

EÖTVÖS LORÁND TUDOMÁNYEGYETEM
TERMÉSZETTUDOMÁNYI KAR

Antal Ádám

OPTIMALIZÁLÁSI FELADATOK
PÁROS ÖSSZEHASONLÍTÁSI MÁTRIXOKRA

MSc alkalmazott matematikus szakdolgozat

Belső témavezető:

Bérczi-Kovács Erika

ELTE Operációkutatási Tanszék

Külső témavezető:

Bozóki Sándor

MTA SZTAKI



Budapest, 2018

Tartalomjegyzék

1. Bevezetés	5
2. Definíciók	9
2.1. Páros összehasonlítás mátrixok és súlyvektorok	9
2.1.1. Súlyvektorok normálása	12
2.1.2. Egy 3 dimenziós példa	13
2.2. Súlyvektor-számítási módszerek	14
2.2.1. Sajátvektor módszer	15
2.2.2. Legkisebb négyzetek módszere	15
2.2.3. Logaritmikus legkisebb négyzetek módszere	16
2.2.4. Feszítőfás módszer	16
2.2.5. Koszinuszos módszer	17
2.3. Indexek az inkonzisztencia mérésére	17
2.3.1. CR	17
2.3.2. CM	18
2.3.3. GCI	19
2.3.4. Távolságminimalizáló indexek	19
2.4. A CR korlátjai	19
3. Hatékonyság ellenőrzése	23
3.1. Lineáris programozási feladat	23
3.2. Egy algoritmus a domináló hatékony súlyvektorok keresésére	24
3.2.1. Gráfrepresentáció	24
3.2.2. Az algoritmus	25
3.2.3. Az algoritmus elemzése	27
3.2.4. Példák	29
3.3. Lineáris programozási feladat és a gráf alapú algoritmus összehasonlítása	29
4. Numerikus eredmények	30
4.1. Általános megjegyzések	30
4.2. Súlyvektorok hatékonyságának eloszlása különböző módszerek esetén	31
4.2.1. 4 dimenziós eset	31
4.2.2. Magasabb dimenzió	31

4.3. Főátló feletti egynél nagyobb elemek	35
5. Programkönyvtár a mátrixok vizsgálatára	36
5.1. Mátrixok generálása, kiírása és olvasása	36
5.2. Mátrixok szűrése	37
5.2.1. Permutációsűrés	37
5.2.2. <i>CR</i> szerinti szűrés	37
5.3. Módszerek futtatása	37
5.4. Statisztikák, vizualizáció, egyebek	38
5.4.1. Statisztikák készítése	38
5.4.2. Vizualizáció	38
5.4.3. Egyéb hasznos függvények	38
6. Melléklet	43
6.1. Hatékonyság	43
6.2. Mátrixtesztelés	44
6.3. Erősen összefüggővé tevő algoritmus	45

Köszönetnyilvánítás

Szeretnék kiemelten köszönetet mondani témavezetőmnek, Bozóki Sándornak (MTA SZTA-KI), aki mióta csak felkeltette érdeklődésemet a téma iránt, folyamatos együttműködés mellett számos módon segítette a kutatásomat, valamint rengeteg hasznos megjegyzésével emelte a dolgozat minőségét.

Külön köszönet illeti belső konzulensemét, Bérczi-Kovács Erikát (ELTE, Operációkutatási Tanszék), aki elvállalta a konzulensi teendőket.

Hálával tartozom Alexának a mérhetetlen türelemért, amit felém tanúsított, és a sok bátorításért, továbbá családomnak és barátaimnak, akik érdeklődve figyelték a dolgozat fejlődését.

1. fejezet

Bevezetés

Előszó

A döntési szituáció önmagában véve absztrakt fogalom, viszont tudományosan közelíthető és modellezhető. Ezeket a való életben is gyakori jelenségeket a legegyszerűbb, heurisztikára alapuló gyors emberi választól a döntéselmélet komplex modellrendszeréig sok szinten lehet tárgyalni. Feltehetjük azt az általános kérdést, hogy adott több alternatíva esetén milyen megoldást tudunk adni a helyzetre. A következő kérdésekkel konkretizáljuk a problémát:

- Hogyan tudjuk az alternatívák jellemzőit megadni?

Amennyiben kvantitatívan meg tudjuk adni ezeket, nagyobb mozgástérhez juthatunk, különben preferencialista esetében csak a választáselméletben található megoldást.

- Tudjuk-e abszolút módon jellemezni az alternatívákat, vagy csak relatívan?

- Mi a cél: adott alternatívák közül az adott szempontok szerint összességében legjobb(ak) kiválasztása vagy az alternatívák rangsorolása?

- Az alternatívák jellemzői átválthatók-e valamilyen összefüggés alapján?

Amennyiben a költség minimalizálása az egyetlen célunk, akkor mindent arra a skálára átváltva redukálhatjuk a problémát egy egykritériumos döntéshozatalra.

- Tudunk-e explicit célfüggvényt társítani a döntésünkhöz?

A többkritériumos döntéshozatal (Multi-Criteria Decision Making - MCDM) a normatív döntéselmélet egyik fontos területe. A téma erős szubjektív vonulata, hogy hogyan tudja a döntést meghozó fél megmondani, mennyit ér kvantitíve egy-egy tulajdonság a számára – feltesszük azonban, hogy impliciten ismeri a saját preferenciáit.

A modellezés egy további fontos aspektusa az *emberi következetlenség* és *szubjektivitás*. Ez az elméletben tökéletes, egymással összhangban lévő válaszokhoz „zajt”, inkonzisztenciát generál. A valós problémáknál ilyen mindig létezik, viszont a nagysága és a hatása változó lehet.

A különböző modellek szükségessége abban is megmutatkozik, hogy a sokszor rengeteg szempontot a döntéshozó nem tudja *egyszerre figyelembe venni*. A döntéshozóra továbbá a *korlátozott racionalitás* a jellemző: egy bizonyos szint felett a döntéshozó nem érdekelt további optimalizálásban. Ezzel kapcsolatban az olvasót Herbert Simon [45] és [46] munkáihoz irányítjuk.

Alkalmazások

A többkritériumos döntések alkalmazásai igen szerteágazóak: tenderek, mint például vilamosok beszerzése, hatásvizsgálatok, erőforráselosztási problémák, valamint az országok, városok, intézmények, weboldalak, sportolók rangsorolása egyaránt többszempontú döntési feladat. Ezekből a példákból látszik, hogy szükség van matematikailag megalapozott döntéshozatali módszerekre, amelyeknek komoly politikai és társadalmi felhasználásai és ezáltal hatásai vannak.

Rövid áttekintés

A következő összefoglalás [1] és [50] munkáiból származik – ismertebb módszereket példának hozva.

Bonyolultabb módszerek és heurisztikák előtt végezhetünk *előszűréseket*. Ilyen például a dominanciaszűrés, vagy a konjunktív szabály. Az ún. *elemi döntések* egyszerű, egzakt módszerek. Ilyen például a maximin elv vagy a lexikografikus rendezés – utóbbit akkor alkalmazhatjuk, ha a szempontok fontossági sorrendje ismert.

Az összetettebb módszereket osztályozhatjuk aszerint, hogy az alternatívák *átválthatóak* egymásba. Ha igen, akkor mérhetjük azokat *abszolút* vagy *relatív* skálán. A relatív skála lehet *különbség* (pl. MACBETH – Measuring Attractiveness by a Categorical Based Evaluation Technique) vagy *arányskála* (pl. AHP – Analytic Hierarchy Process). Amennyiben nem átválthatóak, használhatunk ún. „*outranking*” módszereket, ilyenek például az ELECTRE vagy a PROMETHEE.

Egy másik népszerű hozzáállás a cél vagy referenciaszint alapján történő döntéshozatal. Ilyen például a TOPSIS (Technique of Order Preference Similarity to the Ideal Solution).

Analytic Hierarchy Process

Amennyiben az alternatívák/szempontok között létezik valamiféle átváltás, akkor egy gyakorlatban hasznos alapötlet, ha a problémát kisebb részekre bontjuk, és a döntéshozónak csak egyszerű, világos kérdéseket kell megválaszolnia, melyekből előállítható az egész feladat szempontsúly-rendszere.

Így kapjuk a többszempontú döntési feladatokra alkalmazott népszerű megoldási technikát, a páros összehasonlítások módszerét, amellyel a döntéshozónak egyidejűleg csak két elemet kell összehasonlítani. Például: „Hányszor fontosabbnak ítéli meg az egyik szempontot a másik szemponttal összevetve?”. Ez a módszer Thomas L. Saaty munkája 1977-ből.

A döntéshozó preferenciáira vonatkozó becslésként kezeljük a számszerű válaszokat (arány-skálán összehasonlított szempontokat vagy alternatívákat), majd súlyokat vagy rangsort jelölünk meg a megfelelő optimalizálási feladatok felírásával és megoldásával. Erre, az ún. *páros összehasonlítások ötletére* épül a vállalati szférában is gyakran használt döntési módszer, az Analytic Hierarchy Process [44].

Az AHP az összehasonlítási értékekből képzett mátrix legnagyobb sajátértékhez tartozó jobboldali sajátvektorából határozza meg a súlyokat, de léteznek más távolságminimalizáló módszerek, azaz valamilyen célfüggvény szerinti legjobb közelítés (legkisebb négyzetek módszere, szinguláris felbontás, célprogramozás). Ha a válaszadó nem konzisztens feleletet ad, eltérő eredményeket kaphatunk ezen módszerekből. Mivel nem létezik kitüntetett legjobb megoldás, ezért ezen módszerek elemzése mind elméleti, mind gyakorlati szempontból fontos és indokolt [16].

Motiváció

A szakdolgozat megírását a döntésanalízis és azon belül is a páros összehasonlításra alapuló módszerek magyar és külföldi irodalmi áttekintése előzte meg. Megjegyezzük, hogy az irodalom egy szűk, releváns halmazát vizsgáltuk csak meg. A területet behatárolja Saaty 1977-ben megírt cikke, valamint, hogy a súlyvektorok Pareto-optimalitásának (hatékony-ságának) vizsgálatára először 2006-ban került sor.

A páros összehasonlítás mátrixokkal kapcsolatos kérdések aktív kutatás alatt állnak hazai tudományos körökben is. Ezen eredmények közül Ábele-Nagy Kristóf [1], Csató László [25] és Poesz Attila [41] doktori értekezésére, valamint Kéri Gerzson gráfelméleti megközelítésére [35] hívjuk fel az Olvasó figyelmét.

Mindazonáltal a hatékony súlyvektorok eloszlásának vizsgálatára, valamint azzal a kérdéssel, hogy aszimptotikusan mennyire fordulhatnak ilyenek elő, nem lelhető fel eredmény. Ez a munka ennek megalapozására irányul. Mind elméleti, mind gyakorlati szempontból támadható ez a kérdés, a szakdolgozat pedig mindkét területre egyaránt fókuszál.

Önálló eredmények

A dolgozat önálló eredményei a következők:

1. A 2.1.5. állításként való formalizálása (ez ismert eredmény)
2. A 2.4.4. következmény
3. A 2.4.5. állítás
4. A 3.2.2. algoritmus formális, precíz leírása
5. A 3.2.2. további elemzése a 3.2.3. alfejezetben a futásidő pontos aszimptotikájának megadásával
6. A 3.2.4. állítás az ellenpéldával
7. Az LP és a gráf algoritmus analitikus összehasonlítása 3.3. fejezetben
8. Az összes numerikus eredmény a 4. fejezetben
9. A teljes programkönyvtár önálló implementálása az 5. fejezetben

10. A példák a mellékletben a 6. fejezetben

Ezek közül az eredmények közül a következők kiemelkedőek (csoportosítva):

1. A 3.2.2. algoritmussal való rengeteg munka és eredmény
2. A programkönyvtár létrehozása és az általa kihozott rengeteg numerikus eredmény a 4. fejezetben
3. Több, sok részletet megvilágító példa leírása
4. Az elméleti vizsgálódások alatt felfedezett új állítások, mint pl. a 2.4.4. és a 2.4.5.

A szakdolgozat felépítése

A 2. fejezetben olyan alapvető definíciókba nyerhet betekintést az Olvasó, mint például a páros összehasonlítás mátrix, a súlyvektorok és ezek hatékonysága. A fejezetben több ezzel kapcsolatos tétel és állítás is szerepel, behatárolva ezzel az ismert és még felfedezetlen részeit ennek a tudománynak. A téma gazdagsága ellenére leginkább azon definíciókra szorítkoztunk melyeket később használunk a dolgozatban, vagy a fontosságára való tekintettel nem hagyhattunk ki.

A 3. fejezetben két módszer szerepel, amelyek a témakör alapvető kérdéseire adnak választ. A második módszer, ami egy gráfokon alapuló algoritmus, egy új eredmény, és ezért kiemelt fontosságú a dolgozatban: számos állítást, példát, és részletes elemezést tartalmaz a fejezet, utóbbit a futásidőre való különös tekintettel végeztük.

A 4. fejezetben foglaltuk össze tömören a numerikus eredményeket, amelyek új információt szolgáltatnak az ismert eredményekhez képest – kapcsolódva a bevezetőben írt vállaláshoz, hogy a hatékony súlyvektorok eloszlásába jobb betekintést nyerhessünk.

Az 5. rész tartalmazza röviden a kutatással párhuzamosan haladt programkönyvtár funkcióit. A könyvtárat eszközként használtuk többek között a numerikus eredmények eléréséhez, nyílt forráskodú, és bárki használhatja.

A 6. fejezet a szakdolgozat hosszabb példáit tartalmazza, amelyeket leginkább helytakarékossági okból raktuk a munka végére.

2. fejezet

Definíciók

2.1. Páros összehasonlítás mátrixok és súlyvektorok

Mint azt a bevezetőben is olvastuk, az egyik központi fogalmunk a páros összehasonlítás, amiből létrejön a páros összehasonlítás mátrix, melyet most formálisan is definiálunk.

2.1.1. Definíció. Páros összehasonlítás mátrix (Pairwise Comparison Matrix):

Az $n \geq 3$ elem (szempont, alternatíva, vagy akár szavazóerő) összehasonlításából az \mathbf{A} $n \times n$ -es páros összehasonlítás mátrixot kapjuk úgy, hogy a mátrix adott a_{ij} eleme az i -edik és a j -edik elem összehasonlításából nyert érték, amely megmutatja, hogy az i -edik szempont, alternatíva vagy szavazóerő hányszor fontosabb, jobb vagy nagyobb a j -ediknél.

