EÖTVÖS LORÁND TUDOMÁNYEGYETEM
TERMÉSZETTUDOMÁNYI KAR

Szabados Balázs

Trópusi lineáris programozás és alkalmazásai

Szakdolgozat
Alkalmazott Matematikus Szakirány
Matematika BSc

Témavezető:
Tóthmérész Lilla
Operációkutatási Tanszék

Budapest, 2023

Tartalomjegyzék

Köszönetnyilvánítás

Ezúton szeretnék köszönetet mondani témavezetőmnek, Tóthmérész Lillának, amiért elvállalta a felvetett témát. Hálás vagyok a rendszeres konzultációkért, ötleteiért és a rengeteg közös gondolkodásért.
Köszönettel tartozom családomnak és barátaimnak biztatásukért és támogatásukért.

Bevezetés

Bővítsük ki a valós számok halmazát a $-\infty$ elemmel és cseréljük le a jól megszokott $(+, \times)$ műveletpárt $(\max, +)$-ra. Az így kapott struktúrát nevezzük trópusi félgyűrűnek. Segítségével számos probléma egyszerűbb alakra hozható, új tételek és algoritmusok írhatók fel különböző kombinatorikai, optimalizálási feladatok megoldására.

Simon Imre, magyar származású matematikus elsők között foglalkozott ezen területtel. Brazíliában, a São Pauló-i egyetemen dolgozott - innen a trópusi jelző -, tiszteletére nevezték el trópusi matematikának.

Alapvető algebrai definíciók és problémák trópusi megfelelői fontos kombinatorikai jelentéssel bírnak. Mátrixok determinánsa maximális súlyú párosításnak, legnagyobb sajátértékük pedig maximális körátlagnak felel meg. Legrövidebb irányított utak hosszát mátrixhatványozással számíthatjuk ki. Ezekről, és hasonló kapcsolatokról szól az első fejezet.

A második szakaszban bemutatunk egy modellt, melyben különböző ütemezéselméleti feladatok kényelmesen leírhatók, majd megfogalmazzuk trópusi változataikat. Később, ezen problémák egyikére konkrét megoldást is mutatunk Krivulin egy tételének felhasználásával.

A harmadik szekcióban tárgyalt alternáló módszer központi szerepet játszik a trópusi rendszerek megoldásainak megkeresésében. Számos más algoritmus épül rá, fejleszti tovább újabb feladatok megoldása érdekében. Bebizonyítjuk az algoritmus helyességét és hogy futási ideje pszeudopolinomiális. Ezután prezentálunk két variációt az alternáló módszerre, melyek a feladat egészértékű megoldásait hivatottak megtalálni.

A klasszikus lineáris programozás mintájára bevezetjük a maxlineáris programozást a negyedik részben. Definiáljuk a maxlineáris függvényeket és különböző algoritmusokat adunk a feltételrendszertől és attól függően, hogy a célfüggvényt éppen minimalizálni vagy maximalizálni szeretnénk.

Végül, kitekintésképp először egy játékelméleti feladat kapcsolatát írjuk fel egy trópusi rendszerrel, majd pedig egy filogenetikai tétel matematikai változatát mutatjuk be a trópusi algebra és geometria eszközeinek segítségével.

1. A trópusi algebra és kapcsolata a kombinatorikával

1.0.1. Definíció. Legyenek $a, b \in \mathbb{T} \coloneqq \mathbb{R} \cup \{-\infty\}$. Ekkor

- $a \oplus b \coloneqq \max(a, b)$,
- $a \otimes b \coloneqq a + b$.

1.0.2. Állítás. $(\mathbb{T}, \oplus, \otimes)$ *kommutatív félgyűrű.*

Bizonyítás. Legyenek $a, b, c \in \mathbb{T}$.

1. $(\mathbb{T}, \oplus)$ kommutatív egységelemes félcsoport, melynek neutrális eleme $\varepsilon \coloneqq -\infty$:
   - $(a \oplus b) \oplus c = \max(\max(a, b), c) = \max\{a, b, c\} = \max(a, \max(b, c)) = a \oplus (b \oplus c)$
   - $\varepsilon \oplus a = \max(\varepsilon, a) = a = \max(a, \varepsilon) = a \oplus \varepsilon$
   - $a \oplus b = \max(a, b) = \max(b, a) = b \oplus a$
2. $(\mathbb{T}, \otimes)$ kommutatív egységelemes félcsoport, melynek egységeleme 0:
   - $(a \otimes b) \otimes c = (a + b) + c = a + (b + c) = a \otimes (b \otimes c)$
   - $0 \otimes a = 0 + a = a = a + 0 = a \otimes 0$
   - $a \otimes b = a + b = b + a = b \otimes a$
3. A trópusi szorzás balról és jobbról is disztributív a trópusi összeadásra nézve:
   - $a \otimes (b \oplus c) = a + \max(b, c) = \max(a + b, a + c) = (a \otimes b) \oplus (a \otimes c)$
   - $(a \oplus b) \otimes c = \max(a, b) + c = \max(a + c, b + c) = (a \otimes c) \oplus (b \otimes c)$
4. $\mathbb{T}$ annihilátora ε:
   - $\varepsilon \otimes a = \varepsilon + a = \varepsilon = a + \varepsilon = a \otimes \varepsilon$

$\square$

1.0.3. Megjegyzés. A félgyűrű teljesen rendezett a $\leq$ relációval, ahol $x \leq y \Leftrightarrow x \oplus y = y$. $\varepsilon \neq x \in \mathbb{T}$ esetén x-nek nem létezik ellentettje, azaz $\nexists y \in \mathbb{T} : x \oplus y = \varepsilon$, hiszen $x \oplus y \geq x > \varepsilon$. Ellenben minden $x \in \mathbb{T}$-nek van inverze, vagyis $\exists x^{-1} \in \mathbb{T} : x \otimes x^{-1} = 0$. A szokásos értelemben vett ellentett, $x^{-1} \coloneqq -x$ jó választás lesz, mert ekkor $x \otimes x^{-1} = x + (-x) = 0$.

A következőkben megvizsgáljuk néhány alapvető lineáris algebrai definíció és probléma trópusi változatát, valamint azok kombinatorikai jelentését. Ehhez főként Butkovič publikációit használjuk fel [4, 6].

1.0.4. Definíció (Trópusi determináns)**.** Legyen $A = (a_{ij}) \in \mathbb{T}^{n \times n}$. Ekkor $\det_T(A) \coloneqq \bigoplus_{\pi \in S_n} a_{1\pi(1)} \otimes a_{2\pi(2)} \otimes \cdots \otimes a_{n\pi(n)}$.

Adott $A \in \mathbb{T}^{m\times n}$ mátrix esetén elkészíthetjük a $w : E \to \mathbb{T}$ súlyfüggvénnyel ellátott $G_A = (S, T; E)$ teljes páros gráfot: S csúcsait A soraival, T csúcsait pedig A oszlopaival asszociáljuk, $w(\{i, j\}) := a_{ij}$.

1.0.5. Állítás. *Ekkor* $\det_T(A)$ *értéke megegyezik a* G_A*-beli maximális súlyú teljes párosítás súlyával.*

Bizonyítás. Nézzük meg, mit számolunk ki valójában a trópusi determinánssal: $\det_T(A) = \max_{\pi\in S_n} a_{1\pi(1)}+a_{2\pi(2)}+\cdots+a_{n_\pi(n)}$. Azaz kiválasztunk minden sorból és oszlopból pontosan egy elemet, ezeket összeadjuk, majd az összes ilyen lehetőség közül kiválasztjuk a maximális értékűt. Tehát, azon tagokból, melyekből a maximumot választjuk, egy tag éppen egy teljes párosításnak felel meg G_A-ban. □

Így a trópusi determináns kiszámítására a Gauss-elimináció helyett egy másik $\mathcal{O}(n^3)$ idejű algoritmus, a magyar módszer alkalmas.

Egy $x \in \mathbb{T}^n$ vektort *reguláris*nak nevezünk, ha nem szerepel benne ε elem. Vezessük be az $x^- := (x_i^-)$ sorvektort, ahol $x_i^- = x_i^{-1}$, ha $x_i \neq \varepsilon$, $x_i^- = \varepsilon$ különben. Mátrixok trópusi szorzása a szokásos módon történik, azzal a különbséggel, hogy a standard műveletek helyett azok trópusi megfelelőjét alkalmazzuk.

1.0.6. Példa (Mátrixszorzás)**.** Legyen $A = \begin{pmatrix} 14 & 8 & -6 \\ \varepsilon & -9 & 3 \end{pmatrix}$, $x = \begin{pmatrix} 8 \\ -4 \\ 7 \end{pmatrix}$. Ekkor

$$A \otimes x = \begin{pmatrix} (14\otimes 8)\oplus(8\otimes -4)\oplus(-6\otimes 7) \\ (\varepsilon\otimes 8)\oplus(-9\otimes -4)\oplus(3\otimes 7) \end{pmatrix} = \begin{pmatrix} \max\{14+8, 8-4, -6+7\} \\ \max\{\varepsilon+8, -9-4, 3+7\} \end{pmatrix} = \begin{pmatrix} 22 \\ 10 \end{pmatrix}.$$

A mátrixszorzás a trópusi esetben is hasonló tulajdonságokkal rendelkezik, mint a klasszikusban.

1.0.7. Állítás. *Mátrixok trópusi szorzása asszociatív, de nem kommutatív.*

Bizonyítás. A kommutativitás nem teljesül, például

$$\begin{pmatrix} 1 & 1 \\ -1 & -1 \end{pmatrix} \otimes \begin{pmatrix} -2 & -1 \\ 1 & 2 \end{pmatrix} = \begin{pmatrix} 2 & 3 \\ 0 & 1 \end{pmatrix},$$

de

$$\begin{pmatrix} -2 & -1 \\ 1 & 2 \end{pmatrix} \otimes \begin{pmatrix} 1 & 1 \\ -1 & -1 \end{pmatrix} = \begin{pmatrix} -1 & -1 \\ 2 & 2 \end{pmatrix}.$$

Az asszociativitás belátásához jelölje $i.$ egy mátrix i-edik sorát. Szeretnénk, hogy $((AB)C)_{ij} = (A(BC))_{ij}$ minden i, j-re.

$$(AB)_{i.} = (\max_k a_{ik} + b_{k1}, \max_k a_{ik} + b_{k2}, \ldots, \max_k a_{ik} + b_{kn})$$

Innen

$$((AB)C)_{ij} = \max\{\max_k(a_{ik} + b_{k1}) + c_{1j}, \ldots, \max_k(a_{ik} + b_{kn}) + c_{nj}\}.$$

Hasonló számolással:

$$(A(BC))_{ij} = \max\{\max_k(b_{1k} + c_{kj}) + a_{i1}, \dots, \max_k(b_{nk} + c_{kj}) + a_{in}\}.$$

Tehát

$$((AB)C)_{ij} = (A(BC))_{ij} = \max_{k,l} a_{ik} + b_{kl} + c_{lj}.$$

□

1.0.8. Definíció (Mátrixhatványozás)**.** $M \in \mathbb{T}^{n\times n}$ esetén $M^{\otimes k} := M^{\otimes(k-1)} \otimes M$. Egy M mátrix nulladik hatványa legyen $M^{\otimes 0} = I = \begin{pmatrix} 0 & \varepsilon & \dots & \varepsilon \\ \varepsilon & 0 & \dots & \varepsilon \\ \vdots & \vdots & \ddots & \vdots \\ \varepsilon & \varepsilon & \dots & 0 \end{pmatrix}$.

A későbbiekben az olvashatóság kedvéért $M^{\otimes k}$ helyett egyszerűen M^k-t írunk.

Mátrixok sajátértékeinek és sajátvektorainak meghatározása fontos klasszikus algebrai probléma. A kérdés megfogalmazható a max-algebrában is, azaz egy adott mátrix sajátértékeit és a hozzájuk tartozó sajátvektorokat keressük úgy, hogy az a trópusi műveletekkel megfogalmazott egyenletnek tegyen eleget.

1.0.9. Probléma (Sajátérték-sajátvektor)**.** $A \in \mathbb{T}^{n\times n}$ esetén $A \otimes x = \lambda \otimes x$ mely $x \in \mathbb{T}^n$ vektorokra, $\lambda \in \mathbb{T}$ skalárokra teljesül?

Jelölje $\lambda(A)$ az A mátrix legnagyobb sajátértékét. Definiáljuk az $A \in \mathbb{T}^{n\times n}$ mátrixhoz tartozó $D_A = (V, E)$ irányított gráfot a $w : E \to \mathbb{R}$ súlyfüggvénnyel: $E := \{(i,j) : a_{ij} > \varepsilon\}$, és minden $(i,j) = e \in E$ esetén $w(e) := a_{ij}$.

1.0.10. Tétel (Butkovič [5, 4.2. fejezet])**.** *Legyen $A \in \mathbb{T}^{n\times n}$. Ekkor $\lambda(A)$ megegyezik a D_A-beli maximális körátlaggal (maximum cycle mean), azaz*

$$\lambda(A) = \max \tfrac{a_{i_1 i_2} + a_{i_2 i_3} + \dots + a_{i_{k-1} i_k} + a_{i_k i_1}}{k},$$

ahol a maximumot D_A összes $(i_1, i_2, \dots, i_k, i_1)$ köréből választjuk.

A tétel bizonyítását mellőzzük, mert bár egyszerű fogalmak között teremt kapcsolatot, belátásához mégis meglepően sok lemmára van szükség.
$\lambda(A) = \varepsilon$ pontosan akkor teljesül, ha D_A aciklikus ($\max \emptyset = \varepsilon$), valamint egy D_A gráf élsúlyozása akkor és csak akkor konzervatív, ha $\lambda(-A) \le 0$. Ha $A \in \mathbb{R}^{n\times n}$, akkor $\lambda(A)$ az egyetlen sajátértéke A-nak [9]. Ezen sajátérték kiszámítható $\mathcal{O}(n^3)$ időben Karp algoritmusával [12].

Tegyük fel, hogy adott élsúlyozott D_A digráfban szeretnénk minden csúcsból minden csúcsba a legrövidebb irányított út hosszát megtalálni, azaz olyan $\bar{A}$ mátrixot akarunk konstruálni, melyre $\bar{a}_{ij}$ értéke megegyezik a legrövidebb $i \to j$ út hosszával. Vegyük észre, hogy az A^k mátrix i-edik sorának j-edik eleme éppen a legrövidebb, k élből álló $i \to j$ séta hossza.

1.0.11. Definíció (Kleene star)**.** Legyen $A \in \mathbb{T}^{n \times n}$. $A^* := \bigoplus_{k=0}^{\infty} A^k = I \oplus A \oplus A^2 \oplus \dots$.

1.0.12. Állítás (Butkovič [5, 1.6.10 Állítás])**.** *Ha $\lambda(A) \leq 0$, akkor $A^* = I \oplus A \oplus \cdots \oplus A^{k-1}$ minden $k \geq n$ esetén.*

Bizonyítás. A feltétel szerint $\lambda(A) \leq 0$. Ekkor az 1.0.10 Tétel miatt D_A-ban csak nempozitív összköltségű kör van. Így a D_{-A} gráf konzervatív élsúlyozású, azaz $\bar{A} = A \oplus A^2 \oplus \cdots \oplus A^{n-1}$, valamint $A^* = I \oplus \bar{A}$. □

1.0.13. Állítás. *Ha $\lambda(A) \leq 0$, akkor A^* kiszámítható $\mathcal{O}(n^3)$ időben a Floyd-Warshall algoritmus segítségével.*

A bizonyítás előtt elevenítsük fel a Floyd-Warshall algoritmust.

