogalmazunk néhány feladatot.

5.13. Feladat. Legyen adott a (P) lineáris programozási feladat. Az MBU–szimplex algoritmus alkalmazásakor, ha $\Theta_1 \leq \Theta_2$ akkor a duál megengedett változók indexek halmaza szigorúan nő, és nem csökken, ha $\Theta_1 > \Theta_2$.

Ha $\Theta_1 > \Theta_2$ akkor csak azt tudjuk, hogy az $\mathcal{I}_N^{\oplus}$ index halmaz nem csökken. A végesség bizonyításához az kellene, hogy szigorúan nőjön, ezért szükséges azokat a pivotálásokat részletesebben elemezni, amikor $\Theta_1 > \Theta_2$ áll fenn.

5.14. Feladat. Legyen adott a (P) lineáris programozási feladat. Tekintsünk az MBU–szimplex algoritmus alkalmazásával, egy primál megengedett bázisból, egy x_s vezér változó segítségével előállított, tetszőleges pivotsorozatot. Ekkor a következő pivotálás, amelyre a $\Theta_1 > \Theta_2$ feltétel mellett kerül sor, a következő három tulajdonság teljesül:

(a) $\bar{c}_s < 0$,

(b) ha $\bar{b}_i < 0$ akkor $t_{is} < 0$, és

(c) $\max\left\{\frac{\bar{b}_i}{t_{is}} \mid \bar{b}_i < 0\right\} \leq \min\left\{\frac{\bar{b}_i}{t_{is}} \mid t_{is} > 0\right\}$.

Az MBU-szimplex algoritmus nagyon fontos tulajdonságát fogalmazza meg a következő feladat.

5.15. Feladat. Legyen adott a (P) lineáris programozási feladat. Az MBU–szimplex algoritmus alkalmazásakor a bázis primál megengedettsége helyreáll amikor a vezér változó belép a bázisba.

Az előző feladat egyben azt is jelzi, hogy amikor $\Theta_1 > \Theta_2$ feltétel mellett kerül sor pivotálásra és nem a vezér változó lép be a bázisba, akkor a következő bázis primál megengedettsége elromolhat, vagyis az MBU-szimplex algoritmus primál megengedett bázisból indulva, meglátogathat olyan bázisokat is, amelyek se nem primál, se nem duál megengedett bázisok.

5.16. Feladat. Legyen adott a (P) lineáris programozási feladat. Tegyük fel, hogy az MBU–szimplex algoritmus bármely pivot lépése nem degenerált, azaz $\min\{\Theta_1, \Theta_2\} > 0$. Ekkor az algoritmus véges sok lépésben a kővetkező két állapot valamelyikével ér véget:

(1) a (D) feladatnak nincsen megengedett megoldása, vagy

(2) az MBU–szimplex algoritmus optimális megoldást talál.

A nem degeneráltsági feltétel mellett több is igazolható, a megengedettségi feladathoz hasonlóan igazolható, hogy új vezér változó legfeljebb n alkalommal választható és egy-egy alkalommal legfeljebb $K \in \mathbb{N}$ iteráció lehetséges, mielőtt új vezér változó választására kerül sor. A K szám – hasonlóan a 3. fejezetben tárgyalT MBU-szimplex algoritmushoz – függ a feladat adataitól. Ez egyben azt is jelenti, hogy az algoritmus nem degeneráltsági feladatok esetén sem lesz polinomiális.

Degenerált (P) feladatok esetén, a ciklizálás elkerülése végett, ciklizálást akadályozó index választási szabályt (pl. minimál index, LIFO, MOSV) kell alkalmazni, azokban a helyzetekben, ahol a választás nem egyértelmű.

Az MBU-szimplex algoritmussal kapcsolatban felmerül egy érdekes kérdés. Megfogalmazható az MBU-szimplex algoritmusnak olyan variánsa, amelyik tetszőleges, se nem primál, se nem duál bázisról elindítható és véges ? Másképpen fogalmazva, egy se nem primál, se nem duál bázis esetén meg tudunk-e adni egy primál [vagy duál] vezér változót (duál [vagy primál] nem megengedett változót), úgy, hogy az véges sok lépésben megengedetté váljon és amikor megengedetté válik, akkor a bázis primál [illetve duál] megengedett lesz ?

A kérdés pozitív eldöntése esetén az MBU-szimplex algoritmushoz nem kellene első fázis, jelentősen megnövelve az alkalmazhatóságát.