Informálisan tehát a mátrix a_{ij} eleme a döntéshozó válaszát tartalmazza a „Hányszor fontosabbnak ítéli az i -t a j szempontnál?” típusú kérdésre. Az AHP módszertanában $a_{ij} \in \{1, 2, 3, \dots, 8, 9, \frac{1}{2}, \frac{1}{3}, \dots, \frac{1}{9}\}$ elemeket engedjük meg, illetve döntési problémától függően a köztes értékeket (pl. 1.5). Pl. az 1 jelképezi azt, hogy a két szempont egyformán fontos, míg a 9 azt, hogy rendkívüli mértékben fontosabb az egyik szempont a másikkal.

Feltesszük a következő tulajdonságokat:

1. $a_{ij} > 0 \quad \forall 1 \leq i, j \leq n$
2. $a_{ij} = 1/a_{ji} \quad \forall 1 \leq i, j \leq n$
3. $a_{ii} = 1 \quad \forall 1 \leq i \leq n$

A gyakorlatban előfordul, hogy nem akarjuk a döntéshozóval kitöltetni mind az $\binom{n}{2}$ elemét a mátrixnak. Ezért általánosításként definiáljuk a nem teljesen kitöltött esetet:

2.1.2. Definíció. \mathbf{A} egy nem teljesen kitöltött páros összehasonlítási mátrix, ha felírható

$$\mathbf{A} = \begin{pmatrix} 1 & a_{12} & - & \dots & a_{1n} \\ \frac{1}{a_{12}} & 1 & a_{23} & \dots & - \\ - & \frac{1}{a_{23}} & 1 & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{1}{a_{1n}} & - & \frac{1}{a_{3n}} & \dots & 1 \end{pmatrix}$$

alakban, ahol bizonyos (nem feltétlenül a fentiek szerinti, „-” jellel jelölt) a_{ij} -k hiányoznak.

Továbbra is feltesszük, hogy ha a_{ij} létezik, akkor $a_{ij} > 0$. A hiányzó elemek a definíció szerint bárhol előfordulhatnak, de feltesszük hogy ha a_{ij} létezik, akkor a_{ji} is és $a_{ji} = \frac{1}{a_{ij}}$.

A továbbiakban a teljesen kitöltött esetre értjük a megállapításokat. Jelezzük, amennyiben egy állítás vagy algoritmus érvényes lesz a nem teljesen kitöltött esetre is.

Az $n \times n$ -es teljesen kitöltött páros összehasonlítás mátrixok halmazát \mathcal{PCM}_n -nel jelöljük.

2.1.3. Definíció. Abban az esetben, amikor fennáll a *kardinális tranzitivitás*, azaz $a_{ij}a_{jk} = a_{ik} \forall i, j, k$ indexhármásra, a páros összehasonlítás mátrixot *konzisztensnek*, különben pedig *inkonzisztensnek* nevezzük.

A konzisztencia központi fogalom, a továbbiakban gyakran fogunk hivatkozni rá. Az inkonzisztencia, mint ahogy az előszóban is hivatkoztunk rá, nem elhanyagolható gyakorlati alkalmazások esetében, de először tekintsük a konzisztens mátrixok egy szép karakterizációját.

2.1.4. Definíció. Egy $\mathbf{A} \in \mathcal{PCM}_n$ -hez tartozó $\mathbf{w} \in \mathbb{R}_+^n$ -t *súlyvektornak* nevezzük.

A döntéshozó az a_{ij} mátrixelemmel az általa nem ismert $\mathbf{w} = (w_1, w_2, \dots, w_n)$ súlyvektorból képzett $\frac{w_i}{w_j}$ hányadost közelíti. A feladat a gyakorlatban ennek a megfordítottja: adott \mathbf{A} mátrixhoz olyan \mathbf{w} súlyvektort keresünk, hogy a $\frac{w_i}{w_j}$ hányadosok közelítsék az a_{ij} elemét.

A következő tétel alapozza meg a vizsgálódásainkat:

2.1.5. Állítás. Egy $\mathbf{A} \in \mathcal{PCM}_n$ mátrix pontosan akkor konzisztens, hogy ha konstans szorzó erejéig egyértelműen létezik olyan hozzá tartozó \mathbf{w} súlyvektor, hogy $a_{ij} = \frac{w_i}{w_j} \forall i, j$.

Bizonyítás: Ha létezik egy ilyen \mathbf{w} súlyvektor, akkor konzisztens a mátrix, ugyanis tetszőleges i, j, k indexekre:

$$a_{ij}a_{jk} = \frac{w_i}{w_j} \frac{w_j}{w_k} = \frac{w_i}{w_k} = a_{ik}$$

Megfordítva legyen $\mathbf{w} = (1, \frac{1}{a_{12}}, \frac{1}{a_{13}}, \dots, \frac{1}{a_{1n}})$. Ekkor tetszőleges i, j indexekre kiindulva \mathbf{A} konzisztenciájából:

$$a_{1i}a_{ij} = a_{1j} \implies a_{ij} = \frac{a_{1j}}{a_{1i}} = \frac{\frac{1}{a_{1j}}}{\frac{1}{a_{1i}}} = \frac{w_i}{w_j}$$

Látható, hogy valóban létezik ilyen \mathbf{w} súlyvektor. Az egyértelműséghez legyen $\mathbf{w}' \neq \mathbf{w}$ egy szintén \mathbf{A} -hoz tartozó súlyvektor, és legyen $c = \frac{w'_1}{w_1} = w'_1$. Ekkor $\mathbf{w}' = c\mathbf{w}$.

Indirekt tegyük fel, hogy ez nem igaz, tehát $\exists k : w'_k \neq cw_k$. De ekkor

$$\frac{w'_k}{w'_1} = a_{k1} = \frac{w_k}{w_1} = w_k \implies w'_k = w'_1 w_k = cw_k$$

Ez pedig ellentmondás. □

$$\begin{pmatrix} 1 & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & 1 & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & 1 & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & 1 \end{pmatrix} \quad \begin{pmatrix} 1 & \frac{w_1}{w_2} & \frac{w_1}{w_3} & \cdots & \frac{w_1}{w_n} \\ \frac{w_2}{w_1} & 1 & \frac{w_2}{w_3} & \cdots & \frac{w_2}{w_n} \\ \frac{w_3}{w_1} & \frac{w_3}{w_2} & 1 & \cdots & \frac{w_3}{w_n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{w_n}{w_1} & \frac{w_n}{w_2} & \frac{w_n}{w_3} & \cdots & 1 \end{pmatrix}$$

2.1. ábra. A mátrix eredeti elemei, és a \mathbf{w} súlyvektorral való becslése

Következésképpen ha a mátrix konzisztens, akkor normálás erejéig egyértelműen tudunk ilyen súlyvektort kapni. A gyakorlatban felmerülő probléma a következő: milyen súlyvektort tudunk kinyerni inkonzisztens páros összehasonlítás mátrixokból? Erről lesz szó a továbbiakban. A 6.1.1. példa a mellékletben tartalmaz két egyszerű példát a konzisztencia és az inkonzisztencia demonstrálására.

Először Blanquero, Carrizosa és Conde fogalmazták meg a következő többcélú optimalizálási feladatot [7]. Adott $\mathbf{A} = [a_{ij}]_{i,j=1,\dots,n}$ páros összehasonlítás mátrix esetén a súlyvektor számítása egy olyan többcélú optimalizálási feladat, amelynek célfüggvényei a minimalizálandó $|x_i/x_j - a_{ij}|$, $1 \leq i \neq j \leq n$ függvények, összesen $n^2 - n$ darab.

$$\min_{\mathbf{x} \in \mathbb{R}_+^n} \left| \frac{x_i}{x_j} - a_{ij} \right|_{1 \leq i \neq j \leq n}. \quad (2.1)$$

A mátrix elemeit egyenként minél jobban akarjuk közelíteni az $\frac{x_i}{x_j}$ hányadosokkal.

A hatékonyság vagy Pareto optimalitás központi fogalma a többcélú optimalizálásnak. Definiáljuk a következő függvényeket:

$$f_{ij} : \mathbb{R}_+^n \rightarrow \mathbb{R} \quad f_{ij}(\mathbf{x}) = \left| \frac{x_i}{x_j} - a_{ij} \right|, \quad i, j = 1, \dots, n, \quad (2.2)$$

Mivel $f_{ii}(\mathbf{x}) = 0$ minden $\mathbf{x} \in \mathbb{R}_+^n$ -re és $i = 1, \dots, n$ -re ezek a konstans függvények irrelevánsak az optimalizációs probléma szempontjából, ezért elhagyhatóak.

Legyen a vektor-értékű $\mathbf{f} : \mathbb{R}_+^n \rightarrow \mathbb{R}^{n(n-1)}$ függvény definiálva az f_{ij} , $i, j = 1, \dots, n$, $i \neq j$ komponenseken keresztül. Ekkor nézhetjük az \mathbf{f} minimalizálási problémáját egy nemüres $X \subseteq \mathbb{R}^n$ felett, ami egy általános alakú vektor-optimalizálási problémaként így írható fel:

$$\min_{\mathbf{x} \in X} \mathbf{f}(\mathbf{x}). \quad (2.3)$$

A vektoroptimalizálásban használt fogalmakat (dominálás, hatékonyság) itt már csak a konkrét feladatra alkalmazva definiáljuk.

2.1.6. Definíció. Adott $\mathbf{A} \in \mathcal{PCM}_n$ és hozzá tartozó $\mathbf{w} = (w_1, w_2, \dots, w_n)^\top$, $\mathbf{w}' = (w'_1, w'_2, \dots, w'_n)^\top$ súlyvektorok esetén azt mondjuk, hogy \mathbf{w}' dominálja vagy más néven

javítja \mathbf{w} -t, ha:

$$\left| a_{ij} - \frac{w'_i}{w'_j} \right| \leq \left| a_{ij} - \frac{w_i}{w_j} \right| \quad \text{minden } 1 \leq i, j \leq n, \text{ esetén, és} \quad (2.4)$$

$$\left| a_{k\ell} - \frac{w'_k}{w'_\ell} \right| < \left| a_{k\ell} - \frac{w_k}{w_\ell} \right| \quad \text{valamely } 1 \leq k, \ell \leq n \text{ esetén.} \quad (2.5)$$

A dominancia tranzitív reláció.

2.1.7. Definíció. Rögzített $\mathbf{A} \in \mathcal{PCM}_n$ esetén a \mathbf{w} súlyvektor *hatékony* vagy más néven *Pareto-optimalis*, ha nem létezik olyan \mathbf{w}' súlyvektor, hogy \mathbf{w}' dominálja \mathbf{w} -t.

A definícióból következik, hogy egy súlyvektor hatékonysága nem függ a normálástól, ami egybevág a 2.1.5. állítással, ugyanis normálás erejéig egyértelmű a konzisztens páros összehasonlítás mátrixhoz tartozó súlyvektor.

Egy hatékony súlyvektort tehát nem lehet dominálni, azaz megjavítani úgy, hogy legalább egy pozícióban jobban közelítse a döntéshozó által megadott értéket, miközben egyetlen más pozícióban sem keletkezik rosszabb közelítés.

Többen a hatékonyság gyenge változatát is definiálták ([7, Remark 12], [22, Theorem 2.2]). Erre a megengedőbb definícióra (ami egy gyengébb garanciát is ad) könnyebben tudunk tételeket bizonyítani.

2.1.8. Definíció. Adott $\mathbf{A} \in \mathcal{PCM}_n$ és $\mathbf{w} = (w_1, w_2, \dots, w_n)^\top$, $\mathbf{w}' = (w'_1, w'_2, \dots, w'_n)^\top$ súlyvektorok esetén azt mondjuk, hogy \mathbf{w}' *szigorúan dominálja* vagy *szigorúan javítja* \mathbf{w} -t, ha:

$$\left| a_{ij} - \frac{w'_i}{w'_j} \right| < \left| a_{ij} - \frac{w_i}{w_j} \right| \quad \text{minden } 1 \leq i \neq j \leq n \text{ esetén.} \quad (2.6)$$

2.1.9. Definíció. A \mathbf{w} súlyvektor *gyengén hatékony*, ha nem létezik olyan \mathbf{w}' súlyvektor, ami őt szigorúan dominálná.

2.1.1. Súlyvektorok normálása

A 2.1.5. állítás alapján a súlyvektor konstans szorzó erejéig egyértelmű. Felmerül a kérdés, hogy hogyan adjuk meg a súlyvektorokat, milyen c szorzókat használjunk. Mi a következő két technikát alkalmaztuk:

Normálás Logikus döntés az L_1 norma szerinti megadása a súlyvektoroknak, azaz olyan \mathbf{w} felírás, melyre:

$$\sum_{i=1}^n w_i = 1$$

teljesüljön. Ez gyakorlatilag azt jelenti, hogy a súlyvektorok elemeit százalékékként is értelmezhetjük: hány százalékban tartja a döntéshozó fontosnak az adott szempontot. Hasonlóképpen értelmezhető a súlyvektor más, L_p norma szerinti felírása, pl. L_2 norma szerint nézve.

Relatív felírás A 2.1.5. állítás bizonyításának második felében egy olyan \mathbf{w} súlyvektort néztünk, amiben az első koordinátát egynek választottuk. Ez is egy lehetséges mód a felírásra: egy adott fix (első) koordinátát mindig egynek választunk (ha nem az első, akkor kicserélhetjük az elsőre), és a többi koordináta hozzá képest relatívan van mérve.

$$\mathbf{w} = (1, w_2, w_3, \dots, w_n)$$

Például ha $w_j > 1$, akkor tudjuk, hogy a j -edik alternatíva jobb mint az első.

2.1.2. Egy 3×3 -as példa

A gyenge hatékonyság valóban gyengítése a hatékonyságnak – ha egy súlyvektor hatékony, akkor gyengén hatékony is, a következő példán látható, hogy ez megfordítva nem feltétlenül igaz.