Algoritmus 1 Floyd-Warshall [13]

for $k = 1 \dots n$ **do**
    for $i = 1 \dots n$ **do**
        for $j = 1 \dots n$ **do**
            $A(i,j) := \min[A(i,j), A(i,k) + A(k,j)]$
        end for
    end for
end for

Bizonyítás. Ismét felhasználva, hogy $\lambda(A) \leq 0$, a D_{-A} gráf konzervatív súlyozású. Figyeljük meg, hogy erre a gráfra futtatva a Floyd-Warshall algoritmust, az output éppen a $-\bar{A}$ mátrix. Innen pedig A^* egyszerűen megkapható: $A^* = I \oplus \bar{A}$. □

Általában A^* kiszámítása nem könnyű feladat, hiszen a $\lambda(A) \leq 0$ feltétel éppen a konzervativitást biztosította, nemkonzervatív súlyozású gráfban pedig nehéz a legrövidebb utak megtalálása.

Egy mátrix trópusi értelemben vett nyoma a $\mathrm{tr}(A) = \bigoplus_{i=1}^{n} a_{ii}$ kifejezés értéke, normája pedig $||A|| = \bigoplus_{i=1}^{n} \bigoplus_{j=1}^{n} a_{ij}$. Ezenkívül legyen $\mathrm{Tr}(A) = \bigoplus_{m=1}^{n} \mathrm{tr}(A^m)$. Ekkor a legnagyobb sajátérték felírható

$$\lambda(A) = \bigoplus_{m=1}^{n} \mathrm{tr}^{1/m}(A^m)$$

alakban is az 1.0.10 Tétel alapján.

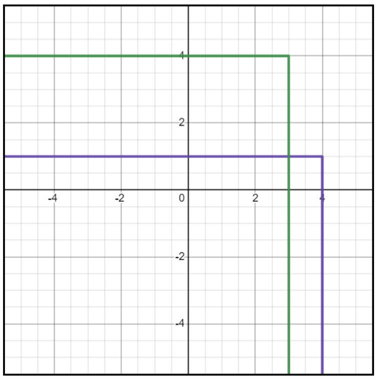
1.0.14. Állítás (Carrè-egyenlőtlenség)**.** *Tegyük fel, hogy $\mathrm{Tr}(A) \leq 0$. Ekkor $A^m \leq A^* \quad \forall m \geq 0$.*

Bizonyítás. $\mathrm{Tr}(A) = \bigoplus_{m=1}^{n} \mathrm{tr}(A^m) \leq 0 \Rightarrow$ minden m-re A^m főátlójában csak nempozitív elemek állnak. Ekkor D_A-ban csak nempozitív összköltségű irányított körök vannak, ezért $\lambda(A) \leq 0$. Az 1.0.12 állítás miatt $A^* = \bigoplus_{l=0}^{k-1} A^l$ minden $k \geq n$-re, azaz $A^* = I \oplus A \oplus \cdots \oplus A^{n-1}$. Ha $m \leq n-1$, akkor A^m szerepel az előbbi tagok közt, így igaz az $A^m \leq A^*$ egyenlőtlenség. Ha $m > n-1$, akkor vegyük észre, hogy $A^m \leq A^{n-1}$, ezért $A^m \leq A^*$ ismét teljesül. □

Térjünk át trópusi lineáris, azaz $A \otimes x = b$ alakú rendszerek vizsgálatára [4] felhasználásával, ahol $A = (a_{ij}) \in \mathbb{R}^{m \times n}, x \in \mathbb{R}^n, b \in \mathbb{R}^m$. Elsőre talán nehéz elképzelni, hogyan is néznek ki egy trópusi lineáris egyenletrendszer megoldásai. Ehhez tekintsük a következő példát:

$$A = \begin{pmatrix} -2 & 1 \\ 0 & -1 \end{pmatrix}, u = \begin{pmatrix} x \\ y \end{pmatrix}, b = \begin{pmatrix} 2 \\ 3 \end{pmatrix}.$$

Zölddel jelöljük az A mátrix első sorához tartozó egyenletet, lilával pedig a másodikhoz tartozót (1. ábra). Az egyenletrendszer megoldása a $(x, y) = (3, 1)$ pár.



1. ábra. Az $Au = b$ egyenletrendszer megoldásai

Ezen feladatoknál egy fontos kérdés, hogy adott A, b esetén létezik-e (egyértelmű) x megoldás. A jobb oldalt nullává tehetjük, normalizálhatjuk a rendszert, ha beszorzunk

$$B = \begin{pmatrix} b_1^{-1} & \varepsilon & \dots & \varepsilon \\ \varepsilon & b_2^{-1} & \dots & \varepsilon \\ \vdots & \vdots & \ddots & \vdots \\ \varepsilon & \varepsilon & \dots & b_m^{-1} \end{pmatrix}$$

-vel balról, ahol $b_i^{-1} = -b_i$.

1.0.15. Állítás. *$\tilde{x}$ pontosan akkor megoldása az $A \otimes x = b$ egyenletrendszernek, ha $\tilde{x}$ megoldása a $B \otimes A \otimes x = 0$ rendszernek is.*

Bizonyítás. $(\Rightarrow)$ $B \otimes A \otimes \tilde{x} = B \otimes b = 0$.

$(\Leftarrow)$ $B \otimes A \otimes \tilde{x} = 0$. Szorozzunk be a $B^{-1} = \begin{pmatrix} b_1 & \varepsilon & \dots & \varepsilon \\ \varepsilon & b_2 & \dots & \varepsilon \\ \vdots & \vdots & \ddots & \vdots \\ \varepsilon & \varepsilon & \dots & b_m \end{pmatrix}$ mátrixszal balról. Ekkor $I \otimes A \otimes \tilde{x} = A \otimes \tilde{x} = B^{-1} \otimes 0 = b$. $\square$

Így elég normalizált rendszereket vizsgálnunk. Jelölje

$$S = \{x \in \mathbb{R}^n : A \otimes x = 0\}$$

a megoldások halmazát, $\bar{x}_j := -\max a_{ij}$ minden $j \in [n]$-re, $\bar{x} := (\bar{x}_1, \dots, \bar{x}_n)^T$, $M_j := \{k \in [m] : a_{kj} = \max_i a_{ij}\}$ minden $j \in [n]$-re.

1.0.16. Tétel (Zimmermann [19]). *$x \in S$ pontosan akkor, ha $x \leq \bar{x}$ és $\bigcup_{j \in N_x} M_j = M$, ahol $N_x = \{j \in [n] : x_j = \bar{x}_j\}$.*

1.0.17. Következmény. *A következő állítások ekvivalensek:*

(i) $S \neq \emptyset$,

(ii) $\bar{x} \in S$,

(iii) $\bigcup_{j \in [n]} M_j = M$.

1.0.18. Következmény. *$S = \{\bar{x}\}$ pontosan akkor, ha*

(i) $\bigcup_{j \in [n]} M_j = M$ *és*

(ii) $\bigcup_{j \in N'} M_j \neq M \ \forall N' \subseteq N, N' \neq N$.

Tekintsük a következő feladatokat:

1.0.19. Probléma ([Egyértelmű] megoldhatóság). Mikor van az $A \otimes x = b$ rendszernek [egyértelmű] megoldása?

1.0.20. Probléma ([Minimális] fedés). Adott egy véges M halmaz, és annak $M_1, \dots, M_n$ részhalmazai. $\bigcup_{j=1}^{n} M_j = M$ [és $\bigcup_{j=1, j \neq k}^{n} M_j \neq M$]?

1.0.21. Tétel. *Minden trópusi lineáris rendszerhez tudunk konstruálni egy véges alaphalmazt és hozzá részhalmazokat úgy, hogy az 1.0.19 és 1.0.20 problémák ekvivalensek legyenek.*

Bizonyítás. Következik az 1.0.17 és 1.0.18 következményekből. □

Az előbbi tétel fordítva is igaz, azaz ha adott egy alaphalmaz és neki részhalmazai, akkor tudunk olyan trópusi feladatot gyártani, hogy az 1.0.19 és 1.0.20 problémák ekvivalensek. Legyen $M = [m]$, $M_1, \ldots, M_n$ részhalmazai M-nek. Definiáljuk az $A = (a_{ij}) \in \mathbb{R}^{m \times n}$ mátrixot a következőképp:

$$a_{ij} = \begin{cases} 1 & \text{ha} \quad i \in M_j \\ 0 & \text{különben} \end{cases} \quad \forall i \in [m], j \in [n].$$

A következő két tételre saját bizonyítást adunk.

1.0.22. Tétel. $\displaystyle\bigcup_{j \in [n]} M_j = M \iff \exists x : A \otimes x = 0.$

Bizonyítás. ($\Rightarrow$) Az A mátrix minden sorában van legalább egy 1-es. Legyen $x_j = -1$, ha A j-edik oszlopában van 1-es, 0 különben. Ekkor az x vektor megoldása az $A \otimes x = 0$ rendszernek.
($\Leftarrow$) Tegyük fel, hogy $\bigcup_{j \in [n]} M_j \neq M$. Ekkor létezik csupa nulla sor az A mátrixban, jelöljük ezt a_i-vel. Feltehető, hogy az M_j halmazok nemüresek, így A-nak van olyan sora, melynek j-edik koordinátája 1. Ekkor egy x lehetséges megoldás j-edik eleme mindenképp -1 kell legyen. De ilyenkor $a_i \otimes x = -1$, ami ellentmondás. □

1.0.23. Tétel. $\displaystyle\bigcup_{j \in [n]} M_j = M \textit{ és } \bigcup_{j \in N'} M_j \neq M \quad \forall N' \subseteq N, N' \neq N \iff \exists! x : A \otimes x = 0.$

Bizonyítás. ($\Rightarrow$) x létezése következik az előző tételből, azt kell belátnunk, hogy egyértelmű is. Indirekt tegyük fel, hogy x és $\tilde{x}$ is megoldás és a j-edik koordinátájukban különböznek. Ekkor az $a_i \otimes x$ és az $a_i \otimes \tilde{x}$ kifejezésekben a maximum semelyik i-re nem a j-edik koordinátában vétetik fel. Töröljük A-ból a j-edik oszlopot. Feltehető, hogy ekkor az $A \otimes x = 0$ rendszernek már csak egyetlen megoldása van. Az előző tétel miatt $\bigcup_{k \neq j} M_k = M$, azaz a részhalmazok uniója M_j nélkül is fedi M-et, ami ellentmondás.
($\Leftarrow$) A részhalmazok uniója valóban fedi M-et az előző tétel miatt, kell még, hogy ez egy minimális fedés is egyben. Tegyük fel, hogy egy M_j halmaz elhagyásával az unió még mindig fedi az alaphalmazt. Ekkor az A mátrix minden sorának, amelynek j-edik eleme 1, van legalább még egy nemnulla eleme. Legyen $x_i = 0$ minden $i \neq j$-re. Ekkor tetszőleges $x_j < -1$ esetén az x vektor megoldása az $A \otimes x = 0$ egyenletnek, de ez ellentmond a feltételünknek, miszerint pontosan egy megoldás létezik. □

2. Alkalmazás az ütemezéselméletben

Ebben a fejezetben bemutatunk egy viszonylag általános modellt, melyben felírunk néhány ütemezéselméleti feladatot, valamint megfogalmazzuk őket trópusi lineáris program formájában, majd hatékony módszert mutatunk egyikük megoldására. Mindehhez nagyrészt Nikolai Krivulin cikkét [15] használjuk fel.

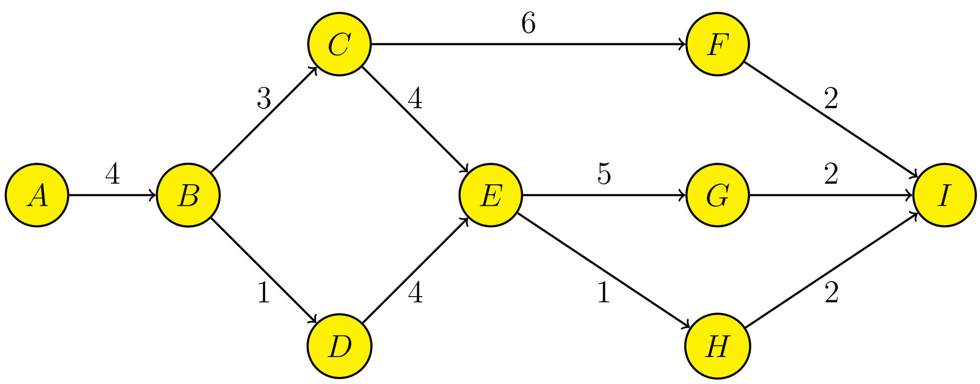
2.1. A probléma általánosan

Tegyük fel, hogy ütemezést kell elkészítenünk n darab munkához úgy, hogy a teljes átfutási idő minimális legyen. A következő megszorítások nehezíthetik az optimális megengedett ütemezés megtalálását.

- Rendelkezésre állási idő (g_j): azon időpillanat, amikor legkorábban elkezdhetjük az adott munkát
- Határidő (f_j)
- Kezdés-befejezés (c_{ij}): alsó korlát a j munka megkezdése és az i munka befejezése közti időintervallumra ($c_{ij} := \varepsilon$, ha nincs kezdés-befejezés kikötés az i, j párra)
- Befejezés-kezdés (d_{ij}): alsó korlát a j munka befejezése és az i munka megkezdése közti időintervallumra ($d_{ij} := \varepsilon$, ha nincs megadva befejezés-kezdés feltétel az i, j párra)

Jelölje x_i azon időpillanatot, amikor az i munkát elkezdjük, y_i pedig azt, amikor befejezzük minden $i \in \{1, 2, \ldots, n\}$-re.

Bemelegítésként tekintsük a következő, PERT-módszerrel megoldható feladatot leíró $G = (V, A)$ élsúlyozott digráfot és írjuk fel az előbbiekben megfogalmazott modell segítségével.



Tegyük fel, hogy a j munkát legkorábban az $j + 1$ időpillanatban tudjuk elkezdeni, és szeretnénk végezni az egész projekttel legkésőbb a 18 időpillanatig.
Először írjuk fel a kezdési időkre vonatkozó feltételt,

$$g_j = j + 1 \quad j = 1, \ldots, 11,$$

illetve azt, hogy legkésőbb a 18 időpillanatig be szeretnénk fejezni minden munkát:

$$f_j = 18 \quad j = 1, \ldots, 11.$$

Végül pedig írjuk fel a feladatok sorrendjét és az elvégzésükhöz szükséges időket:

$$c_{ji} = \begin{cases} w(e_i) + w(e_j) & \text{ha} \quad e_i, e_j \in A, i \neq j \\ w(e_i) & \text{ha} \quad e_i \in A, i = j \\ \varepsilon & \text{különben} \end{cases},$$

valamint

$$d_{ji} = \begin{cases} 0 & \text{ha} \quad e_i, e_j \in A, i \neq j \\ -w(e_i) & \text{ha} \quad e_i \in A, i = j \\ \varepsilon & \text{különben} \end{cases}.$$

Most vizsgáljuk meg a problémát általánosan is, ha a cél az átfutási idő minimalizálása. Kezdetben legyenek csak kezdés-befejezés feltételeink:

$$x_j + c_{ij} \leq y_i \quad j = 1, \ldots, n.$$

Mivel azt szeretnénk, hogy az átfutási idő minimális legyen, ezért biztosan lesz olyan munka, melyre a fenti egyenlőtlenség egyenlőséggel teljesül, így

$$y_i = \max_{j \in \{1, \ldots, n\}} (x_j + c_{ij}).$$

Az átfutási idő felírható a legkésőbbi befejezési idő és a legkorábbi kezdési idő különbségeként:

$$\max_{i \in \{1, \ldots, n\}} y_i - \min_{i \in \{1, \ldots, n\}} x_i = \max_{i \in \{1, \ldots, n\}} y_i + \max_{i \in \{1, \ldots, n\}} (-x_i).$$

Behelyettesítve az előbb nyert képletet y_i-re, a következőt kapjuk:

2.1.1. Probléma. Adott n munka c_{ij} kezdés-befejezés megszorításokkal. Mely x vektorra lesz a

$$\max_{i \in \{1, \ldots, n\}} \max_{j \in \{1, \ldots, n\}} (x_j + c_{ij}) + \max_{i \in \{1, \ldots, n\}} (-x_i)$$

kifejezés értéke minimális?