5.5. Módosított szimplex algoritmus

Tárgyalásunk eddigi részében fontos szerepet játszott a pivot tábla. A pivot táblát, a feladat egy olyan reprezentációjának tartottuk, amelyen könnyen megfogalmazhatók a megállási kritériumok, a majdnem leállási táblák, a bázis táblák, pivot oszlop és pivot sor kiválasztásának a feltételei és egyéb az algoritmusok értelemezésében és elemzésében segítő fontos részletek. A pivot tábla segítséget nyújtott numerikus feladatok megoldásának a bemutatásában. A pivot tábla hasznossága ellenére, természetesen, a lineáris programozási pivot algoritmusok számítógépes implementációja során kevés szerep jut a pivot táblának, hiszen iterációról-iterációra nem számoljuk ki a pivot táblát. Megmutatható, a kompozíciós tulajdonság (1.49. Következmény) segítségével, hogy elegendő az eredeti adatainkat és az aktuális bázis inverzét ismerni és ezekből kiszámítható a jobb oldal transzformált vektora, a redukált költségek, amelyek segítségével a primál és duál emegengedettség eldönthető illetve a pivot sor (oszlop) elemei kiszámíthatók.

Felmerül a kérdés, hogy ha primál szimplex módszer esetén az aktuális bázis inverze, a jobb oldal transzformált vektora, a redukált költségek, a pivot oszlop és a pivot pozíció adottak, akkor hogyan határozható meg a következő primál megengedett bázis, bázis inverze ?

A kérdés megválaszolásához a szomszédos bázisokat kell először tanulmányoznunk.

5.17. Definíció. Legyen adott az $A = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n] \in \mathbb{R}^{m\times n}$ mátrix. Az A mátrix oszlop vektorai által alkotott két bázisát, B_1 és B_2 bázisokat, *szomszédos bázisok*nak nevezzük, ha mindössze egy vektorban térnek el egymástól.

Az egyik bázist jelölje B és az A mátrixnak azt az oszlop vektorát, amelyben a két bázis eltér, $\mathbf{a}$. Ekkor nyilván a B bázis volta miatt

$$\mathbf{a} = B\,\mathbf{v}, \qquad \text{azaz} \qquad \mathbf{a} = \sum_{j\in I_B} v_j\,\mathbf{a}_j$$

teljesül, valamely $\mathbf{v} \in \mathbb{R}^m$ vektor esetén. Ha a $v_p \neq 0$, akkor az $\mathbf{a}_p$ a következő módon fejezhető ki

$$\mathbf{a}_p = \frac{1}{v_p}\,\mathbf{a} - \sum_{j\in I_B\setminus\{p\}} \frac{v_j}{v_p}\,\mathbf{a}_j.$$

A B_a mátrix, az A mátrix következő oszlopaiból áll

$$\{\mathbf{a}_j : j \in I_B \text{ és } j \neq p\} \cup \{\mathbf{a}\}$$

és az A mátrix oszlop vektoraiból álló véges vektor rendszer bázisát alkotja.

Nyilván, a B_a pontosan akkor nem szinguláris mátrix, ha $v_p \neq 0$. Ekkor az előző egyenlet

$$\mathbf{a}_p = B_a\bar{\mathbf{v}} \qquad \text{alakban írható, ahol} \qquad \bar{\mathbf{v}}^T = \left[-\frac{v_1}{v_p}, \ldots, -\frac{v_{p-1}}{v_p}, \frac{1}{v_p}, -\frac{v_{p+1}}{v_p}, \ldots, -\frac{v_m}{v_p}\right]^T.$$

Definiáljuk a $P = [\mathbf{e}_1, \ldots, \mathbf{e}_{p-1}, \bar{\mathbf{v}}, \mathbf{e}_{p+1}, \ldots, \mathbf{e}_m]$ *pivot mátrix*ot. Ekkor

$$B = B_a\,P \qquad \text{és így} \qquad B^{-1} = P^{-1}B_a^{-1} \qquad \text{azaz} \qquad B_a^{-1} = PB^{-1}$$

Tehát egyik bázisról áttérni valamelyik szomszédosra, ha az első bázis az inverz mátrixával adott, akkor egy pivot mátrixszal balról való szorzással térhetünk át a szomszédos bázis inverzére.