A következő példa arra is hasznos lesz a továbbiakban, hogy egy kis dimenziós példán megfigyeljük, hogy egy adott súlyvektorhoz nem csak véges sok szigorúan javított hatékony súlyvektor tartozhat.

2.1.10. Példa.

$$\mathbf{A} = \begin{pmatrix} 1 & 3 & 9 \\ 1/3 & 1 & 4 \\ 1/9 & 1/4 & 1 \end{pmatrix},$$

$$\mathbf{w} = \begin{pmatrix} 4 \\ 2 \\ 1 \end{pmatrix} \quad \left[\begin{array}{c} w_i \\ w_j \end{array} \right] = \begin{pmatrix} 1 & 2 & 4 \\ 0.5 & 1 & 2 \\ 0.25 & 0.5 & 1 \end{pmatrix},$$

A következő két hatékony súlyvektor szigorúan dominálja \mathbf{w} -t:

$$\mathbf{w}' = \begin{pmatrix} 9 \\ 3 \\ 1 \end{pmatrix} \quad \left[\begin{array}{c} w'_i \\ w'_j \end{array} \right] = \begin{pmatrix} 1 & 3 & 9 \\ 1/3 & 1 & 3 \\ 1/9 & 1/3 & 1 \end{pmatrix},$$

$$\mathbf{w}'' = \begin{pmatrix} 9 \\ 4 \\ 1 \end{pmatrix} \quad \left[\begin{array}{c} w''_i \\ w''_j \end{array} \right] = \begin{pmatrix} 1 & 2.25 & 9 \\ 0.4444 & 1 & 4 \\ 1/9 & 1/4 & 1 \end{pmatrix},$$

Továbbá \mathbf{w}' -nek és \mathbf{w}'' -nek minden konvex kombinációja is hatékony, valamint szigorúan dominálja \mathbf{w} -t:

$$\mathbf{w}''' = \begin{pmatrix} 9 \\ x \\ 1 \end{pmatrix} \quad \left[\begin{array}{c} w'''_i \\ w'''_j \end{array} \right] = \begin{pmatrix} 1 & 9/x & 9 \\ x/9 & 1 & x \\ 1/9 & 1/x & 1 \end{pmatrix},$$

ahol $3 \leq x \leq 4$.

2.2. Súlyvektor-számítási módszerek

Formálisan egy súlyvektor-számítási módszer egy $g : \mathcal{PCM}_n \rightarrow \mathbb{R}_+^n$ függvény. Megköveteljük továbbá, hogy ha \mathbf{A} konzisztens, akkor a 2.1.5. állításbeli \mathbf{w} vektor legyen \mathbf{A} g -beli értéke, tehát a módszer az ismert pontnak, mint függvénynek egy kiterjesztése legyen.

Az imént bevezettük a gyenge hatékonyság fogalmát, és noha a hatékonyságot preferáljuk, és szeretnénk, ha ez az erősebb jellemzés lenne igaz rá, minimálisan elvárható egy adott módszertől, hogy mindig gyengén hatékony súlyvektort adjon.

Egy \mathbf{w} súlyvektor kinyerésére egy páros összehasonlítási mátrixból számos súlyozási módszer ismert az irodalomban, melyeket többek között [4]-ban összefoglalva lapozhat fel az olvasó (onnan származik a következő bevezetés).

Néhány összefoglaló található Golany és Kress [33], Choo és Wedley [20], Lin [39], Bajwa, Choo és Wedley [6], illetve Fedrizzi és Brunelli [30] munkáiban. A kutatásokból az körvonalazódik, hogy nem létezik egyetlen univerzális módszer, ami jobban teljesítene a többinél.

Egy lehetséges hozzáállás a hatékony súlyvektor megkeresésének távolságminimalizálási problémaként való felírása, mint például a legkisebb négyzetek módszere [29], ami mindig hatékony súlyvektor(oka)t eredményez. Léteznek olyan távolságminimalizáló feladatok, amik nem mindig adnak hatékony súlyvektort. Ilyen például a sajátvektor módszer [44], ami a páros összehasonlítási mátrix legnagyobb sajátértékéhez tartozó jobboldali sajátvektorát adja meg, és ez szintén egy (igaz, speciálisabb) távolságfüggvényt minimalizál (Fichtner [31, 32]).

Blanquero, Carrizosa és Conde [7] figyelte meg, hogy ez a sajátvektor nem mindig hatékony, annak ellenére hogy szükséges és elégséges feltételek még nem ismertek.

További, jónak ígérkező módszerek a feszítőfás módszerek, amelyekről először Tsyganok [51, 52] írt. Ez a módszer figyelembe veszi az összehasonlítások $n - 1$ elemű összefüggő részalmazait, tehát az n^{n-2} darab feszítőfát abban a gráfban, ahol a mátrixelemek származtatják a gráf éleit az n csúc között. Minden feszítőfa egy-egy súlyvektort definiál, amit aztán megfelelő módon aggregálunk: számtani közép ([47, 48, 51, 52], AMAST (arithmetic mean of all spanning tree weight vectors)), vagy geometriai közép szerint [19, 40]. Míg az AMAST súlyvektor nem mindig hatékony (ld. 4. fejezetet), addig a geometriai középről bebizonyították, hogy ekvivalens [19, 40] a logaritmikus legkisebb négyzetek módszerével (mellyel foglalkoztak [23, 24, 27, 42]), ami mindig hatékony súlyvektort számol ki [7, Corollary 7].

A sajátvektor és a logaritmikus legkisebb négyzetek módszere megegyezik a 3×3 -as esetben [23, 24], így a nemtriviális esetek 4 vagy annál nagyobb dimenziósak.

A koszinuszos maximalizáló módszer [38] azon a geometriai elgondoláson alapszik, hogy a vektorok hasonlóságát a bezárt szögük is jellemzi, vagy azok koszinusza, azaz ha a skaláris szorzatuk nagy. Az egyszerűen kiszámolható egyértelmű megoldást Kou és Lin publikálta [38, Theorem 2]. A geometriai intuíció ellenére ez a vektor is lehet nem hatékony, ahogyan ezt a 4. fejezetben látni is fogjuk.

További súlyozási módszerek ismertetése található Rapcsák Tamás jegyzetének [43]

I.2.2. alfejezetében, Ábele-Nagy Kristóf értekezésének [1] 2.3.4 alfejezetében, valamint Komáromi Éva cikkében [37].

Ebben a dolgozatban a következő módszerekkel foglalkoztunk módszeresen.

2.2.1. Sajátvektor módszer

Ez az egyik legelterjedtebb, és legtöbb ideje vizsgált módszer. Az AHP, azaz az Analytic Hierarchy Process többszempontú döntési modellje is ezt használja.

A módszer egy $\mathbf{A} \in \mathcal{PCM}_n$ -hez az \mathbf{A} mátrix legnagyobb sajátértékéhez tartozó jobboldali sajátvektort jelöli meg súlyvektorként [44].

A módszer alapját szolgáltatja a következő tétel, mely bár nem garantálja a legnagyobb sajátértékhez tartozó sajátvektor hatékonyságát, de a gyengébb tulajdonságot igen:

2.2.1. Lemma. *Ha $\mathbf{A} \in \mathcal{PCM}_n$ konzisztens, akkor a sajátvektor módszer által megjelölt súlyvektor megegyezik az 2.1.5. állításbeli súlyvektorral.*

Bizonyítás: Ha \mathbf{A} konzisztens, akkor a rangja 1, mivel egy tetszőleges i . sor megegyezik az első sor $\frac{w_1}{w_i}$ -szeresével (ez következik a konzisztenciából). Mivel a rangja 1, ezért egyetlen nemnulla sajátértéke van, aminek meg kell egyeznie a mátrix nyomával. A főátlóban mindenhol egyesek állnak, ezért a legnagyobb sajátérték n , így $\mathbf{A}\mathbf{w} = n\mathbf{w}$.

A bal oldal i -edik koordinátája:

$$(\mathbf{A}\mathbf{w})_i = \sum_{j=1}^n a_{ij}w_j = \sum_{j=1}^n \frac{w_i}{w_j}w_j = \sum_{j=1}^n w_i = nw_i$$

□

A következő tételt Bozóki Sándor és Fülöp János bizonyította, amely fontos eredménye a súlyvektorok hatékonyság-vizsgálatával kapcsolatos kutatásoknak.

2.2.2. Tétel. [16, 3.1 Tétel] *A legnagyobb sajátértékhez tartozó sajátvektor gyengén hatékony.*

2.2.3. Állítás. *A legnagyobb sajátértékhez tartozó sajátvektor nem mindig hatékony.*

Bizonyításnak a dolgozat végére mellékeltem a 6.1.2. példát, amelyben egy sajátvektor módszer által kapott nem hatékony súlyvektor szerepel annak javításával és a javított mátrixelemekkel együtt. Mint ahogy azt korábban említettük az első példát Blanquero, Carrizosa és Conde adta [7].

2.2.2. Legkisebb négyzetek módszere

A legkisebb négyzetek módszere (Least Squares Method, röviden LSM [12, 21, 34]) a következő probléma megoldását javasolja súlyvektornak:

2.2.4. Definíció.

$$\begin{aligned} \min \sum_{i,j} \left(a_{ij} - \frac{w_i}{w_j} \right)^2 \\ \sum_i w_i = 1 \\ w_i > 0, \quad i = 1, 2, \dots, n. \end{aligned} \tag{2.7}$$

Ez a probléma egy általában nemkonvex optimalizálási probléma, melynek több lokális (sőt akár több globális) optima is lehet.

2.2.3. Logaritmikus legkisebb négyzetek módszere

A logaritmikus legkisebb négyzetek (Logarithmic Least Squares Method, vagy LLSM [23, 24, 27]) problémát a következőképp definiálhatjuk:

2.2.5. Definíció.

$$\begin{aligned} \min \sum_{i,j} \left[\log a_{ij} - \log \left(\frac{w_i}{w_j} \right) \right]^2 \\ w_i > 0, \quad i = 1, 2, \dots, n. \end{aligned} \tag{2.8}$$

Itt tehát a legkisebb négyzetek módszerével ellentétben a logaritmusok különbségét próbáljuk minimalizálni négyzetösszeg értelemben.

Mindkét módszer adaptálható a nem teljesen kitöltött esetre, ekkor a szumma kevesebb mint $\frac{n(n-1)}{2}$ elemet tartalmaz, ott csak az ismert a_{ij} -kre vizsgáljuk a különbségeket.

Két kapcsolódó tételt emelünk ki:

2.2.6. Állítás. [26, Definition 2.5] A (2.8) feladat megoldása megkapható expliciten a következő formulával:

$$w_i^{LLSM} = \sqrt[n]{\prod_{k=1}^n a_{ik}}$$

2.2.7. Tétel. ([7, Corollary 7]) A (2.8) feladat által adott súlyvektor hatékony.

2.2.4. Feszítőfás módszer

Tekintsük a K_n teljes gráfot, ahol n a mátrix sorainak száma. A gráf egy feszítőfájához a következő módon feleltetünk meg egy súlyvektort: a feszítőfa élei eredeti mátrixra visszavetítve kijelölnek néhány elem párt, amik a súlyvektorra vonatkozó megkötéseknek feleltethetők meg. Ezen $n - 1$ megkötés által kaptunk egy normálástól eltekintve egyértelmű súlyvektort. Az összes feszítőfához tartozó súlyvektort pedig valamilyen módszerrel átlagolva kaphatjuk a feszítőfák egyenkénti súlyvektoraiból.

Ezen módszert is vizsgálták már korábban ([19, 40, 47, 51, 52]), például a geometriai középéről a következő állítást fogalmazhatjuk meg:

2.2.8. Tétel. ([40]) *A feszítőfák súlyainak geometriai közepéből kapott vektor megegyezik a logaritmikus legkisebb négyzetek módszerével kapott megoldással.*

2.2.9. Következmény. (2.2.7.-ből) *A fenti súlyvektor hatékony is.*

A geometriai közép mellett a feszítőfákból számolt súlyvektorok számtani középpel történő átlagolása is ismert az irodalomban. Kaphatunk zárt, explicit formulát egy adott dimenzióban az eredeti a_{ij} elemekre vonatkozóan, bár ezek rendszerint egyre összetettebb képleteket adnak. Adott n dimenzióra az összes feszítőfán végigiterálni pedig a gyakorlatban nehezen kivitelezhető.

2.2.5. Koszinuszos módszer

Szokásos hozzáállás az optimalizálási problémáknál, hogy a feladatot valamilyen távolság-minimalizálási problémaként írjuk fel. Egy ehhez hasonló elgondolásra alapul a koszinuszos módszer, ahol egy olyan súlyvektort keresünk, ami a többi vektor által bezárt szögek összegének minimalizálására törekszik.

Egy újabb módszer az ún. koszinuszos módszer [38]. A következő formula adja meg a $\mathbf{w} = (w_1, w_2, \dots, w_n)^T$ vektor i -edik koordinátáját:

$$w_i = \sum_{j=1}^n b_{ij} / \sqrt{\sum_{k=1}^n \left(\sum_{j=1}^n b_{kj} \right)^2}, \quad \text{ahol } b_{ij} = a_{ij} / \sqrt{\sum_{k=1}^n a_{kj}^2}$$

2.3. Indexek az inkonzisztencia mérésére

Habár a konzisztencia bináris tulajdonság, vannak mátrixok, melyekről úgy érezhetjük, hogy *nagyon távol* vannak attól, hogy konzisztenseknek nevezhessük őket. Egy egyszerű példa, amikor a szempontok/alternatívák körbeverik egymást ($a_{ij} > 1$, $a_{jk} > 1$ és $a_{ik} > 1$ egyszerre teljesül), tehát ha bármilyen sorrendet állítunk is fel az i, j és k szempont/alternatíva között a w súlyvektor által, valamelyik mátrixelemet biztosan 1-nél kisebb értékkel fogjuk közelíteni. Ebben a fejezetben ezen fogalom precíz matematikai bevezetéséről lesz szó.

Sok inkonzisztencia indexet definiáltak, pl. Fedrizzi és szerzőtársai munkáiban [8, 9], valamint az inkonzisztenciáról egy részletes magyar munkát találhat az olvasó Poesz Attila [41] és Ábele-Nagy Kristóf [1] disszertációiban. A következőkben az utóbbi két forrás segítségével adunk áttekintést.