A probléma egyszerűbb alakra hozható, ha alkalmazzuk a trópusi műveleteket:

$$\min\Big(\bigoplus_{i \in \{1, \ldots, n\}} \bigoplus_{j \in \{1, \ldots, n\}} c_{ij} x_j \Big)\Big(\bigoplus_{i \in \{1, \ldots, n\}} x_i^{-1} \Big),$$

ahol a konkatenáció trópusi szorzást jelöl.

Nézzük meg, hogyan írhatjuk fel a rendelkezésre állási idő és határidő feltételeket.

$$g_i \leq x_i, \quad y_i = \max_{j \in \{1,\ldots,n\}} (x_j + c_{ij}) \leq f_i$$

Szintén trópusi műveletekre áttérve, illetve hozzávéve az előző problémához a következő feladatot kapjuk:

2.1.2. Probléma. Adott n munka c_{ij} kezdés-befejezés feltételekkel, g_i rendelkezésre állási időkkel és f_i határidőkkel.

$$\begin{gathered}\bigoplus_{j \in \{1,\ldots,n\}} c_{ij} x_j \leq f_i, \\ g_i \leq x_i, \quad i = 1, \ldots, n, \\ \min\Big(\bigoplus_{i \in \{1,\ldots,n\}} \bigoplus_{j \in \{1,\ldots,n\}} c_{ij} x_j\Big)\Big(\bigoplus_{i \in \{1,\ldots,n\}} x_i^{-1}\Big).\end{gathered}$$

Végül vizsgáljuk meg a befejezés-kezdés feltételt is. Ezen feltételek a következőképp fogalmazhatók meg:

$$y_j + d_{ij} \leq x_i, \quad j = 1, \ldots, n.$$

Ismét felhasználva, az $y_i = \max_{j \in \{1,\ldots,n\}} (x_j + c_{ij})$ egyenlőséget:

$$\max_{k \in \{1,\ldots,n\}} (x_k + c_{jk}) + d_{ij} \leq x_i, \quad j = 1, \ldots, n,$$

illetve egy egyenlőtlenséggé átírva a rendszert:

$$\max_{j \in \{1,\ldots,n\}} \Big(\max_{k \in \{1,\ldots,n\}} (x_k + c_{jk}) + d_{ij}\Big) \leq x_i.$$

Ugyanezt trópusi műveletekkel így írhatjuk:

$$\bigoplus_{j \in \{1,\ldots,n\}} \Big(\bigoplus_{k \in \{1,\ldots,n\}} c_{jk} x_k\Big) d_{ij} \leq x_i.$$

Ezen feltételeket figyelembe véve és kombinálva a rendelkezésre állási idő megszorításokkal felírhatjuk az alábbi problémát.

2.1.3. Probléma. Adott n munka c_{ij} kezdés-befejezés, d_{ij} befejezés-kezdés és g_i rendelkezésre állási idő feltételekkel.

$$\begin{gathered}\bigoplus_{j \in \{1,\ldots,n\}} \Big(\bigoplus_{k \in \{1,\ldots,n\}} c_{jk} x_k\Big) d_{ij} \leq x_i, \\ g_i \leq x_i, \quad i = 1, \ldots, n, \\ \min\Big(\bigoplus_{i \in \{1,\ldots,n\}} \bigoplus_{j \in \{1,\ldots,n\}} c_{ij} x_j\Big)\Big(\bigoplus_{i \in \{1,\ldots,n\}} x_i^{-1}\Big).\end{gathered}$$

Más, ütemezéselméletben gyakran használt célfüggvények is megfogalmazhatók az új műveleteinkkel. Írjunk fel ezek közül néhányat.
A $\sum C_j$ célfüggvény minimalizálásakor azt szeretnénk, hogy a munkák befejezési idejeinek összege a lehető legkisebb legyen. Ezt az alábbi módon írhatjuk:

$$\min \bigotimes_i y_i$$

Az $L_{\max}$ függvény értéke a legtöbbet késő munka késése. Ahhoz, hogy ezt felírhassuk, készítsük el az $y := (y_1, y_2, \ldots, y_n)$ vektort. A cél trópusi alakja:

$$\min y \otimes f^-$$

Ha a késő munkák számát szeretnénk a minimalizálni, akkor a $\sum U_j$ függvényérték csökkentése a feladat, ahol $U_j = \begin{cases} 1 & \text{ha a } j \text{ munka késik} \\ 0 & \text{különben} \end{cases}$. Trópusi jelölésekkel:

$$\min \bigotimes_j U_j$$

Megjegyezzük, hogy a második kivételével, ezen célfüggvények nem lineárisak a trópusi műveletekkel, ellentétben a klasszikus esettel.

2.2. A 2.1.3. Probléma megoldása

Ebben az alfejezetben megoldást adunk a 2.1.3 Problémára. Az előző részben felírt 2.1.1 és 2.1.2 feladatok hasonlóképp megoldhatók különböző lemmák és tételek segítségével.

Először írjuk át más alakra a 2.1.3 Problémát. Vezessük be a $C := (c_{ij})$ és $D := (d_{ij})$ mátrixokat, valamint a $\mathbf{0} := (0, 0, \ldots, 0)^T$ vektort. Ekkor a probléma így írható:

$$\begin{gathered} DCx \oplus g \leq x \\ \min \mathbf{0}^T C x x^- \mathbf{0} \end{gathered}$$

A következőkben ismertetjük a probléma megoldásához szükséges eredményeket. Tekintsük kezdésnek az alábbi problémát.

2.2.1. Probléma. Adottak az $A, B \in \mathbb{T}^{n\times n}$ mátrixok és a $g \in \mathbb{T}^n$ vektor. Szeretnénk olyan $x \in \mathbb{T}^n$ reguláris vektort találni, amely minimalizálja az

$$x^- A x$$

kifejezést és kielégíti a

$$Bx \oplus g \leq x$$

feltételt.

2.2.2. Tétel (Krivulin [14])**.** *Legyenek az A és B mátrixok olyanok, melyekre $\lambda(A) > \varepsilon$ és* $\mathrm{Tr}(B) \leq 0$. *Ekkor a 2.2.1 Problémában a minimum értéke*

$$\theta = \bigoplus_{k=1}^{n} \bigoplus_{0 \le i_1 + \dots + i_k \le n-k} \mathrm{tr}^{1/k}(AB^{i_1} \dots AB^{i_k})$$

és az összes reguláris megoldás így írható:

$$x = (\theta^{-1} A \oplus B)^* u, \quad u \ge g.$$

A következő a 2.1.3 probléma általánosítása.

2.2.3. Probléma. Legyen $B \in \mathbb{T}^{n \times n}$, $g, p, q \in \mathbb{T}^n$. Keressük azon $x \in \mathbb{T}^n$ vektorokat, hogy

$$\begin{array}{l} Bx \oplus g \le x \\ \min q^- x x^- p \end{array}$$

Megmutatjuk, hogy ezen feladat hogyan oldható meg, majd visszavezetjük rá a 2.1.3 problémát.

2.2.4. Tétel. *Legyen $p \neq \mathbf{0}$, q reguláris vektor és B olyan mátrix, melyre* $\mathrm{Tr}(B) \le 0$. *Ekkor a minimum értéke a 2.2.3 Problémában*

$$\theta = q^- B^* p$$

és az összes reguláris megoldás alakja:

$$x = (\theta^{-1} p q^- \oplus B)^* u, \quad u \ge g.$$

Bizonyítás. Első lépésként írjuk át más alakra a célfüggvényünket.

$$q^- x x^- p = (q^- x)(x^- p) = (x^- p)(q^- x) = x^- p q^- x,$$

ahol kihasználtuk a mátrixszorzás asszociativitását. A középső egyenlőség azért igaz, mert $q^- x, x^- p \in \mathbb{T}$, így ezekre a trópusi szorzás kommutatív. Ismét kihasználva az asszociativitást:

$$x^- p q^- x = x^- (p q^-) x =: x^- A x.$$

Alkalmazva a 2.2.2 Tételt, a minimum értéke:

$$\theta = \bigoplus_{k=1}^{n} \bigoplus_{0 \le i_1 + \dots + i_k \le n-k} \mathrm{tr}^{1/k}(p q^- B^{i_1} \dots p q^- B^{i_k}).$$

Az A mátrix egy eleme: $a_{ij} = p_i - q_j$. Kihasználva a mátrix nyomának definícióját:

$$\mathrm{tr}(p q^- B) = \max_{ij}(p_i - q_j + B_{ji})$$

Legyen $c = q^- B$. Ekkor $c_i = \max_j (B_{ji} - q_j)$, valamint $cp = \max_i (c_i + p_i)$. Tehát

$$q^- B p = \max_{ij}(B_{ji} - q_j + p_i).$$

Azaz

$$\mathrm{tr}(p q^- B) = q^- B p,$$

amiből

$$\mathrm{tr}(pq^- B^{i_1} \dots pq^- B^{i_k}) = q^- B^{i_1} p \dots q^- B^{i_k} p.$$

Alsó becslést nyerünk θ-ra, ha a $k = 1$ tag kivételével az összeset tagot elhagyjuk.

$$\theta \geq \bigoplus_{i=0}^{n-1} \mathrm{tr}(pq^- B^i) = \bigoplus_{i=0}^{n-1} q^- B^i p = q^- B^* p$$

Most adjunk felső becslést θ értékére. A Carrè-egyenlőtlenséget felhasználva az alábbira jutunk.

$$q^- B^{i_1} p \dots q^- B^{i_k} p \leq q^- B^* p \dots q^- B^* p = (q^- B^* p)^k$$

Ennek felhasználásával megfelelő becslést kapunk.

$$\theta = \bigoplus_{k=1}^{n} \bigoplus_{0 \leq i_1 + \dots + i_k \leq n-k} (q^- B^{i_1} p \dots q^- B^{i_k} p)^{1/k} \leq q^- B^* p$$

Azaz összességében $q^- B^* p \leq \theta \leq q^- B^* p$. Így beláttuk, hogy $\theta = q^- B^* p$. Végül meg kell mutatnunk, hogy az összes megoldás előáll a tételben szereplő alakban. Ismét az $A = pq^-$ helyettesítéssel alkalmazva a 2.2.2 tételt, következik az állítás. $\square$

Használjuk fel az előbbi tételünket. $p = \mathbf{0}$, $q^- = \mathbf{0}^T C$ és $B = DC$ helyettesítéssel megoldást adhatunk az alfejezet elején kimondott, 2.1.3-mal ekvivalens problémára. Mondjuk is ki ezt egy tétel formájában.

2.2.5. Tétel. *Legyen C olyan mátrix, melynek minden oszlopa reguláris, D pedig olyan, melyre* $\mathrm{Tr}(DC) \leq 0$. *Ekkor a*

$$DCx \oplus g \leq x$$
$$\min \mathbf{0}^T C x x^- \mathbf{0}$$

problémában a minimum értéke

$$\theta = \mathbf{0}^T C (DC)^* \mathbf{0} = ||C(DC)^*||,$$

és az összes reguláris megoldás alakja:

$$x = (\theta^{-1} \mathbf{0} \mathbf{0}^T C \oplus DC)^* u, \quad u \geq g.$$

3. Az alternáló módszer és változatai

Az $Ax \oplus c = Bx \oplus d$ alakú egyenleteket kétoldali trópusi lineáris rendszernek (TSS) nevezzük. Itt $A, B \in \mathbb{T}^{m\times n}$, valamint $c, d \in \mathbb{T}^m$. Ha $c = d = \varepsilon$, akkor homogénnek hívjuk a rendszert. Az $Ax = By$ ($A \in \mathbb{T}^{m\times n}, B \in \mathbb{T}^{m\times k}$) egyenlet egy speciális homogén rendszer. Az ilyen alakú feladatokat szeparált változójú rendszernek fogjuk nevezni. Kényelmi okokból definiáljuk a következő fogalmat.

Egy $x \in \mathbb{T}^n$ vektort nevezzünk *kvázi-valós*nak, ha $\bigoplus_{i=1}^n x_i \in \mathbb{R}$, azaz x nem a csupa ε vektor. Hasonlóan, egy mátrixot nevezzünk kvázi-valósnak, ha minden sora és oszlopa kvázi-valós.

Motivációképp mutassunk egy feladatot, amely felírható $Ax \oplus c = Bx \oplus d$ alakban [7]. Legyenek $P_1, P_2, \ldots, P_m$ termékek, melyek néhány komponensből állnak össze, amelyek elkészítéséhez n processzor áll rendelkezésünkre. Minden egyes processzor egy adott P_i termék elkészítéséhez úgy járul hozzá, hogy elkészíti az egyik komponensét. Feltesszük, hogy minden egyes processzor minden termék komponensén egyszerre tud dolgozni és ezt el is kezdi, amint bekapcsoljuk. Legyen a_{ij} azon idő, amelyre a j processzornak szüksége van a P_i termék komponensének elkészítéséhez. Jelölje x_j a j-edik processzor kezdési idejét. Tetszőleges i-re a P_i termék minden komponense legkésőbb

$$\max\{x_1 + a_{i1}, \ldots, x_n + a_{in}\}$$

időre készen lesz.
Tegyük fel, hogy az előzőektől függetlenül vannak $Q_1, Q_2, \ldots, Q_m$ termékeink is, melyeket k processzorral kell elkészítenünk. Itt legyenek a komponensek elkészítéséhez szükséges idők b_{ij}-vel jelölve, valamint legyen a j-edik processzor bekapcsolásának ideje y_j.
A *szinkronizációs probléma* lényege olyan kezdési időpontot találni mind az $n + k$ processzorhoz, amelyre az összes (P_i, Q_i) termékpár pontosan ugyanakkorra készül el. Ez a feladat a következő alakban írható:

$$\max\{x_1 + a_{i1}, \ldots, x_n + a_{in}\} = \max\{y_1 + b_{i1}, \ldots, y_k + b_{ik}\} \quad i = 1, \ldots, m.$$

Ha még azt is feltesszük, hogy P_i nem készülhet el a c_i, Q_i pedig a d_i időpont előtt, akkor így változik a feladat:

$$\max\{x_1 + a_{i1}, \ldots, x_n + a_{in}, c_i\} = \max\{y_1 + b_{i1}, \ldots, y_k + b_{ik}, d_i\} \quad i = 1, \ldots, m.$$

Ez pedig nyilván ekvivalens az

$$Ax \oplus c = Bx \oplus d$$

rendszerrel.

Hasznos lenne, ha inhomogén rendszerek mindig átalakíthatók lennének szeparált változójú rendszerré, hiszen ekkor elég lenne az utóbbival foglalkoznunk. Szerencsére ez mindig megtehető, ennek bizonyításához azonban ki kell mondanunk néhány segédállítást Butkovič könyvének [5, 7.4. fejezet] segítségével.

3.0.1. Lemma (Elhagyási törvény)**.** *Legyenek $v, w, a, b \in \mathbb{R}, a > b$. Ekkor minden x valós számra*

$$v \oplus a \otimes x = w \oplus b \otimes x \Longleftrightarrow v \oplus a \otimes x = w.$$

Bizonyítás. $(\Rightarrow)$ $v \oplus a \otimes x \geq a \otimes x > b \otimes x$, ezért $w \oplus b \otimes x = w$. Tehát $v \oplus a \otimes x = w$.
$(\Leftarrow)$ $w \geq a \otimes x > b \otimes x$, így $w = w \oplus b \otimes x$. □

Egy fontos megfigyelés, hogy az elhagyási törvény akkor is igaz marad, ha több olyan tag is van, amely tartalmaz változót.

3.0.2. Következmény. *Legyenek $a = (a_1, \ldots, a_n), b = (b_1, \ldots, b_n) \in \mathbb{R}^{1\times n}$, $c, d \in \mathbb{R}$. Ekkor minden $x = (x_1, \ldots, x_n)^T \in \mathbb{R}^n$ vektorra*

$$a \otimes x \oplus c = b \otimes x \oplus d \iff a' \otimes x' \oplus c = b' \otimes x' \oplus d,$$

ahol az a', b', x' vektorok úgy keletkeznek, hogy minden a_i, b_i, x_i elemet elhagyunk az a, b, x vektorokból, melyre $a_i < b_i$ vagy $a_i > b_i$ teljesül.