Ezzel megadtuk azt, hogyan térhetünk át szomszédos bázisra. Ha a pivot pozíció (amelyik esetünkben a $v_p \neq 0$ elem volt), a hányados-teszt segítségével kerül kijelölésre egy pivot algoritmus során és a kiindulási bázisunk is primál megengedett bázis volt, akkor az új szomszédos bázis is primál megengedett bázis lesz.

Ennyi előkészítés után, megfogalmazhatjuk a primál szimplex algoritmus un. *módosított* változatát, amelyik esetén nem szükséges a teljes pivot táblát kiszámolni, hanem csak a következő információkra lesz szükségünk: eredeti adatok $A \in \mathbb{R}^{m\times n}$ mátrix, $\mathbf{b} \in \mathbb{R}^m$ jobb oldal vektor, $\mathbf{c} \in \mathbb{R}^n$ a célfüggvény együtthatóinak a vektora, az aktuális primál megengedett bázis inverze B^{-1}, a jobb oldal transzformált vektora

$\bar{\mathbf{b}} = B^{-1}\mathbf{b}$, a duál feladat (aktuális) megoldás vektora $\mathbf{y}^T = \mathbf{c}_B^T B^{-1}$, a redukált költségek $\bar{\mathbf{c}} = \mathbf{c} - \mathbf{y}^T A$ vektora. Határozzuk meg az

$$\mathcal{I}_- := \{i \in \mathcal{I}_N \,|\, \bar{c}_i < 0\}$$

halmazt.

Ezekből az adatokból kiindulva a módosított (primál) szimplex algoritmus meghatározza a pivot oszlopot, majd pedig a hányados-teszt segítségével a pivot pozíciót. Ezek után elkészíti a P pivot mátrixot, kiszámolja az új bázis mátrix inverzét, az új transzformált jobb oldal vektort és az új redukált költségeket.

Módosított (primál) szimplex algoritmus lineáris programozási feladatokra

---

Bemenő adatok: a *kanonikus* (P) feladat, az A mátrix, $\mathbf{b}$ és $\mathbf{c}$ vektorok. Adott továbbá egy *primál megengedett* B bázis, az I_B index halmazzal és a B^{-1} mátrixszal. Továbbá ismertek a bázis inverz segítségével transzformált vektorok $\bar{\mathbf{b}}$ és $\bar{\mathbf{c}}$, illetve az aktuális duál megoldás $\mathbf{y}$ is.

Begin
  while $\mathcal{I}_- \neq \emptyset$ **do**
    begin legyen $q \in \mathcal{I}_-$ tetszőleges;
      if $\mathbf{t}_q = B^{-1}\mathbf{a}_q \leq \mathbf{0}$
        then *STOP:* a $\mathcal{D} = \emptyset$ (*duál nem megengedettségi kritérium*)
        else
          begin

$$\vartheta := \min\left\{\frac{\bar{b}_j}{t_{jq}} : j = 1, 2, \ldots, m \text{ és } t_{jq} > 0\right\}; \text{ (hányados teszt)}$$

$$\frac{\bar{b}_k}{t_{kq}} = \vartheta = \frac{x_p}{t_{kq}} \quad p \in \mathcal{I}_B \text{ azaz } x_p \text{ a } k.\text{ bázis változó;}$$

$$x_q := \vartheta \quad x_i := x_i - \vartheta\, t_{k_i q}, \; i \in I_B \text{ és az } x_i \text{ a } k_i.\text{ bázis változó;}$$

$$\bar{v}_k := \frac{1}{t_{kq}} \quad \bar{v}_j := -\bar{v}_k t_{jq}, \; j = 1, 2, \ldots, m, \; j \neq k \text{ és } B^{-1} := PB^{-1}$$

          end
      endif
      pivotálás: $\mathcal{I}_B := \mathcal{I}_B \cup \{q\} \setminus \{p\}$, $\bar{\mathbf{b}} := P\bar{\mathbf{b}}$,
        számítsuk ki az $\mathbf{y}, \bar{\mathbf{c}}_N$ vektorokat és az I_- index halmazt
    end
  endwhile
    optimalitási kritérium: $\mathcal{I}_- = \emptyset$, számítsuk ki az optimális célfüggvényértéket;

end.

Természetesen a primál szimplex módszerhez hasonlóan, a *módosított* MBU-szimplex– illetve a *módosított* criss-cross algoritmus variánsok is elkészíthetők. A módosított variánsoknál az egyik kulcskérdés az, hogy melyik változót melyik feltételhez rendeltük bázis változóként.