2.3.1. CR

A Consistency Ratio vagy röviden *CR* Saaty eredeti ötlete volt [44]. Először definiáljuk a Consistency Index-et (*CI*), mint segédváltozót:

2.3.1. Definíció. Egy adott $\mathbf{A} \in \mathcal{PCM}_n$ mátrix konzisztencia indexét a következő képlet határozza meg:

$$CI_n = \frac{\lambda_{\max} - n}{n - 1}$$

ahol λ_{\max} az \mathbf{A} legnagyobb sajátértéke.

Mivel $\lambda_{\max} \geq n$, ezért $CI_n \geq 0$. Ez az érték önmagában is hasznos, érdemes összevetni egy adott dimenzióra nézve az átlagos CI_n -nel. Az AHP módszertanában ismert elemekből (tipikusan $a_{ij} \in \{1, 2, 3, \dots, 8, 9, \frac{1}{2}, \frac{1}{3}, \dots, \frac{1}{9}\}$, de ettől eltérhetünk) véletlenszerűen generált $n \times n$ mátrixok átlagos λ_{\max} -ához viszonyítunk – ezt jelöljük $\overline{\lambda_{\max}}$ -al.

RI_n -nel jelöljük azt az értéket, amit akkor kapunk, ha a 2.3.1. definícióban megadott képletbe $\overline{\lambda_{\max}}$ -ot helyettesítünk λ_{\max} helyére.

2.3.2. Definíció. Egy adott $\mathbf{A} \in \mathcal{PCM}_n$ mátrix CR -jét (Consistency Ratio, inkonzisztencia index) a következő formula adja meg:

$$CR(\mathbf{A}) = \frac{CI_n(\mathbf{A})}{RI_n}$$

A sokak által kritizált heurisztika alapján azokat a mátrixokat tüntetjük ki, melyekre $CR < 0.1$ teljesül. Őket nevezzük elfogadható inkonzisztenciájú mátrixoknak. Ez a korlát önkényes, sok érv szól mellette és ellene, mindemellett a λ_{\max} meghatározása sem teljesen indokolt, hiszen ez az érték függ attól, hogy az a_{ij} -ket milyen halmazból választjuk ki.

2.3.2. CM

A Consistency Measure-t (CM) Koczkodaj vezette be [36]. A CM ún. triádos elgondolásra épül.

2.3.3. Definíció. Adott $\mathbf{A} \in \mathcal{PCM}_n$ esetén *triádnak* nevezzük azt a 3×3 -as részmátrixot, amely az \mathbf{A} mátrixból három i, j, k index által meghatározott oszlopok és sorok metszeteiből áll össze.

Adott i, j, k indexek esetén jelöljük ezt $\mathbf{A}^{i,j,k}$ -val.

Nyilvánvalóan erre a részmátrixra teljesül, hogy $\mathbf{A}^{i,j,k} \in \mathcal{PCM}_3$.

Triádokra nagyon egyszerűen megválaszolhatóak olyan kérdések, mint pl. ha két elemét lefixáljuk, minimum mennyire kell megváltoztatnunk a harmadikat, hogy egy konzisztens mátrixot kapjunk. Ezen három szám minimuma egy a triádra jellemző érték, és egy egyszerű képlettel megadható. A mátrix CM -je ezen triádokra számolt érték maximuma lesz (ami arra az $\mathbf{A}^{i,j,k}$ triádra vétetik fel, ahol az a_{ij}, a_{jk} és a_{ik} ezen szempont alapján a legtávolabb vannak).

Egy adott $\mathbf{T} \in \mathcal{PCM}_3$ triád esetén

$$\mathbf{T} = \begin{pmatrix} 1 & a & b \\ 1/a & 1 & c \\ 1/b & 1/c & 1 \end{pmatrix}$$

A következő formula adja meg a CM értékét [28]:

$$CM(\mathbf{T}) = \min \left\{ \left| 1 - \frac{b}{ac} \right|, \left| 1 - \frac{ac}{b} \right| \right\}$$

A maximum kiszámításakor feljegyzett i, j, k indexek rámutatnak a probléma helyére, polinomiálisan számolható ($\binom{n}{3}$ triád van, ahol n általában alacsony), viszont csak a legrosszabb triádot veszi figyelembe, azaz globálisan nem érzékeny.

2.3.3. GCI

A Geometric Consistency Indexet (GCI) Aguarón és Moreno-Jimenez vezette be [3], és a logaritmikus legkisebb négyzetek problémához kötődik.

2.3.4. Definíció. A következő képlettel adható meg a GCI index:

$$GCI(\mathbf{A}) = \frac{2}{(n-1)(n-2)} \sum_{i=1}^n \sum_{j=1}^n \left(\log a_{ij} - \log \frac{w_i}{w_j} \right)^2$$

Ugyan ez a definíció függ a \mathbf{w} súlyvektortól, de ha pl. az LLSM probléma megoldását használjuk (\mathbf{w}^{LLSM}), akkor egy jól definiált mértéket kapunk.

Ezen indexre is definiáltak elfogadhatósági küszöbök, ezek ismertetésétől eltekintünk.

2.3.4. Távolságminimalizáló indexek

A legközelebbi konzisztens mátrixtól való távolságként is definiálhatjuk azt a feladatot, hogy inkonzisztencia mértéket rendeljünk adott $\mathbf{A} \in \mathcal{PCM}_n$ mátrixokhoz. A konzisztens mátrixokhoz egyértelműen rendelhető súlyvektor, így a távolságminimalizálendő képlet a súlyvektor függvényébe is átírható, tehát elég a súlyvektor függvényében keresni illet. Például a legkisebb négyzetes célfüggvény (2.7) optimális értéke is egy inkonzisztencia index.

2.4. A CR korlátjai

A leggyakrabban használt index a CR , így a példákban ezt az indexet fogjuk vizsgálni.

A következő példában azt reprezentáljuk, hogy a tetszőlegesen kicsi CR esetén is születhet nem Pareto-hatékony súlyvektor. (Ugyanakkor később látni fogjuk, hogy más összefüggés érvényes, ha a mátrix CR -ja szerint nézzük a súlyvektor számítási módszerek hatékonyságát (ld. 4. fejezet).)

2.4.1. Példa. A következő példa [13]-ból származik. Tekintsük az alábbi parametrikus

páros összehasonlítás mátrixot:

$$\mathbf{A}(p, q) = \begin{pmatrix} 1 & p & p & p & \dots & p & p \\ 1/p & 1 & q & 1 & \dots & 1 & 1/q \\ 1/p & 1/q & 1 & q & \dots & 1 & 1 \\ \vdots & \vdots & \vdots & \ddots & & \vdots & \vdots \\ \vdots & \vdots & \vdots & & \ddots & \vdots & \vdots \\ 1/p & 1 & 1 & 1 & \dots & 1 & q \\ 1/p & q & 1 & 1 & \dots & 1/q & 1 \end{pmatrix} \in \mathcal{PCM}_n,$$

ahol $n \geq 4, p > 0$ és $1 \neq q > 0$.

Erről a mátrixról belátható [13, Proposition 1.], hogy CR -je tetszőlegesen kicsi lehet, ahogy $q \rightarrow 1$, viszont domináns sajátvektora nem hatékony.

Az alábbi példa nemcsak azt demonstrálja, hogy tetszőlegesen nagy CR esetén is lehet hatékony súlyvektort kapni, hanem egy éles korlátot kaphatunk a páros összehasonlítás mátrixok λ_{\max} -ára.

2.4.2. Példa. A következő példa [5]-ből származik.

Tekintsük a következő mátrixot páratlan n -re:

$$\mathbf{B} = \begin{pmatrix} 1 & 9 & \frac{1}{9} & 9 & \dots & \frac{1}{9} \\ \frac{1}{9} & 1 & 9 & \frac{1}{9} & \dots & 9 \\ 9 & \frac{1}{9} & 1 & 9 & \dots & \frac{1}{9} \\ \vdots & \vdots & & \ddots & & \vdots \\ \vdots & \vdots & & & \ddots & \vdots \\ 9 & \frac{1}{9} & 9 & \dots & \frac{1}{9} & 1 \end{pmatrix} \in \mathcal{PCM}_n$$

valamint ugyanennek a struktúrának a paraméteres változatát, amelyben minden 9-es helyére S , és $\frac{1}{9}$ helyére $\frac{1}{S}$ kerül ($S \geq 1$).

2.4.3. Tétel. [5, Proposition 2]

Adott $S \geq 1$ -re legyen $\mathbf{A} \in \mathcal{PCM}_n$, hogy $\forall 1 \leq i, j \leq n : 1/S \leq a_{ij} \leq S$. Akkor legyen λ_{\max} a legnagyobb (pozitív, valós a Perron-Frobenius tétel miatt) sajátértéke. Ekkor $n \leq \lambda_{\max} \leq 1 + \frac{1}{2}(n-1)(S + \frac{1}{S})$ teljesül. Az egyenlőtlenség éles: az alsó korlát teljesül, ha \mathbf{A} konzisztens, a felső korlát pedig ha \mathbf{A} megegyezik a 2.4.2. példa szerint definiált \mathbf{B} mátrix paraméteres változatával, ahol a paraméter S .

A következőkben azt fogjuk belátni, hogy egy adott n páratlan dimenzióban a max CR értéke csak RI_n értékétől függ.

2.4.4. Következmény. A szokásos ($a_{ij} \in \{1, 2, \dots, 9, \frac{1}{2}, \dots, \frac{1}{9}\}$) elemű páros összehasonlítási mátrixra $CI_n \leq 3.55$.

Bizonyítás: A maximumot 2.4.2. példában definiált \mathbf{B} -nél veszi fel. Emlékezve a 2.3.1. definícióra, továbbá mivel $S = 9$, ezért:

$$\begin{aligned} CI_n(\mathbf{B}) &= \frac{\lambda_{\max}(\mathbf{B}) - n}{n - 1} = \frac{1 + \frac{1}{2}(n - 1)(S + \frac{1}{S}) - n}{n - 1} = \frac{1 + \frac{1}{2}(n - 1)(9 + \frac{1}{9}) - n}{n - 1} = \\ &= \frac{1 + 4.5\dot{5}(n - 1) - n}{n - 1} = \frac{4.5\dot{5}(n - 1) - (n - 1)}{n - 1} = \frac{3.5\dot{5}(n - 1)}{n - 1} = 3.5\dot{5} \end{aligned}$$

□

Tehát valóban, a \mathbf{B} mátrix esetén elérjük a $CI_n = 3.5\dot{5}$ korlátot, és ott vétetik fel a maximális CR . Megelőlegezve a 3.2.3. példa állítását, mely szerint \mathbf{B} sajátvektora $(\frac{1}{n}, \frac{1}{n}, \dots, \frac{1}{n})$, belátható, hogy (az EM által adott súlyvektor) hatékony is lesz. De például a COS módszer esetén ez analitikusan a képlet alapján is levezethető:

2.4.5. Állítás. *Legyen n továbbra is páratlan. A koszinuszos módszer a \mathbf{B} mátrixra L_1 normálás után a $\mathbf{w} = (\frac{1}{n}, \frac{1}{n}, \dots, \frac{1}{n})$ súlyvektort adja.*

Bizonyítás: Vezessük be a következő jelöléseket. Tetszőleges j -re:

$$\begin{aligned} c_1 &= \sum_{i=1}^n a_{ij} = 1 + \frac{n-1}{2} \cdot \frac{1}{9} + \frac{n-1}{2} \cdot 9 \\ c_2 &= \sum_{i=1}^n a_{ij}^2 = 1 + \frac{n-1}{2} \cdot \left(\frac{1}{9}\right)^2 + \frac{n-1}{2} \cdot 9^2 \end{aligned}$$

Tehát egy sor vagy oszlop összege illetve négyzetes összege konstans, és nem függ az indextől.

$$b_{ij} = a_{ij} / \sqrt{\sum_{k=1}^n a_{kj}^2} = \frac{a_{ij}}{\sqrt{c_2}}$$

Ezt felhasználva:

$$\begin{aligned} w_i &= \sum_{j=1}^n b_{ij} / \sqrt{\sum_{k=1}^n \left(\sum_{j=1}^n b_{kj}\right)^2} = \sum_{j=1}^n \frac{a_{ij}}{\sqrt{c_2}} / \sqrt{\sum_{k=1}^n \left(\sum_{j=1}^n \frac{a_{kj}}{\sqrt{c_2}}\right)^2} = \\ &= \sum_{j=1}^n \frac{a_{ij}}{\sqrt{c_2}} / \frac{1}{\sqrt{c_2}} \sqrt{\sum_{k=1}^n \left(\sum_{j=1}^n a_{kj}\right)^2} = \sum_{j=1}^n a_{ij} / \sqrt{\sum_{k=1}^n \left(\sum_{j=1}^n a_{kj}\right)^2} \\ &= \sum_{j=1}^n a_{ij} / \sqrt{\sum_{k=1}^n c_1^2} = c_1 / c_1 \sqrt{\sum_{k=1}^n 1} = \frac{1}{\sqrt{n}} \end{aligned}$$

□

Megjegyzés: a páratlan dimenzió feltétele valóban szükséges, numerikus példák igazolják, hogy pl. az előző állítás sem igaz páros dimenzióra.

Bár a CI_n maximumát mindig ugyanott éri el (páratlan n -ekre), az átlagos CI_n , amit RI_n -nel jelöltünk változik az n függvényében. A következő táblázatban írtuk fel az értékeket kis n -ekre, felhasználva [18] eredményeit.

Dimenzió	$\max CI_n$ becslése	$\overline{\lambda_{\max}}$	RI_n	$\max CR$ becslése
3	3.5555	4.0484	0.5242	6.783
4	< 3.5555	6.6525	0.8842	<4.021
5	3.5555	9.4347	1.1087	3.207
6	<3.5555	12.244	1.2488	<2.847
7	3.5555	15.045	1.3408	2.652

2.1. táblázat. $\max CI_n$, $\max CR$, $\overline{\lambda_{\max}}$ és RI_n becslése különböző dimenziók esetén

3. fejezet

Hatékonyság ellenőrzése

Ebben a fejezetben arra a kérdésre szeretnénk algoritmust találni, hogy adott $\mathbf{A} \in \mathcal{PCM}_n$, és hozzá tartozó \mathbf{w} súlyvektorra a súlyvektor hatékony-e, és ha nem esetleg kaphatunk-e őt domináló hatékony súlyvektort.