Bizonyítás. Feltehető, hogy $a_1 > b_1$. Ekkor, ha az $x_2, x_3, \ldots, x_n$ változókat fixnek képzeljük, akkor $v = a_2 \otimes x_2 \oplus \cdots \oplus a_n \otimes x_n \oplus c$, $a = a_1$, $x = x_1$, és w, b-re hasonló helyettesítésekkel következik az állítás az előző lemmából. □

3.0.3. Példa. Legyen $a = (2, 4, 4), b = (-1, 4, 4), c = -6, d = 7$. Ekkor minden $x = (x_1, x_2, x_3)$ vektorra

$$2 \otimes x_1 \oplus 4 \otimes x_2 \oplus 4 \otimes x_3 \oplus -6 = -1 \otimes x_1 \oplus 4 \otimes x_2 \oplus 4 \otimes x_3 \oplus 7$$
$$\iff$$
$$2 \otimes x_1 \oplus 4 \otimes x_2 \oplus 4 \otimes x_3 \oplus -6 = 4 \otimes x_2 \oplus 4 \otimes x_3 \oplus 7$$

Feltehetjük, hogy ha egy változó az egyik oldalon véges együtthatóval szerepel, akkor a másik oldalon is: egyszerűen írjunk ε helyett a másik oldali együtthatónál kisebb számot. Így, ha A egy sora (oszlopa) kvázi-valós, akkor feltételezhetjük, hogy B ugyanazon sora (oszlopa) is kvázi-valós és fordítva.

Vizsgáljuk meg azon esetet is, amikor létezik csupa ε sor vagy oszlop. Ekkor az előzőek szerint ezen sor (oszlop) az A és a B mátrixban is csak ε elemet tartalmaz. Ha csak ε-t tartalmazó oszlopunk van, akkor ezen oszlop törölhető mindkét mátrixból és az oszlopokhoz tartozó változó is anélkül, hogy a megoldáshalmaz lényegesen megváltozna, hiszen ezen a tagon sosem vétetne fel a maximum. Abban az esetben, ha csupa ε soraink vannak (legyen ez az i-edik), akkor a rendszernek nem létezik megoldása, ha $c_i \neq d_i$, azonban $c_i = d_i$ esetén pedig bármilyen változó kielégíti az i-edik egyenletet. Így ezen sorok törölhetők mindkét mátrixból. Azaz összességében feltehetjük, hogy A és B is kvázi-valós.

Megjegyezzük még, hogy adott trópusi lineáris egyenletrendszer átalakítható kétoldali rendszerré, ha minden egyenlet olyan, hogy az összes változó szerepel legalább az egyik oldalon. Hiszen ekkor ugyanezt a változót az adott egyenlet másik oldalára is felírhatjuk, csupán arra kell figyelnünk, hogy az együtthatója kisebb legyen.

Az előző megfigyelések után tegyük fel, hogy adott az $Ax \oplus c = Bx \oplus d$ feladat, ahol A és B kvázi-valós. Vezessünk be egy új változót, jelölje ezt x_{n+1}. Ekkor a probléma átalakítható homogén rendszerré:

$$\tilde{A} \otimes z = \tilde{B} \otimes z,$$

ahol $\tilde{A} = (A|c), \tilde{B} = (B|d)$ és $z = (z_1, z_2, \ldots, z_{n+1})^T$.
Könnyen látható, hogy az $Ax \oplus c = Bx \oplus d$ feladat pontosan akkor oldható meg, ha az $\tilde{A} \otimes z = \tilde{B} \otimes z$ rendszernek van olyan megoldása, melyre $z_{n+1} = 0$ teljesül. Nem nehéz meggondolni azt sem, hogy ha egy homogén rendszernek minden eleme véges, akkor pontosan akkor létezik nemtriviális (nem az $(\varepsilon, \ldots, \varepsilon)^T$ vektor) megoldása, ha létezik valós megoldása. Végül lássuk be az alábbi állítást.

3.0.4. Lemma. *Legyenek $A, B \in \mathbb{R}^{m\times n}$, $c, d \in \mathbb{R}^m$. Ekkor*

$$\exists x : Ax \oplus c = Bx \oplus d \iff \exists z \text{ nemtriviális vektor, melyre } \tilde{A}z = \tilde{B}z.$$

Bizonyítás. ($\Rightarrow$) Láttuk, hogy ha az $Ax \oplus c = Bx \oplus d$ rendszernek létezik megoldása, akkor az $\tilde{A}z = \tilde{B}z$ egyenletnek is van olyan megoldása, melyre $z_{n+1} = 0$. Ez egy nemtriviális megoldás is egyben.
($\Leftarrow$) Mivel A, B, c, d valósak, ezért az $\tilde{A}, \tilde{B}$ mátrixok is csak véges elemeket tartalmaznak. Ha a homogén rendszernek van nemtriviális megoldása, akkor az előzőek alapján valós megoldása is van, jelöljük ezt z-vel. Nyilván $\hat{z} = \alpha \otimes z$ is megoldás minden α esetén. Válasszuk α-t olyannak, melyre $\hat{z}_{n+1} = 0$. Ekkor az $Ax \oplus c = Bx \oplus d$ rendszernek is létezik megoldása. $\square$

Tehát inhomogén rendszerek mindig átalakíthatók szeparált változójú rendszerré a következő módon. Tudjuk, hogy az $Ax \oplus c = Bx \oplus d$ rendszert homogénné alakíthatjuk egy új változó bevezetésével. Így jutunk az $\tilde{A} \otimes z = \tilde{B} \otimes z$ egyenlethez. (Feltehető, hogy $\tilde{A}$ és $\tilde{B}$ is kvázi-valós.) Ez nyilván ekvivalens az

$$\begin{aligned} \tilde{A} \otimes x &= y, \\ \tilde{B} \otimes x &= y \end{aligned}$$

feladattal. Azaz az alábbit kapjuk:

$$\begin{pmatrix} \tilde{A} \\ \tilde{B} \end{pmatrix} \otimes x = \begin{pmatrix} I \\ I \end{pmatrix} \otimes y,$$

ami egy szeparált változójú rendszer. Így beláttuk, hogy a továbbiakban elég ilyen alakú rendszereket vizsgálnunk.

Innentől a fő célunk, hogy algoritmust adjunk szeparált változójú rendszerek megoldására, illetve, hogy az algoritmus futási idejéről is kimondjunk egy tételt. Ezekhez szükségünk lesz néhány újabb jelölésre:

- $\bar{\mathbb{T}} = \mathbb{T} \cup \{\infty\}$,
- $a \oplus' b = \min(a, b)$,
- $a \otimes' b = a + b$.

Valamint legyen $-\infty \otimes +\infty = -\infty$ és $-\infty \otimes' +\infty = +\infty$ definíció szerint. Ha A és B megfelelő méretű mátrixok, akkor $A \otimes' B$ az $A \otimes B$ szorzáshoz hasonlóan végzendő el, a $(\oplus, \otimes)$ műveletek helyett a $(\oplus', \otimes')$ műveleteket alkalmazva.

3.0.5. Definíció (Konjugált)**.** Egy $A \in \bar{\mathbb{T}}^{m\times n}$ mátrix konjugáltja az $A^\# \in \bar{\mathbb{T}}^{n\times m}$ mátrix, ahol $A^\# = -A^T$.

Az alábbi, mátrixok szorzására vonatkozó összefüggések hasznosak lesznek a későbbiekben.

3.0.6. Állítás ([8])**.** *Legyenek U, V, W megfelelő méretű mátrixok, melyeknek minden eleme $\bar{\mathbb{T}}$-beli. Ekkor*

(i) $(U \otimes' V) \otimes W \leq U \otimes' (V \otimes W)$,

(ii) $U \otimes (U^\# \otimes' W) \leq W$,

(iii) $U \otimes (U^\# \otimes' (U \otimes W)) = U \otimes W$.

A következő lemma alapvető fontosságú az algoritmus felírásához.

3.0.7. Lemma (Butkovič [5, 3.1.1. Tétel])**.** *Az $Ax \leq b$ trópusi egyenlőtlenségnek mindig létezik megoldása, melyek közül a legnagyobb: $\tilde{x} = A^\# \otimes' b$. $\tilde{x}$-ot az $Ax \leq b$ rendszer fő megoldásának nevezzük.*

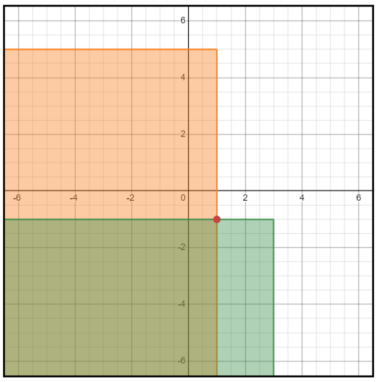
Bizonyítás. Először belátjuk, hogy $\tilde{x}$ valóban megoldás: $a_{kj} \otimes \tilde{x}_j \leq a_{kj} \otimes a_{kj}^{-1} \otimes b_k = b_k$, így $\bigoplus_j (a_{kj} \otimes \tilde{x}_j) \leq b_k$ minden k-ra. Az összes többi x megoldásra pedig $x \leq \tilde{x}$ teljesül, hiszen $a_{ij} \otimes x_j \leq b_i$, így $x_j^{-1} \geq a_{ij} \otimes b_i^{-1}$. Ekkor $x_j^{-1} \geq \max_i a_{ij} \otimes b_i^{-1}$, tehát $x_j \leq (\max_i a_{ij} \otimes b_i^{-1})^{-1} = \min_i a_{ij}^{-1} \otimes b_i = \tilde{x}_j$. $\square$

3.0.8. Megjegyzés. Legyen $\tilde{x}$ az $Ax \leq b$ rendszer fő megoldása. Ekkor $\exists x : Ax = b \Leftrightarrow A\tilde{x} = b$ és $\tilde{x} \geq \hat{x}\ \forall \hat{x} \in \{x : Ax = b\}$.

Szemléltessük egy konkrét példán is egy trópusi egyenlőtlenség-rendszer megoldásait. Legyen

$$A = \begin{pmatrix} 1 & -3 \\ -2 & 2 \end{pmatrix}, u = \begin{pmatrix} x \\ y \end{pmatrix}, b = \begin{pmatrix} 2 \\ 1 \end{pmatrix}.$$

Ábrázoljuk a megoldáshalmazt az $x - y$ síkon!



2. ábra. Az $Au \leq b$ rendszer megoldásai

A narancssárga halmaz az A mátrix első sorához, a zöld a másodikhoz tartozó egyenlőtlenség megoldásai, a piros pont pedig a rendszer fő megoldása. A közös síknegyed a teljes egyenlőtleneg-rendszer minden megoldása.

3.1. Az alternáló módszer

Az alternáló módszert Cuninghame-Green fejlesztette ki a kétezres évek elején az $A \otimes x = B \otimes y$ alakú feladatok megoldására. Ebben a részben az általa írt cikket [8] fogjuk felhasználni. Az előzőekben felírt állítások felhasználásával eljuthatunk kívánt algoritmushoz. Az input $A \in \mathbb{T}^{m \times n}, B \in \mathbb{T}^{m \times k}$ kvázi-valós mátrixok; az output pedig vagy egy (x, y) megoldáspár, vagy pedig bizonyíték arra, hogy nem létezik megoldás.

Algoritmus 2 Alternating Method [5, 7.3.1]

Legyen $x(0) \in \mathbb{R}^n$ tetszőleges
$r := 0$
loop
  $y(r) := B^{\#} \otimes' (A \otimes x(r))$
  $x(r+1) := A^{\#} \otimes' (B \otimes y(r))$
  if $x_i(r+1) < x_i(0)\ \forall i \in [n]$ **then return** 'Nincs megoldás'
  end if
  if $A \otimes x(r+1) = B \otimes y(r)$ **then return** $(x(r+1), y(r))$
  end if
  $r := r + 1$
end loop

Nézzük meg egy konkrét numerikus példán is, hogyan működik az algoritmus.

3.1.1. Példa. Legyen $A = \begin{pmatrix} 2 & 4 & 1 \\ -2 & 3 & -4 \\ -1 & 0 & 5 \end{pmatrix}, B = \begin{pmatrix} 3 & \varepsilon & -1 \\ 0 & 5 & -1 \\ \varepsilon & -2 & 4 \end{pmatrix}$. Legyen a vektor, mellyel elindítjuk az algoritmust $x(0) = \begin{pmatrix} -2 \\ 5 \\ 6 \end{pmatrix}$.

$r = 0:$

$$y(0) = \begin{pmatrix} -3 & 0 & \infty \\ \infty & 5 & 2 \\ 1 & 1 & -4 \end{pmatrix} \otimes' \left[\begin{pmatrix} 2 & 4 & 1 \\ -2 & 3 & -4 \\ -1 & 0 & 5 \end{pmatrix} \otimes \begin{pmatrix} -2 \\ 5 \\ 6 \end{pmatrix} \right] = \begin{pmatrix} 6 \\ 3 \\ 7 \end{pmatrix}$$

$$x(1) = \begin{pmatrix} -2 & 2 & 1 \\ -4 & -3 & 0 \\ -1 & 4 & -5 \end{pmatrix} \otimes' \left[\begin{pmatrix} 3 & \varepsilon & -1 \\ 0 & 5 & -1 \\ \varepsilon & -2 & 4 \end{pmatrix} \otimes \begin{pmatrix} 6 \\ 3 \\ 7 \end{pmatrix} \right] = \begin{pmatrix} 6 \\ 4 \\ 6 \end{pmatrix}$$

$$A \otimes x(1) = \begin{pmatrix} 8 \\ 7 \\ 11 \end{pmatrix} \neq \begin{pmatrix} 9 \\ 8 \\ 11 \end{pmatrix} = B \otimes y(0)$$

$r = 1:$

$$y(1) = \begin{pmatrix} -3 & 0 & \infty \\ \infty & 5 & 2 \\ 1 & 1 & -4 \end{pmatrix} \otimes' \left[\begin{pmatrix} 2 & 4 & 1 \\ -2 & 3 & -4 \\ -1 & 0 & 5 \end{pmatrix} \otimes \begin{pmatrix} 6 \\ 4 \\ 6 \end{pmatrix} \right] = \begin{pmatrix} 5 \\ 2 \\ 7 \end{pmatrix}$$

$$x(2) = \begin{pmatrix} -2 & 2 & 1 \\ -4 & -3 & 0 \\ -1 & 4 & -5 \end{pmatrix} \otimes' \left[\begin{pmatrix} 3 & \varepsilon & -1 \\ 0 & 5 & -1 \\ \varepsilon & -2 & 4 \end{pmatrix} \otimes \begin{pmatrix} 5 \\ 2 \\ 7 \end{pmatrix} \right] = \begin{pmatrix} 6 \\ 4 \\ 6 \end{pmatrix}$$

$$A \otimes x(2) = \begin{pmatrix} 8 \\ 7 \\ 11 \end{pmatrix} = B \otimes y(1)$$

Így a feladat megoldása az $(x(2), y(1))$ pár.

Lássuk be először, hogy az algoritmus helyesen működik [5, 7.3. fejezet]. Ehhez kezdésképp definiáljuk az alábbi leképezéseket:

$$\pi : y \to A^\# \otimes' (B \otimes y),$$
$$\psi : x \to B^\# \otimes' (A \otimes x).$$

Az alternáló módszer egy $(x(r), y(r))_{r=0,1,\dots}$ sorozatot gyárt, ahol $x(r+1) = \pi(y(r))$ és $y(r) = \psi(x(r))$. Egy (x, y) párt nevezzünk *stabil*nak, ha $(x, y) = (\pi(y), \psi(x))$. Ha (x, y) stabil és kielégíti az $Ax = By$ egyenletet, akkor hívjuk *stabil megoldás*nak.