A számítógépes implementációk során a módosított – nem pivot táblás – algoritmus változatokat használjuk.

Utószó

Be lehet-e fejezni egy jegyzet írását ? Nekem nem sikerült. Inkább lekerekítettem a témát és lezártam, abbahagytam, felfüggesztettem. Remélem, hogy ennek ellenére is egy hasznos, oktatásban használható anyag állt össze.

A közel 50 oldalas lineáris algebrai bevezető, majd az azt követő majdnem 100 oldalas lineáris programozás klasszikus fejezetei követik, reményeim szerint jó kiinduló pontját jelenti a lineáris optimalizálás tanulmányozásában.

Mi maradt ki ebből a jegyzetből ? Felsorolni se tudom. Azt azonban leírhatom, hogy egy hosszabb pihenés után, ha átdolgozni szeretném a jegyzetet, miről írnék biztosan.

A lineáris programozás klasszkius fejezeteinek a tárgyalásából hiányzik néhány nagyon fontos téma: a Hirsch-sejtés, a Klee-Minty illetve más exponenciális ellenpéldák témaköre vagy éppen az érzékenységvizsgálat elméleti vagy gyakorlati alkalmazásai. Ezekről a lineáris programozás előadásaimon beszélni szoktam, igaz nem hosszasan. Egyik-másik témáról szakdolgozatot is vezettem, vagy doktoranduszaimmal foglalkoztunk velük és 2000 óta, csütörtökönként megtartott szemináriumaink visszatérő témái voltak.

A pivot algoritmusok kapcsán nagyon sok mindenről lehetne még írni, hiszen az elméleti algoritmusok, amelyeket tárgyaltam, és a gyakorlatban használt, számítógépen megvalósított algoritmusok között eléggé nagy a szakadék. Talán az elméletibb jellegű lineáris programozási előadás sorozatoknak is tartalmazni kellene olyan órákat, a jegyzeteknek olyan fejezeteket, amelyek fényt derítenek arra, hogy milyen numerikus és számítógépes technikákat programoznak be annak érdekében, hogy nagy méretű lineáris programozási feladatokat hatékonyan oldjanak meg. Ezek közül, a lineáris programozási modell előfeldolgozása (preprocessing) és az eredményen elvégzett utómunkálatok témaköre bizonyára egy ilyen terület.

Sajnos ez sem pótolhatja azt a tényt, hogy az előadásaimon minden évben elmondom a *belső pontos primál-duál teljes Newton-lépéses logaritmikus barrier algoritmus*t és az előadás fóliáimon az algoritmu teljes és részletes elemzése megtalálható. Ez egy olyan algoritmus, amely az elméleti alapját képezi azoknak az algoritmusoknak, amelyek vezető lineáris programozási megoldókban kerültek implementálásra, igaz nem ez, hanem az infizibilis variánsuk.[1] Biztosan ez lesz az a fejezet, amely jegyzetem első bővítését képezni fogja.

[1]Ezzel kapcsolatban érdekes olvasmány Mészáros Csaba PhD disszertációja (1996), amelyben

A hallgatók szempontjából, valószínűleg fontos lenne, több, kidolgozott példával illusztrálni a tananyagot illetve a fejezetek végén, a tárgyalt témakörökhöz kapcsolódó gyakorló példákkal illetve gondolkodtató feladatokkal kiegészíteni az anyagot.

Jelenleg, a TÁMOP kiírás feltételeihez igazodva, számos dolgozat illetve témakör hiper hivatkozáson keresztül érhető el, azaz internet és a cikkekhez letöltési lehetőség is szükséges. A jövőben célszerű lenne, klasszikus értelemben vett hivatkozási listát is kialakítani, habár ez lényegében az egész jegyzet alapos átdolgozását jelentené, éppúgy, ahogyan a tárgymutató is, a tanulást segítő, gyorsító fontos eszköz lenne. Ezek kialakítása csak egy komolyabb átdolgozás alkalmával lesz lehetséges. Terveim szerint, először a témakörök bővítésével foglalkoznék és csak azután szánnék időt arra, hogy a klasszikus értelemben vett matematika tankönyvvé alakítsam át a jegyzetet.

egy infizibilis indítású primál-duál belső pontos algoritmust mutat be és elemzi annak gyakorlati hatékonyságát.