3.1. Lineáris programozási feladat

Lineáris programozási feladatként való felírás az első megközelítés, melyet Bozóki és Fülöp írt le először [16]. Bevezetve a következő jelöléseket:

$$I = \left\{ (i, j) \mid a_{ij} < \frac{w_i}{w_j} \right\}$$
$$J = \left\{ (i, j) \mid a_{ij} = \frac{w_i}{w_j}, i < j \right\}$$

Az \mathbf{A} mátrix pontosan akkor konzisztens, ha az I halmaz üres. Ebben az esetben a \mathbf{w} súlyvektor hatékony és $|J| = n(n-1)/2$. A továbbiakban feltesszük, hogy I nem üres. Tekintsük az alábbi nemlineáris optimalizálási feladatot:

$$\begin{aligned} \min \quad & \prod_{(i,j) \in I} t_{ij} \\ & \frac{x_j}{x_i} a_{ij} \leq 1 \quad \forall (i, j) \in I, \\ & \frac{x_i}{x_j} \frac{w_j}{w_i} \frac{1}{t_{ij}} \leq 1 \quad \forall (i, j) \in I, \\ & 0 < t_{ij} \leq 1 \quad \forall (i, j) \in I, \\ & a_{ji} \frac{x_i}{x_j} = 1 \quad \forall (i, j) \in J, \\ & x_1 = 1. \end{aligned} \tag{3.1}$$

Amely feladatnak a változói: $x_i > 0$, $1 \leq i \leq n$ és t_{ij} , $(i, j) \in I$.

A következő tétel szolgál alapul a hatékonyság ellenőrzésének:

3.1.1. Állítás. [16, Proposition 4.1]

A (3.1) optimalizálási feladat optimumértéke legfeljebb 1, és pontosan akkor 1, ha a \mathbf{w} súlyvektor hatékony. Ha \mathbf{w} nem hatékony, akkor az $\mathbf{x} = (x_1, \dots, x_n)$ súlyvektor hatékony és dominálja a \mathbf{w} vektort.

Bár 3.1 egy nemlineáris optimalizálási feladat, egy vele ekvivalens lineáris optimalizálási feladattá alakítható át:

Legyenek

$$\begin{aligned} y_i &= \log x_i & \forall i : 1 \leq i \leq n; \\ v_i &= \log w_i & \forall i : 1 \leq i \leq n; \\ s_{ij} &= -\log t_{ij} & \forall (i, j) \in I; \\ b_{ij} &= \log a_{ij} & \forall i, j : 1 \leq i, j \leq n. \end{aligned}$$

A szigorúan monoton növekvő logaritmus függvényt alkalmazva a célfüggvényre és a feltételek két oldalára, egy ezzel ekvivalens lineáris optimalizálási feladatot kapunk:

$$\begin{aligned} \min \sum_{(i,j) \in I} -s_{ij} \\ y_j - y_i &\leq -b_{ij} & \forall (i, j) \in I, \\ y_i - y_j + s_{ij} &\leq v_i - v_j & \forall (i, j) \in I, \\ y_i - y_j &= b_{ij} & \forall (i, j) \in J, \\ s_{ij} &\geq 0 & \forall (i, j) \in I, \\ y_1 &= 0. \end{aligned} \tag{3.2}$$

A feladat változói pedig y_i , $1 \leq i \leq n$ és $s_{ij} \geq 0$, $(i, j) \in I$.

És kimondható a 3.1.1. állítással analóg kijelentés a logaritmizáltjára is.

3.1.2. Állítás. [16, Theorem 4.1]

A (3.2) lineáris programozási feladat optimumértéke legfeljebb 0, és pontosan akkor 0, ha a \mathbf{w} súlyvektor a (2.1) többcélű optimalizálási feladat hatékony megoldása. Jelölje $(\mathbf{y}^*, \mathbf{s}^*) \in \mathbb{R}^{n+|I|}$ a (3.2) feladat optimális megoldását. Ha \mathbf{w} nem hatékony, akkor az $\exp(\mathbf{y}^*)$ súlyvektor hatékony és dominálja ¹ a \mathbf{w} súlyvektort.

3.2. Egy algoritmus a domináló hatékony súlyvektorok keresésére

3.2.1. Gráfrepresentáció

A módszer alapja a következő definíció és tétel.

¹az eredeti cikkben *belülről* dominálás szerepelt, de abból következik a dominálás is, ld. [16, Corollary 1.]

3.2.1. Definíció. Legyen $\mathbf{A} \in \mathcal{PCM}_n$, és $\mathbf{w} = (w_1, w_2, \dots, w_n)^T$ egy ahhoz tartozó súlyvektor. A $G = (V, \vec{E})_{\mathbf{A}, \mathbf{w}}$ irányított gráfot a következőképp definiálhatjuk:

$$V = \{1, 2, \dots, n\}$$

$$\vec{E} = \left\{ i \rightarrow j \mid \frac{w_i}{w_j} \geq a_{ij}, i \neq j \right\}.$$

A definícióból következik, hogy ha $w_i/w_j = a_{ij}$, akkor létezik egy oda-vissza irányított él i és j között. Vizsgálataink alapja Blanquero, Carrizosa és Conde-nak egy fontos tétele:

3.2.2. Tétel. [7, 9. Tétel] *Legyen $\mathbf{A} \in \mathcal{PCM}_n$. Egy \mathbf{w} \mathbf{A} -hoz tartozó súlyvektor pontosan akkor hatékony, ha $G = (V, \vec{E})_{\mathbf{A}, \mathbf{w}}$ egy erősen összefüggő digráf.*

A hatékony súlyvektor gráfokon alapuló jellemzését fogjuk használni a továbbiakban.

3.2.3. Példa. Térjünk vissza egy pillanatra a 2.4.2. példára! Belátható, hogy az EM által adott súlyvektor a következő lesz: $\mathbf{w} = (\frac{1}{n}, \frac{1}{n}, \dots, \frac{1}{n})$. Ezek után könnyen látható, hogy az EM által adott súlyvektor hatékony lesz, mert a gráf tartalmaz egy irányított Hamilton-kört $n \rightarrow n-1 \rightarrow n-2 \rightarrow \dots \rightarrow 2 \rightarrow 1 \rightarrow n$, ugyanis a 9 és $\frac{1}{9}$ elemeket mindig alul- vagy felülbecsüljük, így az $a_{i,i+1} = 9$ -ek alulbecslése miatt kapunk $i+1 \rightarrow i$ irányított éleket, és $a_{n,1} = 9$, így lesz egy $1 \rightarrow n$ irányított él is.

3.2.2. Az algoritmus

Egyenesen következik a hatékony súlyvektorok ezen jellemzéséből, hogy ha legyártjuk a fent említett digráfot, erős összefüggőségének tesztelésével meg tudjuk mondani $O(m)$, azaz élszám időben (teljes gráf lévén ez pedig $O(n^2)$ időt jelent), hogy a súlyvektor valóban hatékony-e.

Ennél egy lépéssel tovább is mehetünk: ha nem hatékony a súlyvektor, koordinátáinak alkalmas módosításával képesek vagyunk az élek vissza-párját is behúzni vagy esetleg az él irányítását megfordítani, így pedig befolyásolhatjuk a gráf erős összefüggőségét. Ezzel egy polinomiális algoritmust is kaptunk, amellyel egy hatékony súlyvektort nyerhetünk ki. Kicsit jobban odafigyelve arra, hogy mely koordinátákat és mennyivel változtatunk meg, azt is garantálhatjuk, hogy ha az eredeti súlyvektor nem volt hatékony, egy őt domináló hatékony súlyvektort kapjunk. Ezt vette észre tőlünk függetlenül Taji és Mizuno [49] is.

Ez az algoritmus hatásos eszközt ad a kezünkbe hatékony súlyvektor megtalálásához, súlyvektor hatékonyságának teszteléséhez, és nem hatékony súlyvektort domináló hatékony súlyvektor keresésére. Ezek a kérdések mind központi fontosságúak ebben a témakörben.

Algorithm 1 Erősen összefüggővé tétel algoritmusa

Input $\mathbf{A} \in \mathcal{PCM}_n$, G hozzá tartozó gráf, és \mathbf{w} tetszőleges súlyvektor

Output Egy boolean érték (hatékony-e a súlyvektor), és amennyiben nem, egy \mathbf{w}' őt domináló hatékony súlyvektor

```
1: procedure ALG( $\mathbf{A}, G, \mathbf{w}$ )
2:    $G$  erősen összefüggőségi komponensei:  $G_1, G_2, \dots, G_k$ . Legyen  $F : i \mapsto j$ , ha  $i \in G_j$ .
3:   if  $k = 1$  then return (true, -)
4:   for  $i = 1 \dots k - 1$  do ▷ Inicializálás
5:     for  $j = i + 1 \dots k$  do
6:        $l[i, j] = \infty, \quad u[i, j] = 0$ 
7:   for  $i = 1 \dots n - 1$  do
8:     for  $j = i + 1 \dots n$  do
9:        $(i, j) := (\arg \min \{F(i), F(j)\}, \arg \max \{F(i), F(j)\})$ 
10:       $l[F(i), F(j)] = \min \left\{ l[F(i), F(j)], \frac{w_j \cdot a_{ij}}{w_i} \right\}$ 
11:       $u[F(i), F(j)] = \max \left\{ u[F(i), F(j)], \frac{w_j \cdot a_{ij}}{w_i} \right\}$ 
12:    $\mathbf{w}' \leftarrow \mathbf{w}$ 
13:   while true do
14:      $t$  tetszőleges, hogy  $l[k - 1, k] \leq t \leq u[k - 1, k]$ 
15:     for  $i = 1 \dots n$  do
16:       if  $F(i) = n - 1$  then
17:          $w'_i \leftarrow t \cdot w_i$ 
18:     for  $i = 1 \dots k - 2$  do
19:        $l[i, k - 1] = t \cdot l[i, k - 1], \quad u[i, k - 1] = t \cdot u[i, k - 1]$ 
20:      $G' \leftarrow G$ 
21:     for  $i = 1 \dots n$  do
22:       if  $F(i) = n - 1$  then
23:         for  $j = 1 \dots n$  do
24:            $i$  és  $j$  csúcsok között frissítsük a  $G'$  digráf élet a  $\mathbf{w}'$  súlyvektor alapján
25:      $G'$  erősen összefüggő komponensei:  $G'_1, G'_2, \dots, G'_h$ . Legyen  $F' : i \mapsto j$ , ha  $i \in G'_j$ 
26:     if  $h = 1$  then return (false,  $\mathbf{w}'$ )
27:     for  $i = 1 \dots h - 1$  do
28:        $l[i, h] = \min_{j=h-1 \dots k} l[i, j], \quad u[i, h] = \max_{j=h-1 \dots k} u[i, j]$ 
29:      $G \leftarrow G', \quad k \leftarrow h, \quad F \leftarrow F'$ 
```

3.2.3. Az algoritmus elemzése

Az algoritmus leírása

Rövidebb formában megtalálható ez az algoritmus [49, 4. oldal]-ban, ami csak összefoglalja a lépéseket, itt viszont sokkal részletesebben, adatstruktúrális kérdésekig taglalva foglalkozunk vele.

Az algoritmus szubrutinként hívja meg az erősen összefüggő komponensekre való bontás algoritmusát. Ezzel külön nem foglalkozunk, a probléma megoldható $O(m)$ (élszám) időben. Ha bármikor az algoritmus futása során egyetlen erősen összefüggő komponensünk lenne, kilépünk a megfelelő outputtal.

A 4-6. sorok inicializálják az l alsó, és u felső korlátokat tároló tömböt. A 7-11. sorban töltjük fel kezdeti értékekkel – az l tömb (i, j) elemébe az az érték kerül, ami az i -edik és j -edik komponens közötti $e \rightarrow f$ élekre képzelt $\frac{w_f \cdot a_{ef}}{w_e}$ értékek minimuma, hasonlóképp pedig az u tömbbe ezek maximuma kerül. Ahelyett hogy ezeket mindig újra kiszámolnánk, fenntartjuk, és súlyok módosításakor, illetve komponensek összeolvasztásakor megfelelő módon fent tudjuk tartani a struktúra helyességét.

Az algoritmus végén, ha a \mathbf{w} súlyvektor nem volt hatékony, \mathbf{w}' -t fogjuk outputként visszaadni. A fő ciklusmag a 13-29. sorokban van. Az általánosság megszorítása nélkül feltehetjük, hogy az utolsó két indexű komponens olvastjuk össze ($k-1$ -et és k -t – ahol $k > 1$, mert különben kiléptünk volna). Ezen két index tudatában a 14. sorban választunk egy tetszőleges t értéket, melyre az adott kikötés érvényes – ez határozza meg, hogy mennyire változtatjuk meg a komponenshez tartozó i csúcsok w'_i súlyait. A t választásától függ az algoritmus további futása, és a végül kapott eredmény is, viszont a t -t tetszőlegesen megválaszthatjuk az intervallumból. Különböző stratégiákat lehet alkalmazni a megválasztásakor (pl. mindig a minimumot, maximumot, a kettő átlagát, vagy például az élekre írt értékek mediánját), de elméleti eredményt egyikhez sem sikerült eddig társítani.

A t paraméter megválasztása után a 15-17. sorokban megváltoztatjuk a \mathbf{w}' azon koordinátáit, amelyek az adott komponensben vannak, majd a 18-19. sorokban az alsó és felső korlátokat is eszerint módosítjuk. A leíráskor feltettük, hogy az él eredetileg $k-1 \rightarrow k$ futott, fordított esetben nem szorozni, hanem osztani kell a t értékkel. Ésszerű továbbá a kevesebb csúcsból álló komponens módosítani, ilyenkor ismét t -vel való osztás és szorzás szerepe megcserélődik.

A 20-24. sorokban frissítjük a gráfot (egyelőre G' egy másolatát), arra ügyelve, hogy a módosított \mathbf{w} elemei módosíthatták a digráf éleinek irányítását, vagy éppen az él mellett keletkezhetett egy vissza él is a hozzá tartozó mátrixelem tökéletes közelítése esetén.