3.1.2. Tétel. *Minden stabil pár egyben stabil megoldás is.*

Bizonyítás. Tegyük fel, hogy (x, y) stabil. Ekkor a 3.0.6 Állítás második pontját alkalmazva:

$$A \otimes x = A \otimes \pi(y) = A \otimes (A^\# \otimes' (B \otimes y)) \leq B \otimes y = B \otimes \psi(x) = B \otimes (B^\# \otimes' (A \otimes x)) \leq A \otimes x.$$

Következésképp minden egyenlőtlenség egyenlőséggel teljesül, azaz $A \otimes x = B \otimes y$. □

3.1.3. Tétel. *Ha az (x, y) pár megoldás, akkor $(\pi(y), \psi(x))$ stabil megoldás.*

Bizonyítás. Kihasználva a 3.0.6 Állítás harmadik pontját, valamint, hogy (x, y) megoldás: $\psi(\pi(y)) = B^{\#}\otimes'(A\otimes(A^{\#}\otimes'(B\otimes y))) = B^{\#}\otimes'(A\otimes(A^{\#}\otimes'(A\otimes x))) = B^{\#}\otimes'(A\otimes x) = \psi(x)$. Hasonlóan, $\pi(\psi(x)) = \pi(y)$, ezért $(\pi(y), \psi(x))$ stabil, így az előző tétel miatt megoldás is. □

3.1.4. Tétel. *Az $(A \otimes x(r))_{r=0,1,\ldots}$ sorozat monoton csökken.*

Bizonyítás. Felhasználva a 3.0.6 állítás második pontját, valamint azt, hogy $x(r + 1) = \pi(y(r))$ és $y(r) = \psi(x(r))$:

$$A \otimes x(r + 1) = A \otimes \pi(y(r)) = A \otimes (A^{\#} \otimes' (B \otimes y(r))) \leq B \otimes y(r) =$$
$$B \otimes (B^{\#} \otimes' (A \otimes x(r))) \leq A \otimes x(r).$$

□

3.1.5. Tétel. *Az $(x(r))_{r=1,2,\ldots}$ sorozat monoton csökken.*

Bizonyítás. Ismét a 3.0.6 állítás második pontjának segítségével:

$$x(r + 1) = \pi(y(r)) = \pi(B^{\#} \otimes' (A \otimes x(r))).$$

Az előző állítás alapján az $(A \otimes x(r))$ sorozat monoton csökken. Az erre alkalmazott fenti leképezés rendezéstartó: a $\otimes'$ művelet nyilván nem növeli az elemeket, és belátható, hogy $\otimes$ sem (ld. 4.0.2. Lemma). Azaz egy monoton csökkenő sorozatra alkalmazva monoton csökkenő sorozatot kapunk. □

3.1.6. Tétel. *Ha létezik megoldás, akkor az $(x(r))_{r=1,2,\ldots}$ sorozat alulról korlátos minden $x(0)$ esetén.*

Bizonyítás. Nyilván minden (x, y) stabil megoldásra és α valós számra a $\alpha \otimes (x, y)$ pár is stabil megoldás. Válasszunk olyan kicsi α-t, melyre $\alpha \otimes x \leq x(1)$. Így ha létezik megoldás, akkor a 3.1.3 Tétel miatt létezik (u, v) stabil megoldás is, melyre $u \leq x(1)$. Ha pedig $u \leq x(r)$ valamely r-re, akkor

$$x(r + 1) = (\pi \circ \psi)(x(r)) \geq (\pi \circ \psi)(u) = \pi(v) = u.$$

Indukcióval következik a tétel. □

3.1.7. Tétel. *Ha valamely r-re $x_i(r) < x_i(0)$ minden $i \in [n]$ esetén, akkor az $A\otimes x = B\otimes y$ egyenletnek nincs megoldása.*

Bizonyítás. Az előző tétel bizonyításában α választható olyannak, hogy $\alpha \otimes x \leq x(0)$ és legalább egy komponensben egyenlőség teljesül. A 3.1.5 és 3.1.6 tételek miatt ezen komponens nem változik. Így, ha $x(r)$ minden eleme kisebb $x(0)$ megfelelő eleménél, akkor egy kezdeti komponens sem maradt állandó, azaz nem létezhet megoldása a feladatnak. □

3.1.8. Tétel. *Az* $(x(r), y(r))_{r=0,1,\dots}$ *sorozat pontosan akkor konvergens, ha létezik megoldása a feladatnak. Ekkor a konvergencia monoton és egy stabil megoldás a határérték bármely kezdeti* $x(0) \in \mathbb{R}^n$ *esetén.*

Bizonyítás. ($\Rightarrow$) Ha $(x(r), y(r)) \to (\xi, \eta)$, akkor

$$(\xi, \eta) = \lim(x(r+1), y(r)) = \lim(\pi(y(r)), \psi(x(r))) = (\pi(\eta), \psi(\xi)),$$

ahol kihasználtuk a 3.1.6 tételt és a folytonosságot. Így (ξ, η) stabil és a 3.1.2 tétel miatt stabil megoldás is.
($\Leftarrow$) Ha létezik megoldás, akkor $(x(r))$ monoton konvergens a 3.1.5 és 3.1.6 tételek miatt. Hasonlóan $(y(r))$ is. $\square$

Beláttuk tehát, hogy az algoritmus helyes. Hátra van még, hogy a futási idővel kapcsolatban is kimondjunk egy eredményt.

Észrevehetjük, hogy ha A és B egészek, kvázi-valósak és $x(0)$-t is egésznek választjuk, akkor az egészértékűség megőrződik az algoritmus futása során. Azaz ebben az esetben pontosan akkor létezik megoldása az $A \otimes x = B \otimes y$ egyenletnek, ha létezik egész megoldása is. Mivel az $(x(r), y(r))$ sorozat monoton csökkenő, alulról korlátos és egész, ezért az alternáló módszer véges sok lépésben megtalál egy egész megoldást, ha létezik.

3.1.9. Megjegyzés. Mivel most elsősorban az alkalmazásokra fókuszálunk, ezért feltehetjük, hogy a mátrixok és vektorok elemei racionális számok, hiszen valós számokkal úgysem tudunk numerikusan számolni. Ekkor viszont felszorozhatunk az elemek nevezőinek legkisebb közös többszörösével, így egész számokat kapva mindenhol. Persze ilyenkor a megoldáshalmaz nem változik: $A \otimes x = B \otimes y \iff \alpha A \otimes \alpha x = \alpha B \otimes \alpha y$.

A futási időt is a következő, speciális esetben fogjuk vizsgálni: $A, B, x(0)$ egészek, A és B közül legalább az egyik valós (feltehető, hogy ez a mátrix A) és a másik kvázi-valós.

3.1.10. Tétel (Butkovič [5, 7.3.11. Tétel])**.** *Ha* $A \in \mathbb{Z}^{m\times n}$, $B \in (\mathbb{Z} \cup \{-\infty\})^{m\times k}$ *kvázi-valós és az alternáló módszert az* $x(0) \in \mathbb{Z}^n$ *vektorral indítjuk, akkor legfeljebb*

$$(n-1)(1 + \gamma^\# \otimes A^\# \otimes A \otimes \gamma)$$

iteráció után leáll az algoritmus, ahol $\gamma = x(0)$.

Bizonyítás. Tegyük fel, hogy létezik megoldás. A 3.1.7 Tétel miatt van olyan eleme γ-nak (legyen ez γ_j), amely nem változik az algoritmus futása során. Egy ilyen komponenst nevezzünk *alvó*nak, különben nevezzük *éber*nek. Tudjuk, hogy amint $A \otimes x$ értéke nem változik, az algoritmus leáll. Ez legkésőbb akkor következik be, amikor x komponensei annyira lecsökkentek már γ_j-hez képest, hogy nem befolyásolják $A \otimes x$ értékét. Azaz $a_{ik} + x_k \le a_{ij} + \gamma_j$ minden k, i-re. Az u_{kj} jelölés bevezetésével ezt így írhatjuk:

$$u_{kj} = \min_i (a_{ij} + \gamma_j - a_{ik}) = (A^\# \otimes' A)_{kj} + \gamma_j.$$

Ez tehát jelenti, hogy az éber x_k-t dominálja az alvó γ_j. Mivel x_k értéke monoton csökken, ez a dominancia megmarad az algoritmus hátralevő részében. Sajnos j értékét nem ismerjük, azonban a dominanciát biztosítani tudjuk, ha minden komponenst potenciális alvónak tekintünk. Azaz x_k biztosan dominálva van, ha az értéke kisebb, mint

$$\beta_k = \min_j u_{kj} = \min_j((A^\# \otimes' A)_{kj} + \gamma_j) = (A^\# \otimes' A \otimes' \gamma)_k.$$

Így x_k legfeljebb

$$w_k = \gamma_k - \beta_k + 1$$

-gyel csökken a dominálás előtt. Az éber komponensek száma legfeljebb $n-1$, és minden iterációban legalább egy éber komponens értéke legalább eggyel csökken (különben $A \otimes x$ értéke nem változna és az algoritmus megállna). Tehát az iterációk száma a dominálás előtt legfeljebb

$$(n-1)\max_k w_k = (n-1)\max_k(\gamma_k - \beta_k + 1) = (n-1)(1+\beta^\# \otimes \gamma) = (n-1)(1+\gamma^\# \otimes A^\# \otimes A \otimes \gamma).$$

Ha nem létezik megoldás, akkor legfeljebb ennyi iteráció után x összes komponense lecsökken, ezt az algoritmus jelzi és megáll. □

Kérdés lehet még, hogy bár az algoritmus bármilyen kezdeti $x(0)$ vektorral futtatható, hogyan lenne érdemes megválasztanunk ahhoz, hogy a futásidő minél rövidebb legyen. Ennek megválaszolásához tekintsük az alábbiakat.
Ha $A \in \mathbb{T}^{n\times n}$ és $\lambda \in \mathbb{T}$, jelölje a véges szubsajátvektorok halmazát:

$$V^*(A,\lambda) = \{x \in \mathbb{R}^n : A \otimes x \leq \lambda \otimes x\}.$$

Valamint $V^*(A) \coloneqq V^*(A, \lambda(A))$.

3.1.11. Lemma (Butkovič [5, 1.6.28. Lemma])**.** *Legyen $A \in \mathbb{T}^{n\times n}$, $\lambda(A) > \varepsilon$. Ha $z \in V^*(A)$, akkor*

$$\lambda(A) = z^\# \otimes A \otimes z = \min_{x\in\mathbb{R}^n} x^\# \otimes A \otimes x.$$

Az előző lemmából és a 3.1.10 tételből következik, hogy az $A^\# \otimes A$ mátrix egy véges szubsajátvektora jó választás lesz kezdeti $x(0)$ vektornak. (Mivel A egész, így valójában $A^\# \otimes A$ minden szubsajátvektora véges lesz.) Ha így választjuk $x(0)$-t tehát, akkor az iterációk száma legfeljebb

$$(n-1)(1 + \lambda(A^\# \otimes A)).$$

Legyen $K(A) = \max\{|a_{ij}| : i \in [m], j \in [n]\}$ minden $A \in \mathbb{R}^{n\times n}$ mátrixra. Ekkor $|\lambda(A)| \leq K(A)$. Ezt könnyen beláthatjuk: ha $\frac{a_{i_1 i_2}+\cdots+a_{i_k i_1}}{k}$ a maximális körátlag, akkor

$$|\lambda(A)| = |\tfrac{a_{i_1 i_2}+\cdots+a_{i_k i_1}}{k}| \leq \tfrac{|a_{i_1 i_2}|}{k} + \cdots + \tfrac{|a_{i_k i_1}|}{k} \leq \tfrac{K(A)}{k} + \cdots + \tfrac{K(A)}{k} = k\tfrac{K(A)}{k} = K(A).$$

Így $0 \leq \lambda(A^{\#} \otimes A) \leq 2K(A)$.

Az alternáló módszerben $y(r)$ és $x(r+1)$ kiszámítása $(mn+mk)$ lépés. Az első **if** feltétel ellenőrzése pedig n lépés, míg a másodiké $(mn+mk+m)$. Minden eddigit felhasználva az alternáló módszer futási ideje:

$$(n-1)(1+2K(A))\mathcal{O}(m(n+k)) = \mathcal{O}(mn(n+k)K(A)).$$

Összességében megfogalmazhatjuk tehát a következő tételt.

3.1.12. Tétel (Butkovič [5, 7.3.12. Tétel])**.** *Az alternáló módszer pszeudopolinomiális, ha A és B egész mátrixok, melyek közül az egyik valós, a másik kvázi-valós.*

Végül nézzük meg az alternáló módszer futási idejét a szinkronizációs problémára, ha A, B, c és d is csak véges egész elemeket tartalmaz. Ekkor az $\begin{pmatrix} \tilde{A} \\ \tilde{B} \end{pmatrix}$ mátrix véges, az $\begin{pmatrix} I \\ I \end{pmatrix}$ mátrix pedig kvázi-valós. Ha K-val jelöljük $K(\tilde{A}|\tilde{B})$-t, akkor a futásidő

$$\mathcal{O}(2mn(n+m)K) = \mathcal{O}(mn(m+n)K),$$

ahol $2m$ az egyenletek és $n+m$ a változók száma. Azaz egy véges, egész mátrixok alkotta homogén rendszer megoldható az alternáló módszerrel pszeudopolinomiális időben.

3.2. Egészértékű megoldások

A szinkronizációs problémában megkövetelhetjük, hogy minden kezdési idő diszkrét érték legyen, így ebben az esetben a TSS-nek az egész megoldásait keressük. Mutassunk először algoritmust [17] felhasználásával a szeparált változójú rendszerek egész megoldásainak megtalálására. Itt ismét feltesszük, hogy A és B kvázi-valósak. Az algoritmus valójában az alternáló módszer, apró változtatással.

Az input: $A \in \mathbb{T}^{m\times n}, B \in \mathbb{T}^{m\times k}$ kvázi véges mátrixok és az $x(0) \in \mathbb{Z}^n$ vektor. Az output: (x, y) egész megoldáspár az $Ax = By$ feladatra, vagy bizonyíték, hogy ilyen nem létezik.

Algoritmus 3 SEP-INT-TSS [17]

$r := 0$
loop
    $y(r) := \lfloor B^{\#} \otimes' (A \otimes x(r)) \rfloor$
    $x(r+1) := \lfloor A^{\#} \otimes' (B \otimes y(r)) \rfloor$
    if $x_i(r+1) < x_i(0)\ \forall i \in [n]$ **then return** 'Nincs megoldás'
    end if
    if $A \otimes x(r+1) = B \otimes y(r)$ **then return** $(x(r+1), y(r))$
    end if
    $r := r+1$
end loop

3.2.1. Tétel ([17])**.** *Az algoritmus helyes és, ha A véges, akkor legfeljebb*

$$(n-1)(1+\gamma^{\#}\otimes A^{\#}\otimes A\otimes\gamma)$$

iteráció után leáll. Az algoritmus futási ideje

$$\mathcal{O}(mn(n+k)K(A)),$$

ahol $K(A) = \lceil \max\{|a_{ij}| : i \in [m], j \in [n]\} \rceil$.

Bizonyítás. A tétel belátható az alternáló módszernél látott állítások és bizonyítások kissé módosított változataival [17]. □

A fejezet elején láttuk, hogy bármely TSS átalakítható homogén rendszerré. Így szeretnénk algoritmust adni a homogén rendszerek megoldására is. A korábbiakban látott gondolatokat használva az alábbira juthatunk:
Input: $A', B' \in \mathbb{T}^{m\times n}$ kvázi-valósak, $I \in \mathbb{T}^{m\times m}$, és az $x(0) \in \mathbb{Z}^n$ kezdővektor.
Output: Az $A'x = B'x$ rendszer $x \in \mathbb{Z}^n$ megoldása, vagy bizonyíték, hogy nem létezik ilyen.

Algoritmus 4 GEN-INT-TSS [17]

$r := 0, A := \begin{pmatrix} A' \\ B' \end{pmatrix}, B := \begin{pmatrix} I \\ I \end{pmatrix}$
loop
    $y(r) := B^{\#} \otimes' (A \otimes x(r))$
    $x(r+1) := \lfloor A^{\#} \otimes' (B \otimes y(r)) \rfloor$
    if $x_i(r+1) < x_i(0)\ \forall i \in [n]$ **then return** 'Nincs megoldás'
    end if
    if $A \otimes x(r+1) = B \otimes y(r)$ **then return** $(x(r+1), y(r))$
    end if
    $r := r+1$
end loop

A GEN-INT-TSS algoritmus valójában megegyezik a SEP-INT-TSS algoritmussal, azzal a különbséggel, hogy először el kell készítenünk két új mátrixot az inputból, illetve $y(r)$ kiszámításánál nem veszünk alsó egészrészt, mivel csak x-től követeljük meg, hogy egész vektor legyen.

A következő tétel bizonyítása ismét nagyon hasonló az alternáló módszer és SEP-INT-TSS algoritmusoknál kimondott állításokéhoz [17].

3.2.2. Tétel ([17])**.** *Az algoritmus helyes, és legfeljebb*

$$(n-1)(1+\gamma^{\#}\otimes A^{\#}\otimes A\otimes\gamma)$$

iteráció után leáll, ahol $A = \begin{pmatrix} A' \\ B' \end{pmatrix}$ *véges. Az algoritmus futási ideje*

$$\mathcal{O}((mn(n+k))K(A'|B')).$$

4. Maxlineáris programozás

Ebben a fejezetben maxlineáris függvényeket fogunk maximalizálni, illetve minimalizálni különböző trópusi lineáris feltételek mellett. Ehhez ismét segítségünkre lesz a Butkovič által írt átfogó könyv [5, 10. fejezet]. Milyen függvényeket hívunk maxlineárisnak?

4.0.1. Definíció (Maxlineáris függvény)**.** Ha $f \in \mathbb{T}^n$, akkor az $f(x) = f^T \otimes x$ ($f : \mathbb{T}^n \to \mathbb{T}$) függvényt maxlineáris függvénynek nevezzük.

Fontos megjegyezni, hogy míg a klasszikus lineáris programozási feladatoknál a maximalizálás egyszerűen lecserélhető minimalizálásra a célfüggvény negálásával, addig ez maxlineáris függvények esetén nem mindig tehető meg. Legyen például $f = (1, \varepsilon, -4)^T$. Ekkor $f(x)$ maxlineáris, de $-f(x)$ nem. Így maxlineáris célfüggvények esetén külön algoritmust kell majd készítenünk attól függően, hogy éppen maximalizálni vagy minimalizálni szeretnénk. Említsük meg a maxlineáris függvények két hasznos tulajdonságát.

4.0.2. Lemma (Butkovič [5, 10.0.7. Lemma])**.** *Legyen $f(x) = f^T \otimes x$ maxlineáris függvény. Ekkor*

(i) $f(x)$ max-homogén és max-additív, azaz $f(\alpha \otimes x \oplus \beta \otimes y) = \alpha \otimes f(x) \oplus \beta \otimes f(y)$, minden $x, y \in \mathbb{T}^n$ és $\alpha, \beta \in \mathbb{T}$ esetén.

(ii) $f(x)$ izotón, azaz $f(x) \leq f(y)$ minden $x, y \in \mathbb{T}^n$, $x \leq y$-ra.

Bizonyítás. (i) $f(\alpha \otimes x \oplus \beta \otimes y) = f^T \otimes (\alpha \otimes x \oplus \beta \otimes y) = f^T \otimes \alpha \otimes x \oplus f^T \otimes \beta \otimes y = \alpha \otimes f^T \otimes x \oplus \beta \otimes f^T \otimes y = \alpha \otimes f(x) \oplus \beta \otimes f(y)$.
(ii) $f(x) = f^T \otimes x \leq f^T \otimes y = f(y)$. □

A fejezet során végig feltesszük, hogy $f \in \mathbb{R}^n$.

4.1. Egyoldali feltételek

Egyoldali rendszernek az $A \otimes x = b$ alakú egyenleteket nevezzük. Vizsgáljuk meg, hogyan találhatunk optimális megoldást a maxlineáris célfüggvényünkre nézve ilyen feltételek mellett [5, 10.1. fejezet].

4.1.1. Probléma. $A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m, f \in \mathbb{R}^n$

$$\begin{gathered} A \otimes x = b \\ f(x) \to \min / \max \end{gathered}$$

Az első fejezetben használt jelölésekhez hasonlóan legyen a rendszer megoldáshalmaza $S = \{x \in \mathbb{R}^n : A \otimes x = b\}$, illetve legyenek $\bar{x}_j = \min_{i \in M} b_i \otimes a_{ij}^{-1}$, $\bar{x} = (\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n)^T$. Elevenítsük fel azt is (1.0.16 Tétel), hogy $x \leq \bar{x}$ minden $x \in S$ esetén, valamint $x \in S$ pontosan akkor, ha $x \leq \bar{x}$ és $\bigcup_{j \in N_x} M_j = M$.
A 4.1.1 probléma azon változatát, melyben minimalizálni [maximalizálni] szeretnénk, jelöle $\mathrm{MLP}_1^{\min}$ [$\mathrm{MLP}_1^{\max}$], az optimális megoldások halmazát pedig $S_1^{\min}$ [$S_1^{\max}$].

4.1.2. Lemma. *Ha $S \neq \emptyset$, akkor $\bar{x} \in S_1^{\max}$.*

Bizonyítás. Legyen $\hat{x} \in S$. Ekkor az 1.0.16 tételből következik, hogy $\hat{x} \leq \bar{x}$. Az $f(x)$ maxlineáris függvény izotonitása miatt $f(\hat{x}) \leq f(\bar{x})$. $\square$

Az előző lemma miatt elég az $\mathrm{MLP}_1^{\min}$ feladatra algoritmust mutatnunk. Az input az $A \in \mathbb{R}^{m\times n}$ mátrix és a $b \in \mathbb{R}^m, f \in \mathbb{R}^n$ vektorok. Az output: $x \in S_1^{\min}$.

Algoritmus 5 ONEMAXLINMIN [3]

1: Számoljuk ki az $\bar{x}$ vektort és az M_j $(j \in [n])$ halmazokat.
2: Rendezzük sorba az $f_j \otimes \bar{x}_j$ $(j \in [n])$ értékeket. Feltehető, hogy

$$f_1 \otimes \bar{x}_1 \leq f_2 \otimes \bar{x}_2 \leq \cdots \leq f_n \otimes \bar{x}_n.$$

3: $J \coloneqq \{1\}, r = 1.$
4: **if** $\bigcup_{j\in J} M_j = M$ **then**
5:     STOP ($x_j = \bar{x}_j$, ha $j \in J$; különben legyen x_j elég kicsi).
6: **end if**
7: $r \coloneqq r+1, J \coloneqq J \cup \{r\}.$
8: Goto 4.

4.1.3. Megjegyzés. Az 5. programsorban az "elég kicsi" jelentheti például a következőt: $x_j \leq f_j^{-1} \otimes f_r \otimes \bar{x}_r$.

Ismét nézzük meg egy példán, hogyan működik az algoritmus.

4.1.4. Példa. Legyenek $A = \begin{pmatrix} -2 & 3 & 1 \\ 0 & -1 & 4 \\ -4 & 2 & 0 \end{pmatrix}, b = \begin{pmatrix} 5 \\ 3 \\ 4 \end{pmatrix}, f = \begin{pmatrix} 4 \\ -2 \\ 1 \end{pmatrix}$.

$\bar{x} = (3, 2, -1)^T, M_1 = \{2\}, M_2 = \{1, 3\}, M_3 = \{2\}$

$f_1 \otimes \bar{x}_1 = 7, f_2 \otimes \bar{x}_2 = 0, f_3 \otimes \bar{x}_3 = 0 \Rightarrow f_3 \otimes \bar{x}_3 \leq f_2 \otimes \bar{x}_2 \leq f_1 \otimes \bar{x}_2$

$J = \{3\}, r = 3$:

$$M_3 = \{2\} \neq \{1, 2, 3\} = M$$

$r = 2, J = J \cup \{2\} = \{3, 2\}$:

$$M_3 \cup M_2 = \{1, 2, 3\} = M$$
$$x \coloneqq (f_1^{-1} \otimes f_2 \otimes \bar{x}_2, \bar{x}_2, \bar{x}_3)^T = (-4, 2, -1)^T$$

4.1.5. Tétel. *A ONEMAXLINMIN algoritmus helyes és a futási ideje $\mathcal{O}(mn^2)$.*

Bizonyítás. Az algoritmus helyes, hiszen az 1.0.16 tételben szereplő ekvivalencia jobb oldala szerint késztjük el az x megoldást; illetve ezen x tényleg minimalizálja a célfüggvényt, mert azon a lehető legkisebb $f_i \otimes x_i$ értéken fog felvétetni a maximum, amelyre $x_i = \bar{x}_i$ még muszáj teljesüljön ahhoz, hogy x megoldás legyen. A futási idő pedig így számolható: az első programsor $\mathcal{O}(mn)$, a második $\mathcal{O}(n \log n)$ lépés, a 4-8. programsorok legfeljebb n-szer futnak le (mivel x hossza n), és minden egyes futásuk $\mathcal{O}(mn)$ ideig tart. $\square$

4.2. Kétoldali feltételek

Ismét maxlineáris $f(x)$ célfüggvényekre nézve szeretnénk optimális $x \in \mathbb{R}^n$ megoldást találni, azonban a feltételünk most kétoldali, $A \otimes x \oplus c = B \otimes x \oplus d$ alakú [5, 10.2. fejezet].

4.2.1. Probléma. $A, B \in \mathbb{R}^{m \times n}$, $c, d \in \mathbb{R}^m$, $f \in \mathbb{R}^n$

$$A \otimes x \oplus c = B \otimes x \oplus d$$
$$f(x) \to \min / \max$$

Jelölje az $f(x)$ függvény minimalizálásának [maximalizálásának] problémáját $\mathrm{MLP}^{\min}$ [$\mathrm{MLP}^{\max}$]. Legyen a kétoldali rendszer megoldáshalmaza $S = \{x \in \mathbb{R}^n : A \otimes x \oplus c = B \otimes x \oplus d\}$, valamint $S^{\min} := \{x \in S : f(x) \leq f(z) \quad \forall z \in S\}$ és $S^{\max} := \{x \in S : f(x) \geq f(z) \quad \forall z \in S\}$.

Láttuk korábban, hogy egész elemekből álló kétoldali rendszerről el tudjuk dönteni, hogy létezik-e megoldása pszeudopolinomiális időben az alternáló módszerrel. Ennek felhasználásával készítünk intervallumfelező módszereket az $\mathrm{MLP}^{\min}$ és $\mathrm{MLP}^{\max}$ feladatok megoldására.

4.2.2. Lemma. *Legyen $\alpha, \alpha' \in \mathbb{R}$, $\alpha' < \alpha$ és $f(x) = f^T \otimes x$, $f'(x) = f'^T \otimes x$, ahol $f_j < f'_j$ minden $j \in [n]$-re. Ekkor minden $x \in \mathbb{R}^n$ esetén*

$$f(x) = \alpha \iff f(x) \oplus \alpha' = f'(x) \oplus \alpha.$$

Bizonyítás. $(\Rightarrow)$ $f^T \otimes x = \alpha$, kihasználva, hogy $\alpha' < \alpha$ kapjuk, hogy $f^T \otimes x \oplus \alpha' = \alpha$, és az elhagyási törvény miatt $f^T \otimes x \oplus \alpha' = f'^T \otimes x \oplus \alpha$.
$(\Leftarrow)$ $f(x) \oplus \alpha' = f'(x) \oplus \alpha$, az elhagyási törvényt alkalmazva $f(x) \oplus \alpha' = \alpha$, mivel $\alpha' < \alpha$, így $f(x) = \alpha$. □

A lemma fontos következménye, hogy az, hogy az $f(x)$ függvény valamilyen α értéket elér, ekvivalens egy maxlineáris egyenletrendszer megoldhatóságával. Pontosabban:

4.2.3. Tétel. *Pontosan akkor teljesül $f(x) = \alpha$ valamely $x \in S$-re, ha az alábbi inhomogén trópusi egyenletrendszernek van megoldása:*

$$A \otimes x \oplus c = B \otimes x \oplus d$$
$$f(x) \oplus \alpha' = f'(x) \oplus \alpha,$$

ahol $\alpha' < \alpha$ és $f'_j < f_j$ minden $j \in [n]$ esetén.

A tételből következik, hogy ha a problémában csak egész számok szerepelnek, akkor a célfüggvény pontosan akkor vesz fel egy egész értéket egy valós megoldással, ha azt egy egész megoldással is felveszi.

Ahhoz, hogy intervallumfelező módszert tudjunk kifejleszteni szükség van a következő állításra.

4.2.4. Állítás. *Ha $x, y \in S$, $f(x) = \alpha < \beta = f(y)$, akkor minden $\gamma \in (\alpha, \beta)$ esetén létezik $z \in S$, melyre $f(z) = \gamma$.*

Bizonyítás. Legyen $\lambda = 0, \mu = \beta^{-1} \otimes \gamma, z = \lambda \otimes x \oplus \mu \otimes y$. Mivel $\gamma \in (\alpha, \beta)$, ezért $\lambda \oplus \mu = 0$. $z \in S$, hiszen:

$$\begin{gathered} Az \oplus c = A(\lambda x \oplus \mu y) \oplus c = A(\lambda x \oplus \mu y) \oplus (\lambda \oplus \mu)c = A(\lambda x \oplus \mu y) \oplus \lambda c \oplus \mu c = \\ \lambda(Ax) \oplus \mu(Ay) \oplus \lambda c \oplus \mu c = \lambda(Ax \oplus c) \oplus \mu(Ay \oplus c) = \lambda(Bx \oplus d) \oplus \mu(By \oplus d) = \\ B(\lambda x \oplus \mu y) \oplus \lambda d \oplus \mu d = B(\lambda x \oplus \mu y) \oplus (\lambda \oplus \mu)d = B(\lambda x \oplus \mu y) \oplus d = Bz \oplus d, \end{gathered}$$

ahol az egymás után írás trópusi szorzást jelöl. Illetve $f(z) = \gamma$ is teljesül:

$$f(z) = f(\lambda \otimes x \oplus \mu \otimes y) = \lambda \otimes f(x) \oplus \mu \otimes f(y) = 0 \otimes \alpha \oplus \beta^{-1} \otimes \gamma \otimes \beta = \alpha \oplus \beta^{-1} \otimes \gamma \otimes \beta = \gamma,$$

ahol kihasználtuk az $f(x)$ függvény max-homogenitását és max-additivitását, illetve azt, hogy $\gamma \in (\alpha, \beta)$. □

Mondjunk ki néhány feltételt az optimális megoldások létezésével kapcsolatban. Legyen $f^{\min} = \inf_{x \in S} f(x)$ és $f^{\max} = \sup_{x \in S} f(x)$. Foglalkozzunk először az alsó korláttal. Feltehető, hogy a kétoldali rendszerben $c \geq d$, mert $c < d$ esetben c értékét d-re növelve a megoldáshalmaz nem változna. Legyen $M^{>} = \{i \in [m] : c_i > d_i\}$. Adott $r \in M^{>}$-re legyen $L_r = \min_{k \in [n]} f_k \otimes c_r \otimes b_{rk}^{-1}$, valamint $L := \max_{r \in M^{>}} L_r$. Lássuk be, hogy $c \geq d$ esetben L a célfüggvény alsó korlátja.

4.2.5. Lemma. *Ha $c \geq d$, akkor $f(x) \geq L$ minden $x \in S$ megoldásra.*

Bizonyítás. Ha $M^{>} = \emptyset$, akkor az állítás igaz, hiszen $L = -\infty$. Különben legyen $x \in S$, $r \in M^{>}$. Ekkor

$$(B \otimes x)_r \geq c_r,$$

mivel $(B \otimes x)_r = c_r$, ha a kétoldali rendszer bal oldalán c_r-en vétetik fel a maximum, illetve $(B \otimes x)_r > c_r$, ha $(A \otimes x)_r$-en. Ezért $x_k \geq c_r \otimes b_{rk}^{-1}$ valamely $k \in [n]$-re. Így $f(x) \geq f_k \otimes x_k \geq f_k \otimes c_r \otimes b_{rk}^{-1} \geq L_r$. □

A lemma segítségével megfogalmazhatunk egy szükséges és elégséges feltételt $f^{\min}$ korlátosságához.