A 25. sorban újra meghatározzuk az erősen összefüggő komponenseket, és kilépünk, ha csak egy maradna. A $k-1$ -edik és k -edik G -beli komponens összehúzásával kapott gráfban $k-1$ -nél sokkal kevesebb komponens is lehet a G' gráfban. Szintén az általánosság megsértése nélkül feltehetjük, hogy az utolsó $k-h$ komponens olvad össze, azaz a G -beli G_h, G_{h+1}, \dots, G_k komponensekből lesz a G'_h , utolsó komponens a G' gráfban.

Ezekre a komponensekre mind ki kell számolni a többi megmaradó (G'_1, \dots, G'_{h-1}) komponensek közötti l és u korlátokat – ezt tesszük meg a 27. és 28. sorban. A ciklust lezáró

29. sorban az ideiglenes vesszős változókat véglegesítjük, és új ciklust kezdünk.

Algoritmus helyességének bizonyítása

Az algoritmus helyességének bizonyítása megtalálható Taji és Mizuno eredeti cikkében [49, Theorem 1]. Lényegében azt látják be, hogy minden ciklusban $h - 1$. és h . összeolvasztásával csökken az erősen összefüggő komponensek száma, valamint a t ezen választásával $\left| \frac{w_i}{w_j} - a_{ij} \right| \geq \left| \frac{w'_i}{w'_j} - a_{ij} \right| \geq 0$ az algoritmus minden while ciklusában (13-29. sorok) teljesül. Sőt egy lépés után biztosan lesz olyan egyenlőtlenség, amely szigorúan teljesül, azaz \mathbf{w}' valóban dominálni fogja az eredeti súlyvektort.

Futásidő

Az algoritmusok elemzésénél általában, és itt különösen fontos, hogy egyrészt miből áll az input, másrészt pedig hogy ennek tükrében mi az az n , amihez képest viszonyítjuk a futásidőt. Ha n -nek a gráf csúcsainak számát, azaz a szempontok/alternatívák számát tekintjük, akkor ha a mátrixot elemenként, a gráf éleit pedig egyenként (és pl. nem impliciten) adjuk meg, akkor pusztán a mátrix, illetve a G gráf beolvasása külön-külön $O(n^2)$ futásidejű.

Azt a megállapítást tesszük, hogy n -nel a csúcsok számát, tehát az alternatívák/szempontok számát fogjuk nevezni. A gyakorlathoz alkalmazkodva feltesszük, hogy a mátrix értékei explicit meg vannak adva, ebből kifolyólag az input mérete $O(n^2)$.

Az algoritmus egyes elemei a következő futásidejűek:

1. A 4-6. sorokban inicializáljuk az l és u statikus tömb változókat. Az algoritmus elején $k \times k$ méretűek, és egyre kisebb részét használjuk, ahogy a komponensek száma csökken. Az értékekkel való feltöltés $O(k^2)$, legrosszabb esetben $O(n^2)$ helyet foglal és ugyanennyi a futási ideje.
2. A 7-11. sorban töltjük fel az alap értékekkel. Hasonlóképp $O(k^2)$ avagy legrosszabb esetben $O(n^2)$ a futásidő.
3. A 13-29. sorig tartó ciklusról nem tudjuk, hogy hányszor fut le, de mindegyik lépésben legalább két erősen összefüggő komponens összeolvad, tehát a komponensek száma legalább eggyel csökken. Emiatt és mivel legfeljebb n komponens lehet (mind-egyik csúcs külön komponens, n csúcsú tournamentet alkotva), legrosszabb esetben n -szer futunk végig cikluson. Megjegyezzük, hogy ez valóban be is következhet, ha n különálló komponensből indulunk, és ha minden alkalommal olyan szerencsétlenül választjuk a t értékét, hogy csak két különálló komponens sikerül összeolvasztani.
4. A 15-17. sor és a 18-19. sor for ciklusai korábbi indoklás miatt legfeljebb n lépést igényelnek (elemi lépések legfeljebb n -szer).
5. Külön vizsgálendő a 21-24. sor és a 27-28. for ciklusai. Az előbbinél azt figyelhetjük meg, hogy éleket csak az aktuálisan összehúzendó komponens és a többi komponens között frissíthetünk, tehát mindegyik élet legfeljebb egyszer dolgozzuk fel az algoritmus teljes lefutása során, és így a ciklusok számától függetlenül ez összesen maximum

$O(e) = O(n^2)$ idejű lehet. A 18-19. sorban lévő ciklusnál azt kell meglátni, hogy egy adott while ciklusban a minimum- és maximumkeresések alatt $k - h + 1$ elemet kell összehasonlítani, és azok az elemek többé nem kerülnek felhasználásra (akár törölhetjük is), mert elmentjük az $l[i, h]$ és $u[i, h]$ változókba a minimumot. Mivel eredetileg $O(n^2)$ ilyen elem van, ezért az összes ciklus alatt nem végezhetünk összesen több mint $O(n^2)$ lépést.

Tehát az algoritmus összesített futásideje $O(n^2)$.

3.2.4. Példák

Az algoritmust illusztrálja a Függelékben található 6.3.1. példa. A példák kapcsán a következő állítást is bizonyítani tudjuk majd:

3.2.4. Állítás. *Egy w súlyvektor bemenet esetén a t választásaitól és a komponensek összeolvasztásának sorrendjétől függetlenül az algoritmus nem feltétlenül képes megtalálni akár melyik őt domináló hatékony súlyvektort.*

Bizonyítás: Létezik ellenpélda: adott nem hatékony w súlyvektort domináló hatékony súlyvektort az algoritmus semmilyen futása nem képes kiadni. A bizonyítás a példához kötődő 6.3.2. állításban található meg. \square

3.3. Lineáris programozási feladat és a gráf alapú algoritmus összehasonlítása

Egy lineáris programozási feladat megoldásához valamilyen LP megoldó algoritmust használhatunk. Tipikus választás a szimplex módszer, de ahhoz képest a gráf algoritmus garantáltan polinomiális lesz, ugyanakkor a szimplex módszer akár exponenciális futásideje alacsonyabb n értékek a gyakorlatban elhanyagolható.

A gráf algoritmus akár több különböző súlyvektort is találhat, de ezzel párhuzamos az LP megoldása is változhat attól függően, hogy ha belsőpontos algoritmussal oldjuk meg, különböző kezdeti megengedett megoldásból indítva a megoldót.

A harmadik megjegyzendő dolog pedig az, hogy a gráf alapú algoritmusba beleláthatunk: explicit alsó és felső korlátjaink vannak, és ezért elemezhetjük, miért azt a súlyvektort adta ki, és így talán jobban beleláthatunk a probléma megjába.

4. fejezet

Numerikus eredmények

Ebben a fejezetben a kutatások néhány numerikus eredményét ismertetjük. Az Olvasó az 5. fejezetben mélyebb betekintést kaphat az eszköztárba, amelyet felhasználtunk, de ebben a fejezetben az eredményekre összpontosítunk (amelyek függetlenek a *hogyan* kérdéstől).

4.1. Általános megjegyzések

A 2.2. alfejezetben ismertetett módszerek közül azokat vizsgáltuk, amelyek nem mindig adnak hatékony megoldást – ezek: a sajátvektor módszer, a számtani középpel aggregált feszítőfás módszer, valamint a koszinuszos módszer. A továbbiakban rendre *EM*, *AMAST* és *COS* rövidítésekkel fogjuk ezeket a módszereket jelölni.

Mint ahogy azt már korábban említettük, a 3×3 -as mátrixok esetében az *EM* módszer megegyezik az *LLSM* megoldásával [24], így szükségképpen hatékony súlyvektort ad az *EM* minden három dimenziós páros összehasonlítás mátrixra.

A páros összehasonlítás mátrixok és a hozzá tartozó súlyvektorok permutációfüggetlenek olyan értelemben, hogy (gyenge) hatékonyság és egyéb jellemzők nem változnak, ha két adott i, j indexre a mátrix i -edik és j -edik sorát, majd oszlopát megcseréljük, majd a súlyvektor i -edik és j -edik koordinátáját cseréljük meg.

4 dimenziótól kezdve vizsgáltuk a mátrixokat. A 4×4 -es mátrixokat teljesen karakterizálni is tudtunk: konkrét listánk van a mátrixokról és azok jellemzőiről, a módszerek által adott súlyvektorokról, és arról is, hogy ezek közül melyek hatékonyak, és melyek nem (ld. 6.2.1. példa). A hatékonyság mellett továbbá más tulajdonságait is vizsgáltuk, ld. pl. 4.3. alfejezetet a főátló feletti egynél nagyobb elemekről.

Jelentős energiabefektetés ellenére sem sikerült teljesen lefuttatni a készített programokat 5 és annál magasabb dimenziós mátrixokra erőforrás hiánya miatt (a kombinatorikus robbanás szuperexponenciális kitevője miatt).

Adott X halmazból generált permutációsűrt mátrixok számát alulról becsljük a következő-

4	$\approx 1.01 \cdot 10^6$
5	$\approx 1.68 \cdot 10^{10}$
6	$\approx 3.98 \cdot 10^{15}$
7	$\approx 1.37 \cdot 10^{22}$

4.1. táblázat. A generált mátrixok számára vonatkozó alsó becslés 4-7 dimenziókra

képpen. Elég a páros összehasonlítás mátrix felső háromszög részét kitölteni: ez n dimenziós mátrix esetén $|X|^{n(n-1)/2}$ mátrixot jelent. Egy adott ilyen mátrixnak oszlopait és sorait egyszerre permutáljuk. Mivel a dimenzió n , $n!$ ilyen permutáció lehetséges, tehát feltéve hogy minden permutáció $n!$ -szor szerepel, a mátrixok számának ezt a képletet kapjuk: $\frac{|X|^{n(n-1)/2}}{n!}$. Ugyanakkor vannak páros összehasonlítás mátrixok, pl. a csupa 1-esből álló mátrix, aminek (összes) permutáltja megegyezik önmagával, így az $n!$ -sal való leosztás hibás. Viszont mivel ekkor is leosztunk, ezért egy alsó becslést kaptunk. Ezek száma néhány alacsony dimenzió esetén a szokásos X halmazra ($X = \{1, 2, \dots, 9, \frac{1}{2}, \dots, \frac{1}{9}\}$, $|X| = 17$) látható a 4.1 ábrán.

A 4×4 -es mátrixokra generált összesítő táblázat a mátrixokat permutáltaktól megszűrve tartalmazza, tipikusan a lexikografikus értelemben vett minimummal feljegyezve. Ebben a táblázatban találhatóak mátrixok, melyekre például a szempontoknak/alternatíváknak már a sorrendje sem egyezik meg egy adott módszer által megjelölt nem hatékony súlyvektor, és annak a (3.2) lineáris program által megadott, öt domináló súlyvektora között.

4.2. Súlyvektorok hatékonyságának eloszlása különböző módszerek esetén

4.2.1. 4 dimenziós eset

Mint azt említettük, az összes permutációszűrt 4×4 -es mátrixot összegyűjtöttük. Számuk pontosan 1007097 (ez egyébként a 4.1. alfejezetben szereplő $\frac{17^6}{4!}$ becsléshez képest 1.36% hibájú). A $CR < 0.1$ korlátot pontosan 32157 mátrix teljesítette ezek közül (ami 3.19%-a a 10 milliós populációnak). A közel 32 ezer mátrixra alkalmaztuk a vizsgálandó módszereket, és azt kaptuk, hogy az *EM* 591 (1.84%), az *AMAST* 197 (0.61%), a *COS* pedig 602 (1.87%) mátrix esetében nem adott hatékony súlyvektort. Ezek az eredmények szerepelnek [4]-ben is.

Megjegyzés: az nem igaz, hogy az egyik módszer jobb lenne a másiknál (pl. nem hatékony súlyvektort ad, ha a másik nem hatékony súlyvektort adott), de relatív gyakoriságban szignifikánsak a különbségek.

4.2.2. Magasabb dimenzió

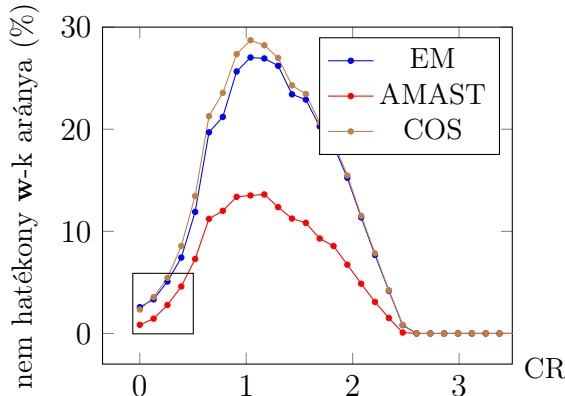
4-nél magasabb dimenzió esetén mindannyiszor találtunk olyan mátrixokat, melyekre az egyik módszer hatékony súlyvektort adott, de a többi nem, valamint olyat is, amelyre egy adott módszer nem adott hatékony súlyvektort, de a többi igen. Ezt 7 dimenzióig ellenőriztük.

Összefoglalóként a 4.4 ábrán láthatóak az eredmények.

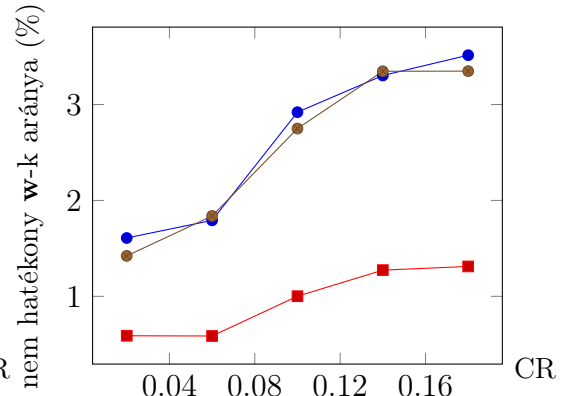
Míg a 4 dimenziós esetben az összes mátrixot le tudtuk generálni, addig a 4.1. alfejezetben leírtak miatt magasabb dimenzió esetén csak véletlen generált mátrixokra tudunk hagyatkozni. A módszer hasznos például hisztogram és egyéb ábrák készítésében, amikor

a CR függvényében akarunk kirajzolni egy arányt, de nem ad teljesen pontos eredményt (azt csak az összes legenerált mátrix ismeretében tudnánk megmondani).