4.2.6. Tétel. *$f^{\min} = -\infty$ pontosan akkor, ha $c = d$.*

Bizonyítás. Ha $c = d$, akkor $\alpha \otimes x$ megoldás minden $x \in \mathbb{R}^n$ esetén, ha $\alpha < 0$ elegendően kicsi. Így, ha $\alpha \to -\infty$, akkor $f(\alpha \otimes x) = \alpha \otimes f(x) \to -\infty$.
Ha $c \neq d$, akkor ismét feltehetjük, hogy $c \geq d$, és az előző lemmából következik az állítás ($L > -\infty$, mivel $c \neq d$). □

Vizsgáljuk meg az $f^{\max}$ függvény korlátosságát is.

4.2.7. Lemma. *Legyen $c \geq d$. Ha $x \in S$ és $(A \otimes x)_i > c_i$ minden $i \in [m]$-re, akkor $x' = \alpha \otimes x$ is megoldás és $(A \otimes x')_i = c_i$ néhány $i \in [m]$-re, ahol $\alpha = \max_{i \in [m]}(c_i \otimes (A \otimes x)_i^{-1})$.*

Bizonyítás. Legyen $x \in S$. Mivel $(A \otimes x)_i > c_i$ minden i-re és $c \geq d$, ezért $A \otimes x = B \otimes y$. Ekkor minden α valós számra $\alpha \otimes x$ is megoldás.

$$(A \otimes (\alpha \otimes x))_i = \alpha \otimes (A \otimes x)_i = [\max_{j \in [m]}(c_j \otimes (A \otimes x)_j^{-1})] \otimes (A \otimes x)_i \geq c_i$$

minden $i \in [m]$-re, és legalább egy i-re egyenlőséggel teljesül. □

Legyen $U = \max_{r \in [m]} \max_{j \in [n]} f_j \otimes a_{rj}^{-1} \otimes c_r$. Bizonyítsuk be, hogy U jó felső korlát lehet az $f(x)$ függvényre.

4.2.8. Lemma. *Ha $c \geq d$, akkor*

(i) Ha $x \in S$ és $(A \otimes x)_r \leq c_r$ néhány $r \in [m]$-re akkor $f(x) \leq U$.

(ii) Ha az $A \otimes x = B \otimes x$ egyenletnek nincs nemtriviális megoldása, akkor $f(x) \leq U$ minden $x \in S$ megoldásra.

Bizonyítás. (i) Minden $j \in [n]$-re $a_{rj} \otimes x_j \leq c_j$, ezért

$$f(x) \leq \max_{j \in [n]} f_j \otimes a_{rj}^{-1} \otimes c_r \leq U.$$

(ii) Ha $S = \emptyset$, akkor $f^{\max} = -\infty \leq U$. Ha $S \neq \emptyset$, akkor legyen $x \in S$. Ekkor $(A \otimes x)_r \leq c_r$ néhány $r \in [m]$-re (különben $A \otimes x = B \otimes x$ teljesülne), így az állítás következik az (i) pontból. □

4.2.9. Tétel. *$f^{\max} = +\infty$ pontosan akkor, ha az $A \otimes x = B \otimes x$ rendszernek létezik nemtriviális megoldása.*

Bizonyítás. Feltehető, hogy $c \geq d$. Ha nem létezik nemtriviális x, melyre $A \otimes x = B \otimes x$, akkor az előző lemma miatt készen vagyunk. Különben legyen $z \neq \varepsilon$ megoldása az $A \otimes x = B \otimes x$ egyenletnek. Ekkor minden elegendően nagy α-ra:

$$A \otimes (\alpha \otimes z) = B \otimes (\alpha \otimes z) \geq c \oplus d,$$

vagyis $\alpha \otimes z \in S$. Az állítás következik, ha $\alpha \to +\infty$. □

Megmutatjuk, hogy ha $S \neq \emptyset$ és $f^{\min}$ $[f^{\max}]$ véges, akkor az optimális érték tényleg elérhető. Ha f folytonos függvény, akkor elég belátnunk, hogy az S halmaz megszorítható egy kompakt részhalmazára.

4.2.10. Állítás. *f folytonos minden $x \in \mathbb{R}^n$ pontban.*

Bizonyítás. Legyen $\xi > 0$ tetszőleges, $x' \in \mathbb{R}^n$. Ekkor $|f(x) - f(x')| = |\max_i(f_i + x_i) - \max_i(f_i + x_i')| =: |\max y_i - \max y_i'| \leq |y_1 - y_1'| + \cdots + |y_n - y_n'| = ||y - y'||_1 \leq M||y - y'||_2 = M||x - x'||_2 < \eta := \xi$, ahol kihasználtuk, hogy véges dimenziós terekben a különböző p-normák ekvivalensek ($M \in \mathbb{R}$ megfelelő konstans), illetve azt, hogy $||x - x'||_2 = ||y - y'||_2$, hiszen az f vektorral való trópusi szorzás geometriailag egy eltolást jelent. □

Vezessük be az alábbi jelöléseket:

- $h_j = \min\{\min_{r\in[m]} a_{rj}^{-1} \otimes c_j, \min_{r\in[m]} b_{rj}^{-1} \otimes d_j, f_j^{-1} \otimes L\} \quad j = 1, \ldots, n$

- $h'_j = \min\{\min_{r\in[m]} a_{rj}^{-1} \otimes c_j, \min_{r\in[m]} b_{rj}^{-1} \otimes d_j\} \quad j = 1, \ldots, n$

és $h \coloneqq (h_1, \ldots, h_n)^T$, $h' \coloneqq (h'_1, \ldots, h'_n)^T$. A h' vektor véges, hiszen A, B, c és d elemei valósak, a h vektor pedig pontosan akkor véges, ha $c \neq d$ (azaz az $M^{>}$ halmaz nemüres). Először mutassuk meg az $f^{\min}$ érték elérhetőségét.

4.2.11. Állítás. *$\forall x \in S\ \exists x' \in S : x' \geq h$ és $f(x) = f'(x)$.*

Bizonyítás. Legyen $x \in S$. $x' = x \oplus h$ jó választás lesz, hiszen ha $x_j < h_j$ valamely j koordinátára, akkor a maximum a rendszer egyik oldalán sem és a célfüggvényben sem x_j-n vétetik fel, így x_j-t h_j-re cserélve nem sértünk meg egy egyenletet sem és a célfüggvény értéke sem változik. □

4.2.12. Következmény. *Ha $f^{\min}$ véges és $S \neq \emptyset$, akkor létezik $\bar{S}$ kompakt halmaz, melyre $f^{\min} = \min_{x\in\bar{S}} f(x)$.*

Bizonyítás. Mivel $f^{\min}$ véges, ezért $c \neq d$, azaz a h vektor is véges. Az előző állítás szerint létezik $\tilde{x} \in S$, $\tilde{x} \geq h$. Ekkor az

$$\bar{S} = S \cap \{x \in \mathbb{R}^n : h_j \leq x_j \leq f_j^{-1} \otimes f(\tilde{x}_j), j \in [n]\}$$

halmaz kompakt részhalmaza S-nek és $\tilde{x} \in S$. Indirekt tegyük fel, hogy létezik $y \in S$, melyre $f(y) \leq \min_{x\in\bar{S}} f(x) \leq f(\tilde{x})$. Ekkor az előbbi állítás miatt létezik $y' \geq h, y' \in S, f(y') = f(y)$. Így

$$f_j \otimes y'_j \leq f(y') = f(y) \leq f(\tilde{x})$$

minden $j \in [n]$-re, ezért $y' \in \bar{S}, f(y') < \min_{x\in\bar{S}} f(x)$, ami ellentmondás. □

Lássuk be, hogy $f^{\max}$ is elérhető.

4.2.13. Állítás. *$\forall x \in S\ \exists x' \in S : x' \geq h'$ és $f(x) \leq f(x')$.*

Bizonyítás. Legyen $x \in S$. $x' = x \oplus h$ jó választás lesz, hiszen ha $x_j < h'_j$ valamely j koordinátára, akkor a maximum a rendszer egyik oldalán sem x_j-n vétetik fel, így x_j-t h'_j-re cserélve nem sértünk meg egy egyenletet sem. $f(x')$ pedig legalább akkora, mint $f(x)$, hiszen $x \leq x'$ és a maxlineáris függvények izotónok. □

4.2.14. Következmény. *Legyen*

$$\bar{S}' = S \cap \{x \in \mathbb{R}^n : h'_j \leq x_j \leq f_j^{-1} \otimes U, j \in [n]\}.$$

Ha $f^{\max}$ véges, akkor

$$f^{\max} = \max_{x\in\bar{S}'} f(x).$$

Bizonyítás. Tudjuk, hogy $f^{\max}$ véges, ezért a 4.2.9 Tétel szerint az $A \otimes x = B \otimes x$ egyenletnek nem létezik nemtriviális megoldása. Ekkor a 4.2.8 Lemma (ii) pontja miatt $f(x) \leq U$ minden $x \in S$ esetén. Az előző állításból pedig következik, hogy létezik $x' \in S : x' \geq h'$ és $f(x) \leq f(x')$. Ezen x'-re pedig teljesül, hogy $x' \in \bar{S}'$. □

Összességében megfogalmazhatjuk a következő állítást:

4.2.15. Tétel. *Ha $S \neq \emptyset$ és $f^{\min} > -\infty$ $[f^{\max} < +\infty]$, akkor $S^{\min} \neq \emptyset$ $[S^{\max} \neq \emptyset]$.*

Láttuk, hogy ha $c \geq d$, akkor $f(x) \geq L$ minden $x \in S$-re. Ez az állítás azonban nem túl hasznos, ha $c = d$, hiszen ekkor $L = -\infty$. Mivel szükségünk lesz egy alsó korlátra $f^{\max}$ értékére nézve akkor is, ha $c = d$, legyen $L' = f(h')$. Ekkor a következőt vehetjük észre:

4.2.16. Következmény. *Ha $x \in S$, akkor $x' = x \oplus h'$ teljesíti az $f(x') \geq L'$ egyenlőtlenséget, így $f^{\max} \geq L'$.*

Készen állunk az algoritmusok felírására. Láttuk, hogy pszeudopolinomiális időben le tudjuk ellenőrizni, hogy van-e megoldása a kétoldali rendszernek, illetve azt is, hogy a célfüggvény korlátos-e. Így feltehetjük, hogy létezik megoldása a feltételrendszernek és a célfüggvény korlátos, azaz van optimális megoldása a feladatnak. Az algoritmusok ötlete: keresünk egy x^0 megoldást, amit átskálázunk (ha szükséges), majd pedig intervallumfelezést alkalmazva optimális megoldást találunk. Kulcsfontosságú, hogy olyan $x \in S$ létezése, melyre $f(x) = \alpha$ teljesül, ekvivalens egy egyenletrendszer megoldhatóságával. Az algoritmusok akkor állnak meg, ha adott δ-nál közelebb vagyunk az optimális értékhez. Először a célfüggvény minimalizálására szolgáló algoritmust mutatjuk be.
Input: $A, B \in \mathbb{R}^{m \times n}$, $f \in \mathbb{R}^n$, $c, d \in \mathbb{R}^m, c \geq d, c \neq d$, $\delta > 0$.
Output: $x \in S$, melyre $f(x) - f^{\min} \leq \delta$.

Algoritmus 6 MAXLINMIN [5, 10.2.20]

1: **if** $\exists x \in S : L = f(x)$ **then**
2: STOP $(f^{\min} = L)$
3: **end if**
4: Keressünk egy $x^0 \in S$ megoldást.
5: **if** $\forall i \in [m] : (A \otimes x^0)_i > c_i$ **then**
6: $x^0 := \alpha \otimes x^0$, ahol $\alpha = \max\limits_{i \in [m]} (c_i \otimes (A \otimes x^0)_i^{-1})$
7: **end if**
8: $L(0) := L, U(0) := f(x^0), r := 0$
9: $\xi := \frac{1}{2}(L(r) + U(r))$
10: **if** $\exists x \in S : f(x) = \xi$ **then**
11: $U(r+1) := \xi, L(r+1) := L(r)$
12: **else**
13: $U(r+1) := U(r), L(r+1) := \xi$
14: **end if**
15: $r := r + 1$
16: **if** $U(r) - L(r) \leq \delta$ **then**
17: STOP
18: **else**
19: Goto 9
20: **end if**

4.2.17. Tétel. *A MAXLINMIN algoritmus helyes és az iterációk száma*

$$\mathcal{O}(\log \tfrac{U-L}{\delta}),$$

ahol log *kettes alapú logaritmust jelöl.*

Bizonyítás. Az x^0 megoldás esetleges skálázása után a 4.2.7 Lemma miatt x^0 továbbra is megoldása a feladatnak. Ekkor a 4.2.8 Lemma (i) pontjából következik, hogy $f(x^0) \leq U$. Feltehető, hogy $c \geq d$, ezért $L \leq f(x^0) \leq U$ (4.2.5 Lemma). Ezek után pedig a 4.2.4 Állítás értelmében szabadon alkalmazhatunk intervallumfelezéseket az optimális megoldás keresésére. A futási idő szintén abból következik, hogy az $(L(r), U(r))$ intervallum hossza minden iteráció során feleződik, egészen addig, míg legfeljebb δ nem lesz az $L(r)$ és $U(r)$ közti távolság. $\square$

Hasonló algoritmust írhatunk fel a célfüggvény maximalizálására is.
Input: $A, B \in \mathbb{R}^{m\times n}$, $f \in \mathbb{R}^n$, $c, d \in \mathbb{R}^m, c \geq d, c \neq d$, $\delta > 0$.
Output: $x \in S$, melyre $f^{\max} - f(x) \leq \delta$ vagy bizonyíték, hogy $f^{\max} = +\infty$.

Algoritmus 7 MAXLINMAX [5, 10.2.22]

1: **if** $\exists x \in S : U = f(x)$ **then**
2: STOP ($f^{\max} = U$)
3: **end if**
4: **if** $\exists x : A \otimes x = B \otimes x$ **then**
5: STOP ($f^{\max} = +\infty$)
6: **end if**
7: Keressünk egy $x^0 \in S$ megoldást.
8: $x^0 := x^0 \oplus h'$, ahol $h'_j = \min(\min\limits_{r\in[m]} a_{rj}^{-1} \otimes c_j, \min\limits_{r\in[m]} b_{rj}^{-1} \otimes d_j)$
9: $L(0) := f(x^0), U(0) := U, r := 0$
10: $\xi := \frac{1}{2}(L(r) + U(r))$
11: **if** $\exists x \in S : f(x) = \xi$ **then**
12: $U(r+1) := U(r), L(r+1) := \xi$
13: **else**
14: $U(r+1) := \xi, L(r+1) := L(r)$
15: **end if**
16: $r := r + 1$
17: **if** $U(r) - L(r) \leq \delta$ **then**
18: STOP
19: **else**
20: Goto 10
21: **end if**

4.2.18. Tétel. *A MAXLINMAX algoritmus helyes és az iterációk száma*

$$\mathcal{O}(\log \tfrac{U-L'}{\delta}).$$

Bizonyítás. Az x^0 megoldás skálázása után a 4.2.8 Lemma (i) pontja és a 4.2.16 Következmény miatt x^0 továbbra is megoldása a feladatnak és $L' \leq f(x^0) \leq U$. Ezek után pedig a 4.2.4 Állítás értelmében szabadon alkalmazhatunk intervallumfelezéseket az optimális megoldás keresésére. A futási idő ismét abból következik, hogy az $(L(r), U(r))$ intervallum hossza minden iteráció során feleződik, egészen addig, míg legfeljebb δ nem lesz az $L(r)$ és $U(r)$ közti távolság. $\square$

Ha a 4.2.1 problémában minden elem egész, akkor megmutatható, hogy $f^{\max}$ és $f^{\min}$ értéke is egész. A fenti algoritmusok enyhén módosított változatai ebben az esetben megtalálják a pontos optimális egész megoldást véges sok lépésben és pszeudopolinomiális időben (Butkovič [5, 10.2.4. fejezet]).

5. Kitekintés

Befejezésképp bemutatunk két további alkalmazást. Elsőként a mean payoff játékot és hozzá kapcsolódóan egy tételt ismertetünk. Ehhez Benchimol tézisét [2, 1.1.4. fejezet] használjuk fel. Ezután a dolgozatot filogenetikai fák matematikai vizsgálatával zárjuk [18] alapján, ahol a lineáris programozás helyett a trópusi geometrián lesz hangsúly.

5.1. Mean payoff

Tekintsük a következő kétszemélyes játékot. Legyenek a játékosok Max és Min, valamint tartozzon mindkettőjükhöz egy-egy mátrix: $A, B \in \mathbb{T}^{m \times n}$. Legyen j_0 egy rögzített kezdeti állapot. A játék menete: Min választ egy $i_1 \in [m]$ állapotot, melyre $B_{i_1 j_0} \neq \varepsilon$ és Max fizet Min-nek $B_{i_1 j_0}$ összeget. ($B_{i_1 j_0} < 0$ úgy értendő, hogy Min fizet Max-nak $-B_{i_1 j_0}$ összeget.) Ezután Max választ egy $j_1 \in [n]$ állapotot, melyre $A_{i_1 j_1} \neq \varepsilon$ és Min fizet Max-nak $A_{i_1 j_1}$ összeget. Így tovább a végtelenségig.

Feltesszük, hogy A minden sora és B minden oszlopa kvázi-valós, így mindkét játékos mindig tud olyan állapotot választani, melyre az adott mátrix megfelelő eleme nem ε. Ha a játékosok által generált végtelen állapotsorozat $j_0, i_1, j_1, i_2, j_2, \ldots$, akkor a Max játékoshoz társított mean payoff értéke

$$P_{\max} = \liminf_{p \to \infty} p^{-1}(-B_{i_1 j_0} + A_{i_1 j_1} - B_{i_2 j_1} + A_{i_2 j_2} + \cdots - B_{i_p j_{p-1}} + A_{i_p j_p}),$$

hasonlóan a Min játékos esetén

$$P_{\min} = \limsup_{p \to \infty} p^{-1}(-B_{i_1 j_0} + A_{i_1 j_1} - B_{i_2 j_1} + A_{i_2 j_2} + \cdots - B_{i_p j_{p-1}} + A_{i_p j_p}).$$

Egy stratégiát nevezzünk *pozicionális*nak, ha a következő állapot a jelenlegi egy előre meghatározott függvénye. Fontos eredmény, hogy a játéknak van egy *érték*e és optimális pozicionális stratégiái [10, 16]: létezik $\chi_0 = \chi \in \mathbb{R}$ és pozicionális stratégiája Min-nek és Max-nak, hogy

- $P_{\min} \leq \chi$, ha Min a pozicionális stratégiája szerint játszik. Ez független Max döntéseitől.
- $P_{\max} \geq \chi$, ha Max a pozicionális stratégiája szerint játszik. Ez független Min döntéseitől.

Így, ha mindkét játékos optimálisan játszik, akkor $P_{\min} = P_{\max} = \chi$. Azt mondjuk, hogy a kezdeti j_0 állapot *nyerő* a Max játékosnak, ha $\chi_0 \geq 0$. Természetes kérdés lehet, hogy egy adott kezdeti állapot nyerő-e. Ez a MEAN-PAYOFF döntési probléma. Zwick és Paterson megmutatták, hogy MEAN-PAYOFF $\in$ NP $\cap$ co-NP [20]. Nyitott kérdés, hogy létezik-e polinomiális algoritmus a MEAN-PAYOFF feladatra. Az alábbi tétel kapcsolatot teremt a mean payoff játékok és a trópusi lineáris programozás között.

5.1.1. Tétel. *([1, 3.2. Tétel]) A $j_0 \in [n]$ kezdeti állapot nyerő a Max játékosnak az A és B mátrixokkal játszott mean payoff játékban pontosan akkor, ha létezik $x \in \mathbb{T}^n$, melyre $x_{j_0} = 0$ és*

$$A \otimes x \geq B \otimes x.$$

5.2. Filogenetikai fák

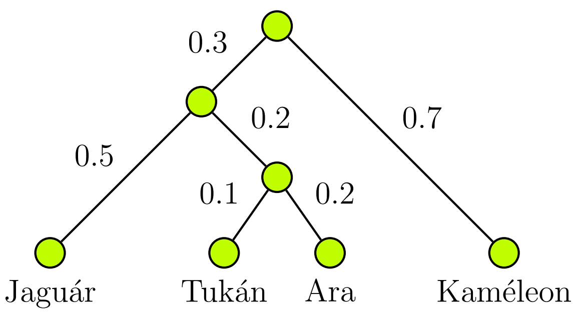
Filogenetikai fák készítése fontos bioinformatikai feladat: ha adott n darab taxon és a köztük levő távolságok, akkor készíthetünk hozzájuk egy n levelű, élsúlyokkal ellátott filogenetikai fát. A fa célja, hogy különböző fajok evolúciós kialakulását, rokonságát szemléltesse. Legyen adott négy trópusi organizmus DNS-sorozata:

$$\begin{aligned} \text{Jaguár} &: AGACGGTCAG\ldots \\ \text{Tukán} &: CGAATCTAAG\ldots \\ \text{Ara} &: CGTATGTACG\ldots \\ \text{Kaméleon} &: GATGCATTAC\ldots \end{aligned}$$

Ilyen módon adott adatokból kikövetkeztethető bármely két taxon közti távolság különböző algoritmusok segítségével. Tegyük fel, hogy meghatároztuk a távolságokat. Ekkor készíthetünk egy D mátrixot, melynek minden d_{ij} eleme éppen az i és a j taxon távolsága. Legyen most a mátrix például

$$D = \begin{pmatrix} 0 & 0,8 & 0,9 & 1,5 \\ 0,8 & 0 & 0,3 & 1,3 \\ 0,9 & 0,3 & 0 & 1,4 \\ 1,5 & 1,3 & 1,4 & 0 \end{pmatrix}.$$

A célunk olyan élsúlyozott fát találni (ha létezik), melynek minden levele egy taxont jelöl és a két levél közti egyértelmű út súlya megegyezik a két taxon távolságával. A példában szereplő D mátrixhoz tudunk ilyen fát készíteni:



5.2.1. Definíció (Metrika)**.** Egy D mátrixot metrikának nevezünk, ha teljesül rá a háromszög-egyenlőtlenség, azaz $d_{ik} \leq d_{ij} + d_{jk}$ minden i, j, k esetén.

A háromszög-egyenlőtlenség mint feltétel felírható trópusi egyenletként is: a D mátrix pontosan akkor metrika, ha $D \otimes' D = D$. Egy D mátrixot hívjunk *fa-metrikának* az $\{1, 2, \dots, n\}$ halmazon, ha létezik olyan n levelű T fa, melyre bármely két i, j levél távolsága éppen d_{ij}. A filogenetikában használatos *négypont-feltétel* a következőt állítja [11, 179. oldal]: D akkor és csak akkor fa-metrika, ha

$$d_{ij} + d_{kl} \leq \max\{d_{ik} + d_{jl}, d_{il} + d_{jk}\}$$

minden $\{i, j, k, l\} \subset \{1, 2, \dots, n\}$ négyesre. Vegyük észre, hogy a feltétel pontosan akkor teljesül, ha a $d_{ij} + d_{kl}, d_{ik} + d_{jl}$ és $d_{il} + d_{jk}$ értékek közül legalább kettő megegyezik. Célunk matematikailag megfogalmazni ezen feltételt. Ehhez azonban szükségünk lesz további trópusi algebrai és geometriai fogalmakra.

*Trópusi polinom*oknak nevezzük a

$$p(x_1, x_2, \dots, x_n) = \bigoplus\nolimits_{i=1}^{\prime k} c_i \otimes' x_1^{\alpha_i^{(1)}} x_2^{\alpha_i^{(2)}} \dots x_n^{\alpha_i^{(n)}}$$

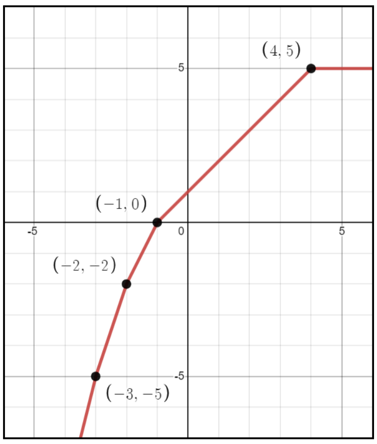
kifejezéseket, ahol a c_i-k valósak, az $\alpha_i^{(l)}$-k egészek bármely i, l esetén. Minden trópusi polinom felfogható egy $\mathbb{R}^n \to \mathbb{R}$ függvényként is. Áttérve klasszikus műveletekre ez a következőt jelenti:

$$p : (x_1, x_2, \dots, x_n) \mapsto \min\{c_1 + \alpha_1^{(1)} x_1 + \alpha_1^{(2)} x_2 + \dots + \alpha_1^{(n)} x_n, \dots\}.$$

Azaz, ha ábrázolni szeretnénk egy ilyen függvényt, akkor véges sok lineáris függvény minimumát kell vennünk, így egy szakaszonként lineáris, konkáv függvényt kapunk. Rajzoljuk le például a

$$\hat{p}(x) = 7x^4 \oplus' 4x^3 \oplus' 2x^2 \oplus' x \oplus' 5$$

függvényt. Egy polinom *gyökei* azon x helyek, ahol a különböző meredekségű lineáris részek váltják egymást.



3. ábra. A $\hat{p}(x)$ trópusi polinom és gyökei

5.2.2. Megjegyzés. $\hat{p}(x)$ átírható a

$$7 \otimes' (x \oplus' -3) \otimes' (x \oplus' -2) \otimes' (x \oplus' -1) \otimes' (x \oplus' 4)$$

gyöktényezős alakra, ahol a gyököket az együtthatókból számolhatjuk ki: $-3 = c_2 - c_1, -2 = c_3 - c_2, -1 = c_4 - c_3, 4 = c_5 - c_4$.

Adott p polinomhoz definiáljuk a $\mathcal{H}(p)$ *hiperfelület*et azon $x \in \mathbb{R}^n$ pontok halmazának, melyekben p legalább kétszer veszi fel a minimumát. Így $x \in \mathcal{H}(p) \iff p$ nem lineáris az x helyen. Azaz egy p polinom által definiált $\mathcal{H}(p)$ hiperfelület pontjai egybeesnek p gyökhelyeivel. A példánkban szereplő polinom a következő hiperfelületet határozza meg:

$$\mathcal{H}(\hat{p}) = \{-3, -2, -1, 4\}.$$

Visszatérve a fa-metrikák problémájára, legyen $X = (x_{ij}) \in \mathbb{R}^{n \times n}$ szimmetrikus mátrix, melynek főátlójában csupa 0 elemek állnak. X maradék $\frac{n(n-1)}{2}$ eleme ismeretlen. Definiáljunk minden $\{i, j, k, l\} \subset \{1, 2, \ldots, n\}$ négyeshez egy másodfokú polinomot:

$$p_{ijkl} \coloneqq x_{ij} \otimes' x_{kl} \oplus' x_{ik} \otimes' x_{jl} \oplus' x_{il} \otimes' x_{jk}.$$

Ezen polinomot nevezzük a *trópusi Grassmann-Plücker-viszony*nak. A polinom definiál egy $\mathcal{H}(p_{ijkl})$ hiperfelületet az $\mathbb{R}^{\binom{n}{2}}$ téren. A *trópusi Grassmann-tér* az összes lehetséges $\{i, j, k, l\}$ négyes választás által meghatározott hiperfelületek metszete:

$$Gr_{2,n} = \bigcap_{1 \leq i < j < k < l \leq n} \mathcal{H}(p_{ijkl}).$$

A $Gr_{2,n}$ teret szokás a filogenetikai fák terének is hívni, hiszen a négypont-feltétel valójában az alábbit jelenti.

5.2.3. Tétel (Négypont-feltétel)**.** *Legyen D az $\{1, 2, \ldots, n\}$ halmazon értelmezett metrika, $X \coloneqq -D$. Ekkor*

$$D \text{ fa-metrika} \iff X \in Gr_{2,n}.$$

Hivatkozások

[1] Marianne Akian, Stéphane Gaubert, and Alexander Guterman. Tropical polyhedra are equivalent to mean payoff games. *Internat. J. Algebra Comput.*, 22(1):1250001, 43, 2012.

[2] Pascal Benchimol. Tropical aspects of linear programming. *Optimization and Control [math.OC]. Ecole Polytechnique*, 2014.

[3] P. Butkovic and A. Aminu. Introduction to max-linear programming. *IMA J. Manag. Math.*, 20(3):233–249, 2009.

[4] Peter Butkovič. Max-algebra: the linear algebra of combinatorics? *Linear Algebra Appl.*, 367:313–335, 2003.

[5] Peter Butkovič. *Max-linear systems: theory and algorithms*. Springer Monographs in Mathematics. Springer-Verlag London, Ltd., London, 2010.

[6] Peter Butkovič. A note on tropical linear and integer programs. *J. Optim. Theory Appl.*, 180(3):1011–1026, 2019.

[7] Peter Butkovič and Marie MacCaig. On the integer max-linear programming problem. *Discrete Appl. Math.*, 162:128–141, 2014.

[8] R. A. Cuninghame-Green and P. Butkovic. The equation $A \otimes x = B \otimes y$ over $(\max, +)$. volume 293, pages 3–12. 2003. Max-plus algebras.

[9] Raymond Cuninghame-Green. *Minimax algebra*, volume 166 of *Lecture Notes in Economics and Mathematical Systems*. Springer-Verlag, Berlin-New York, 1979.

[10] A. Ehrenfeucht and J. Mycielski. Positional strategies for mean payoff games. *Internat. J. Game Theory*, 8(2):109–113, 1979.

[11] Joseph Felsenstein. *Inferring Phylogenies*. Sinauer Associates, Inc., 2004.

[12] Richard M. Karp. A characterization of the minimum cycle mean in a digraph. *Discrete Math.*, 23(3):309–311, 1978.

[13] Zoltán Király. *Algoritmuselmélet*. Typotex, 2022.

[14] N. K. Krivulin. *Methods of Idempotent Algebra for Problems in Modeling and Analysis of Complex Systems*. Saint Petersburg Univ. Press, St. Petersburg, 2009.

[15] Nikolai Krivulin. Tropical optimization problems with application to project scheduling with minimum makespan. *Ann. Oper. Res.*, 256(1):75–92, 2017.

[16] Thomas M. Liggett and Steven A. Lippman. Stochastic games with perfect information and time average payoff. *SIAM Rev.*, 11:604–607, 1969.

[17] M. MacCaig P. Butkovič. The alternating method for finding integer solutions to two-sided systems. *University of Birmingham, School of Mathematics*, 2012/08.

[18] David Speyer and Bernd Sturmfels. Tropical mathematics. *Math. Mag.*, 82(3):163–173, 2009.

[19] K. Zimmermann. Extremální algebra. *Výzkumná publikace Ekonomicko–matematické laboratore pri Ekonomickém ústave CSAV, 46, Praha*, 1976.

[20] Uri Zwick and Michael S. Paterson. The complexity of mean payoff games. In *Computing and combinatorics (Xi'an, 1995)*, volume 959 of *Lecture Notes in Comput. Sci.*, pages 1–10. Springer, Berlin, 1995.

NYILATKOZAT

Név: Szabados Balázs

ELTE Természettudományi Kar, szak: Matematika BSc

NEPTUN azonosító: XM4YBP

Szakdolgozat címe:
Trópusi lineáris programozás és alkalmazásai

A **szakdolgozat** szerzőjeként fegyelmi felelősségem tudatában kijelentem, hogy a dolgozatom önálló szellemi alkotásom, abban a hivatkozások és idézések standard szabályait következetesen alkalmaztam, mások által írt részeket a megfelelő idézés nélkül nem használtam fel.

Budapest, 2023. június 5.

Szabados Balázs

a hallgató aláírása