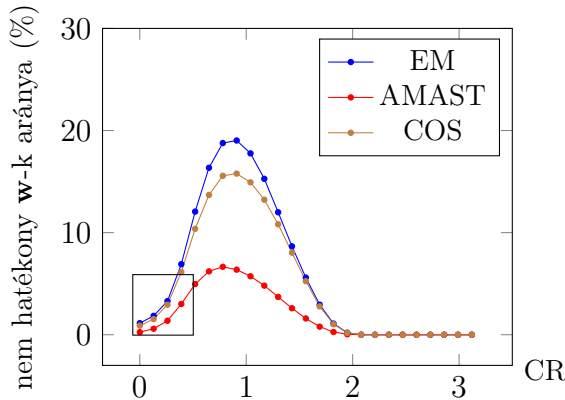
Az ábrák bal oldalán a $[0, CR_{\max}^n]$ intervallumon, a jobb oldalán pedig a $[0, 0.2]$ intervallumon felnagyítva található az adott módszer által adott nem hatékony súlyvektorok aránya.



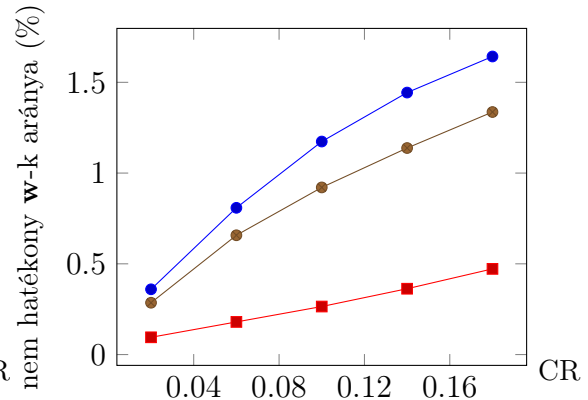
(a) A 4×4 -esek max $CR_4 = 4.02212$ -ig



(b) A 4×4 -es mátrixok a $[0, 0.2]$ -n



(c) Az 5×5 -ösök max $CR_5 = 3.20681$ -ig

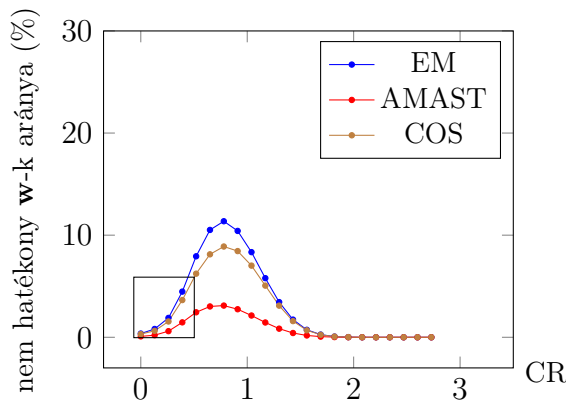


(d) Az 5×5 -ös mátrixok a $[0, 0.2]$ -n

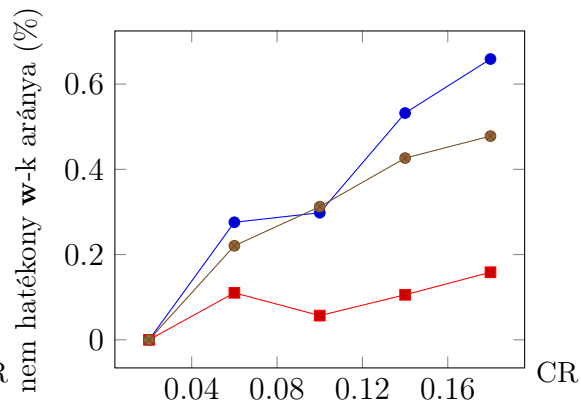
5 és annál nagyobb dimenzióban átlagosan az EM teljesít a legrosszabbul, utána a COS , majd az $AMAST$ módszer következik.

Az ábrákon látható, hogy a módszerek által adott nem hatékony súlyvektorok relatív gyakorisága egy jól látható maximumot (csúcs) hoz létre, és ez a maximum nem a legnagyobb CR környékén található, hanem 1 körül, egyre inkább a 0 felé tolódva. A konkrét értékeket a 4.2 táblázatban foglaltuk össze. A hisztogram intervallumok legkisebb felosztása 0.01 volt, de az ábrákat nagyobb, 0.12-es felosztással ábráztuk, emiatt a tényleges maximum hely és érték eltérhet az ábráról leolvashatótól.

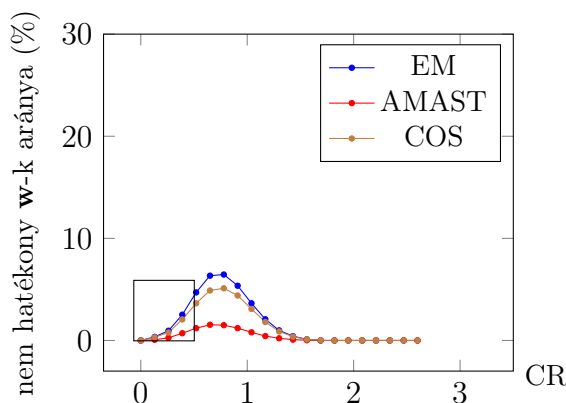
Sejtésként megfogalmazhatjuk, hogy a maximális relatív gyakoriság a dimenzió növekedésével tart a nullához, valamint a maximum helye is monoton csökken (nem látszik, hogy



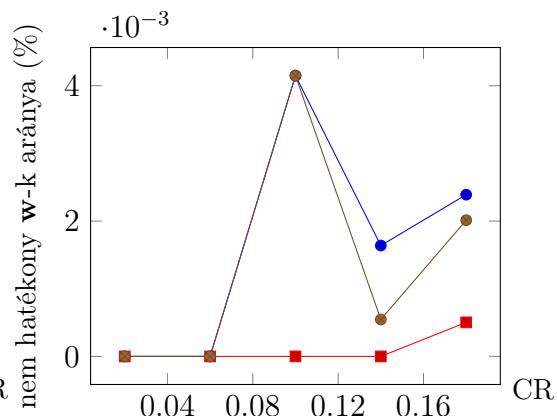
(e) A 6×6 -osok max $CR_6 = 2.84672$ -ig



(f) A 6×6 -os mátrixok a $[0, 0.2]$ -n



(g) A 7×7 -esek max $CR_7 = 2.65175$ -ig



(h) A 7×7 -os mátrixok a $[0, 0.2]$ -n

4.4. ábra. A módszerek által adott nem hatékony súlyvektorok aránya a CR függvényében

Dimenzió	CR_{\max}	Rel. gyak. max hely	Rel. gyak. max érték
4×4	4.02212	1.2	0.3852
5×5	3.20681	0.91	0.1964
6×6	2.84672	0.84	0.1146
7×7	2.65175	0.77	0.0679

4.2. táblázat. A nem hatékony súlyvektorok relatív gyakoriságának maximum helye és értéke dimenzióként, valamint a CR maximális értéke

hova tart). Ennek elméleti síkon történő megtámadása nehéz feladat, mivel akár csak az átlagos λ_{\max} , ezek az eredmények mind numerikus módon kaphatóak meg.

A felnagyított (jobb oldali) ábrákon a $[0, 0.2]$ intervallumon láthatjuk ugyanezeket az értékeket. Ezek nincsenek azonos skálán, mert akkor a legutolsó, 7 dimenziós ábrán már nem lehet látni semmit, olyan kis értékeket venne fel. Noha a populáció nagysága növekszik (véletlenszerűen generált mátrixok száma, amelyekből a hisztogramot építettük), de nem

exponenciálisan, ahogyan az összes generálható mátrixok száma nő. A torzulás 7 dimenzióban már túl nagy, hogy szép ábrát kapjunk, viszont az látható, hogy ahogy a dimenzió 4-ről 7-re nő az y tengely nagyságrendje 1%-ról $10^{-3}\%$ -ra csökken, ami jelentős. Tehát azt kaptuk, hogy a $CR < 0.1$ elfogadási küszöböt teljesítő mátrixokra nagyobb mértékben kapunk hatékony súlyvektorokat, mint hogyha nagyobb CR -ű mátrixokat vizsgálunk.

Sok szempontot/alternatívát tartalmazó mátrixoknak nagy a dimenziója is, ami ha teljesen kitöltött mátrixokkal szeretnénk dolgozni, nagyon sok kérdést jelent a döntéshozó számára ($\frac{n(n-1)}{2}$). Emellett az AHP-ban gyakori, hogy részre bontjuk a feladatot, és így tipikusan kisebb dimenziós mátrixok jelennek meg [50]. Elméletileg egy szép eredmény, hogy nagyobb dimenzió esetén a gyakorlatban használt módszerek aszimptotikusan egyre kisebb eséllyel adnak nem hatékony súlyvektort eredményül, de mivel a gyakorlatban a dimenziót már csak praktikusági okokból is szeretik alacsony tartani, ez nem jelent garanciát arra vonatkozóan, hogy a módszerek elég nagy eséllyel nem „hibáznak”.

Felső korlátok

Megnéztük továbbá azt is, hogy a korábbi, 2.4. alfejezetben vázolt korlátok mennyiben teljesülnek. Ahogyan az várható volt, a páratlan dimenziókban valóban visszakaptuk az elméletben is igazolt felső korlátokat, a páros dimenziókban eredményeit pedig most ismergetjük.

4 dimenziós esetben a teljes mátrixok listájának birtokában meg tudtuk nézni, hogy melyik mátrix maximalizálja a CR -t. Nem meglepő módon a következő $\mathbf{A} \in \mathcal{PCM}_4$ mátrixot kaptuk:

$$\mathbf{A} = \begin{pmatrix} 1 & 9 & \frac{1}{9} & 9 \\ \frac{1}{9} & 1 & 9 & \frac{1}{9} \\ 9 & \frac{1}{9} & 1 & 9 \\ \frac{1}{9} & 9 & \frac{1}{9} & 1 \end{pmatrix}$$

(Egészen pontosan ennek lexikografikusan minimális permutáltját.)

Kiszámítva ezen mátrix λ_{\max} -át 13.7518-at kaptunk, és így a $CR = 3.677$. Összevetve a 2.1 táblázatbeli 4.021 max CR felső korláttól, ez kissé elmarad tőle, tehát páros dimenziókban biztosan nem éles a korlát. Egyelőre nyitott kérdés, hogy adható-e páros dimenziókra éles korlát.

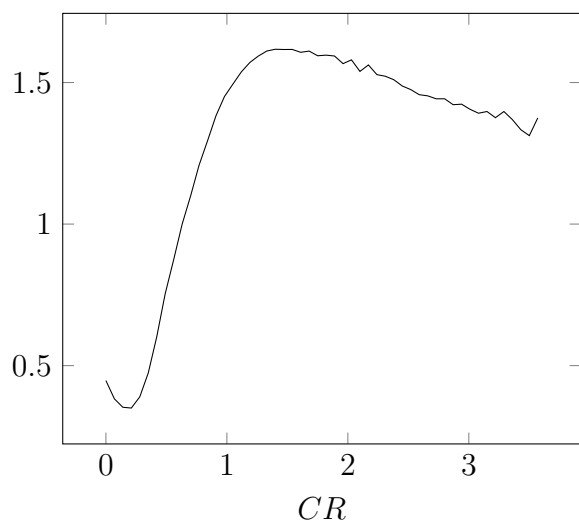
Hat dimenzióban ugyanúgy feltettük ezeket a kérdéseket, viszont ott nem tudtuk leellenőrizni, hogy valóban ez a struktúra maximalizálja-e a CR -t mivel nincs meg az összes generált permutációfüggetlen mátrix. A futtatott programok során nem találtunk az \mathbf{A} -val analóg 6 dimenziós mátrixnál nagyobb CR -t produkáló mátrixot, de mivel az eszközeink a véletlenül alapulnak, ez nem ad semmiféle garanciát. Hat dimenzióban a hasonló struktúra $CR = 2.75413$ -at produkál szemben a táblázat 2.847-es felső becslésével. Ez a két érték már közelebb van egymáshoz, de nem egyeznek meg.

4.3. Főátló feletti egynél nagyobb elemek

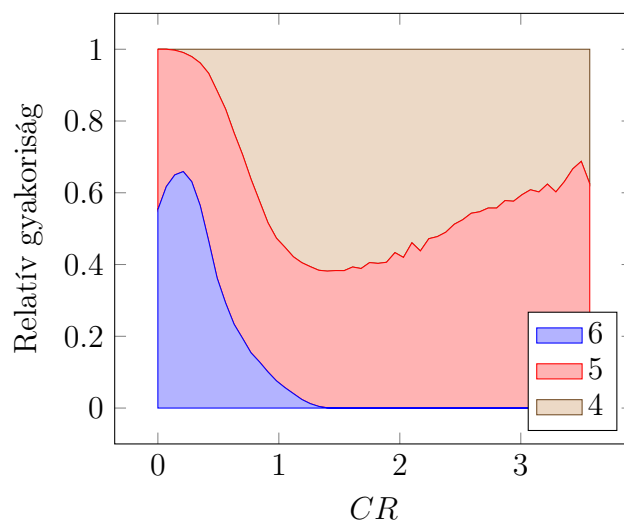
A hatékonyság mellett ugyan marginális kérdés, de a teljes 4×4 -es mátrixok listájának birtokában olyan kérdések is felmerülnek, hogy hány főátló feletti egynél nagyobb elemmel tudjuk reprezentálni a mátrixot.

4 dimenzióban a mátrix vagy transzponáltja tartalmaz legalább három ≥ 1 elemet. Egy adott mátrix összes permutáltján végigiterálva azt kapjuk, hogy valójában mindig tudunk kapni legalább 4 elemet (4.5. ábra). Az eloszlást megfigyelve azt láthatjuk, hogy a „szép” mátrixok (tehát amelyeknek sorai és oszlopai átpermutálhatók úgy, hogy csak ≥ 1 egészek kerüljenek a felső háromszög részébe) az alacsony CR -ű elemek, és egy adott CR után már nem is fordulnak elő (1.2 felett nincs ilyen mátrix).

A bal oldali ábra is érdekes eredményt mutat: az átlag maximumát szintén 1.2 körül éri el meredeken növekedve 0.1-től, ahol lokális minimuma van.



(a) A főátló feletti ≥ 1 elemek átlagos száma



(b) A főátló feletti ≥ 1 elemek eloszlása

4.5. ábra. A 4×4 -es mátrixok főátló feletti ≥ 1 elemek száma

5. fejezet

Programkönyvtár a mátrixok vizsgálatára

A 4. fejezetben ismertettük az eredményeket, és ebben a fejezetben összefoglaljuk az eszközöket, amelyeket használtunk, vagy éppen a kutatással párhuzamosan fejlesztettünk.

A mátrixok egyenkénti vizsgálatához több különböző módszerünk is van. Vannak Matlab és Scilab kódok, melyek pl. a (3.2) LP feladatot oldják meg, és adják ki az eredményt. Az MTA SZTAKI által működtetett programmal is elemezhetünk mátrixokat, ami online megtalálható¹. Mindemellett leginkább a különböző kódok egységesítése és performancia kérdések miatt született egy C++ library a páros összehasonlítás-mátrixok elemzésére.

A kódbázis GitHub-on elérhető², és bárki által letölthető és használható. A program több numerikus példát is tartalmaz, amely segíthet a fogalmak könnyebb megértésében.

A program jelenleg a következő funkcionalitásokkal bír:

1. Mátrixok generálása, kiírása és olvasása
2. Mátrixok szűrése: permutációsűrés, $CR < 0.1$ küszöb
3. A módszerek futtatása mátrixokon (CR meghatározása, sajátvektor, *AMAST*, koszinuszos módszer, LP, erősen összefüggővé tevő gráfos algoritmus)
4. Statisztikák, pl. hisztogramok készítése: vödrök feltöltése
5. Vizuális megjelenítő és elemző

5.1. Mátrixok generálása, kiírása és olvasása

A kód tetszőleges dimenziójú páros összehasonlítás mátrixot képes kezelni. Az írás/olvasás saját formátumba (.mt) vagy CSV-be történik. Tipikusan csak a mátrixok szigorú felső háromszög részét érdemes eltárolni, valamint mivel a szokásos halmazból vesszük az elemeket

¹<http://pcmc.online>

²<https://github.com/cetLEE/AHP-matrix-project>

($\{1, 2, \dots, 9, \frac{1}{2}, \frac{1}{3}, \dots, \frac{1}{9}\}$), elég csak az elem sorszámát eltárolni, ezzel csökkentve a méretet (tehát pl. nem az $\frac{1}{9}$ -et tároljuk el lebegőpontosan, hanem 16-ot).

Az összes generált 4×4 mátrix 1.15 GB helyet foglal, a permutációsűrés utáni kicsit több mint egy millió mátrix már egy kezelhetőbb 49.2 MB nagyságú fájlban tárolható.

Ahogy láthattuk a 4.1. alfejezetben, a kombinatorikus robbanás miatt a jelenlegi hardverekkel nincs reális esély az 5 és annál nagyobb dimenziós mátrixok teljes listáját megkapni. Ezen esetekre létrehoztunk egy alternatív megoldást, amely véletlenszerűen generál mátrixokat (a főtálo fölötti elemeket az $\{1, 2, \dots, 8, 9, \frac{1}{2}, \frac{1}{3}, \dots, \frac{1}{9}\}$ halmazból azonos ($\frac{1}{17}$) valószínűséggel véletlenszerűen töltve ki).

5.2. Mátrixok szűrése

5.2.1. Permutációsűrés

Amint írtuk a korábbi fejezetben is, a permutáció-invariáns kérdés a hatékonyság, gyenge hatékonyság és kapcsolódó fogalmak. Emiatt is fontos volt kiszűrni a „duplikált” elemeket, így a 4×4 -es esetben ezt meg is tettük. 4 dimenzióra ez kb. 2 percet vett igénybe.

Próbálkoztunk magasabb dimenzióban akár a lexikografikus rendezés szerinti minimális elemek megtalálásával (nem pedig egy sok memóriát használó tömbben jelölve, hogy mely elemeket néztük már meg), de az összes permutáción való végigiterálás mindenképpen egy újabb n -es szorzóval növelte a futásidőt, túllépve a számítási kapacitását a mai hardvereknek.

5.2.2. CR szerinti szűrés

A program adott dimenzióra megkeresi a $\overline{\lambda_{\max}}$ értékét, de általában előre lementett számokkal dolgozik. Szubrutinként használva egy másik programcsomagot (Eigen) sajátértéket és sajátvektort is számol, így a CR és egyéb inkonzisztencia-indexek is megkaphatók. Külön szűrő lett beépítve, ha éppen a $CR < 0.1$ küszöböt elérő mátrixokra vagyunk kíváncsiak. Ahogyan az Bozóki és Rapcsák korábbi eredményeiből kiderül, a dimenzió növekedésével a véletlenszerűen generált mátrixoknak egyre kisebb része marad a küszöb alatt, így nagyobb dimenzió esetén egyre ritkábban talál megfelelő mátrixokat a program [18].

A már permutációsűrt, egymillió elemet tartalmazó halmazt továbbszűrve kaptuk meg a 32 ezer elemet, amelyekre legalább egy módszer nem hatékony súlyvektort adott - ez mindösszesen 4 percig tartott (a 3 módszer alkalmazása a mátrixokra és az eredmények aggregálása, majd az eredmény kiírása).

5.3. Módszerek futtatása

A különböző módszereket implementáltunk a mátrixok vizsgálatára:

1. CR meghatározása

2. sajátvektor meghatározása
3. *AMAST* súlyvektor meghatározása alacsony dimenzióra
4. koszinuszos módszer
5. a (3.2) LP feladat LP solverrel történő megoldása
6. erősen összefüggővé tévő gráfos algoritmus

A módszerek sokszor nemtriviális szubrutinokat használnak. A legnagyobb sajátérték és ahhoz tartozó sajátvektor meghatározásához az Eigen iteratív alapú módszert használt.

Az *AMAST* n^{n-2} fesztőfán való végigiterálás számításigényes feladat, 7 dimenzióig a konkrét képletet beleírtuk a programba, mely megadja az a_{ij} -k függvényében a w_k -kat. Ezek a képletek egyre bonyolultabbak, viszont gyorsabb ezt használni, mint minden alkalommal végigiterálni az n^{n-2} fesztőfán.

A gráfos algoritmushoz használható a Lemon programcsomag, amely jól kezel gráfokat (ugyan maga az algoritmus nem ebben lett implementálva, de az elemzéshez hasznos). A Lemon tartalmaz interfészt LP és IP megoldókhoz, így a Lemon-on keresztül hívtam meg a Glpk (General Linear Programming Kit) LP solver motorját.

5.4. Statisztikák, vizualizáció, egyebek

5.4.1. Statisztikák készítése

4×4 -es esetben egy teljes összefoglaló csv lett kimentve, mely a 6.2.1. példán látható. Nagyobb dimenzióban mivel nem volt teljes kollekció, ezért nem íratunk ki ilyen táblázatot, hanem leginkább hisztogramokban dolgoztunk. Ezek eredményei találhatóak például a 4.4 ábrán. A különböző felosztású hisztogramokról szóló példákat a könyvtárba is feltöltöttünk.

5.4.2. Vizuálizáció

Az eredményeket több módon vizualizáltuk. Egyrészt generáltunk \LaTeX kódokat, mint pl. ez a dolgozat, vagy [4] cikk melléklete (pgfplots). A hisztogramok kiírása előtt webes megjelenítőket is lehet használni az adatok azonnali vizualizációjára – ehhez a highcharts.js-t használtam.

5.4.3. Egyéb hasznos függvények

A kutatás során sok zsákutcába botlottunk, ilyen volt például az 5×5 -ös mátrixok generálásának megkísérlése. Ekkor több olyan új módszert dolgoztunk ki, amelyek bár nem vezettek eredményre, jó alapot szolgálhatnak a további kutatásokhoz/új ötletekhez.

A 4×4 -es permutációsúrt mátrixokat kiegészíthetjük 5×5 -ösökké, ezáltal elven kevesebb mátrix között kell kiszűrni a permutációkat. Bár az ötlet jó, futásidőben még mindig megközelíthetetlen (1-2 év), a kiegészítést végző függvényeket viszont a könyvtár tartalmazza.

Irodalomjegyzék

- [1] Ábele-Nagy, K. (2018): Páros összehasonlítás mátrixok a többszemponútú döntésméletben, Ph.D. értekezés, Budapesti Corvinus Egyetem.
- [2] Adamcsek, E. (2008): The Analytic Hierarchy Process and its Generalizations, Diplomamunka, Eötvös Loránd Tudományegyetem.
- [3] Aguarón, J., Moreno-Jiménez, J., M. (2003): The geometric consistency index: Approximated thresholds. *European Journal of Operational Research* 147(1):137-145.
- [4] Antal, A., Bozoki, S., Fulop, J. (2018) On the frequency of inefficient priority vectors derived from 4x4 pairwise comparison matrices (manuscript).
- [5] Aupetit, B., Genest, C. (1993) On some useful properties of the Perron eigenvalue of a positive reciprocal matrix in the context of the analytic hierarchy process. *European Journal of Operational Research* 70(2):263–268
- [6] Bajwa, G., Choo, E.U., Wedley, W.C. (2008) Effectiveness analysis of deriving priority vectors from reciprocal pairwise comparison matrices. *Asia-Pacific Journal of Operational Research*, 25(3):279–299
- [7] Blanquero, R., Carrizosa, E., Conde, E. (2006) Inferring efficient weights from pairwise comparison matrices. *Mathematical Methods of Operations Research* 64(2):271–284
- [8] Brunelli, M., Canal, L., and Fedrizzi, M. (2013): Inconsistency indices for pairwise comparison matrices: a numerical study. *Annals of Operations Research*, 211(1):493-509,.
- [9] Brunelli, M., Fedrizzi, M. (2015): Axiomatic properties of inconsistency indices for pairwise comparisons. *Journal of the Operational Research Society*, 66(1):1-15.
- [10] Bozóki, S. (2006): Súlyok meghatározása páros összehasonlítás mátrixok legkisebb négyzetes közelítése alapján. *Alkalmazott Matematikai Lapok* 23, 121-137.
- [11] Bozóki, S. (2006): Súlyozás páros összehasonlítással és értékelés hasznossági függvényekkel a többszemponútú döntési feladatokban, Ph.D. értekezés, Budapesti Corvinus Egyetem.
- [12] Bozóki, S. (2008): Solution of the least squares method problem of pairwise comparison matrices. *Central European Journal of Operations Research*, 16 (4), 345–358.
- [13] Bozóki, S., (2014): Inefficient weights from pairwise comparison matrices with arbitrarily small inconsistency. *Optimization*, 63(12):1893-1901.

- [14] Bozóki, S., Csató, L., Temesi, J. (2016): An application of incomplete pairwise comparison matrices for ranking top tennis players, *European Journal of Operational Research*, 248(1) 211–218
- [15] Bozóki, S., Fülöp, J. (2017): Páros összehasonlítás mátrixokból számolt súlyvektorok hatékonysága, *Sigma*, XLVIII(1-2) 3-17.o.
- [16] Bozóki, S., Fülöp, J. (2018) Efficient weight vectors from pairwise comparison matrices. *European Journal of Operational Research* 264(2):419-427
- [17] Bozóki, S., Fülöp, J., Rónyai, L. (2010): On optimal completions of incomplete pairwise comparison matrices. *Mathematical and Computer Modelling* 52(2010), 318-333.
- [18] Bozóki, S., Rapcsák, T. (2008) On Saaty's and Koczkodaj's inconsistencies of pairwise comparison matrices. *Journal of Global Optimization* 42(2):157–175.
- [19] Bozóki, S., Tsyganok, V. (≥ 2018) The logarithmic least squares optimality of the geometric mean of weight vectors calculated from all spanning trees for (in)complete pairwise comparison matrices. Under review, <https://arxiv.org/abs/1701.04265>
- [20] Choo, E.U., Wedley, W.C. (2004) A common framework for deriving preference values from pairwise comparison matrices. *Computers & Operations Research* 31(6):893–908
- [21] Chu, A. T. W., Kalaba, R. E. , Spingarn, K. (1979): A comparison of two methods for determining the weight belonging to fuzzy sets. *Journal of Optimization Theory and Applications*, 27 (4), 531–538 .
- [22] Conde, E., Pérez, M.d.l.P.R. (2010) A linear optimization problem to derive relative weights using an interval judgement matrix. *European Journal of Operational Research* 201(2):537–54
- [23] Crawford, G., Williams, C. (1980) Analysis of subjective judgment matrices. The Rand Corporation, Office of the Secretary of Defense USA, R-2572-AF
- [24] Crawford, G., Williams, C. (1985) A note on the analysis of subjective judgment matrices. *Journal of Mathematical Psychology* 29(4):387–405
- [25] Csató, L. (2014): A páros összehasonlításokon alapuló rangsorolás módszertani és alkalmazási kérdései, Ph.D. értekezés, Budapesti Corvinus Egyetem.
- [26] Csató, L. (2019): A characterization of the Logarithmic Least Squares Method, *European Journal of Operational Research*, megjelenés alatt.
- [27] de Graan, J.G. (1980) Extensions of the multiple criteria analysis method of T.L. Saaty. Presented at EURO IV Conference, Cambridge, July 22-25.
- [28] Duszak, Z., Koczkodaj, W., W. (1994): Generalization of a new definition of consistency for pairwise comparisons. *Information Processing Letters* 52(5):273-276.
- [29] Fedrizzi, M. (2013) Obtaining non-dominated weights from preference relations through norm-induced distances. Proceedings of the XXXVII Meeting of the Italian Association for Mathematics Applied to Economic and Social Sciences (AMASES), Stresa, Italy, September 5-7, 2013.
- [30] Fedrizzi, M., Brunelli, M. (2010) On the priority vector associated with a reciprocal relation and a pairwise comparison matrix. *Soft Computing* 14(6):639–645

- [31] Fichtner, J. (1984) Some thoughts about the Mathematics of the Analytic Hierarchy Process. Report 8403, Universität der Bundeswehr München, Fakultät für Informatik, Institut für Angewandte Systemforschung und Operations Research, Werner-Heisenberg-Weg 39, D-8014 Neubiberg, F.R.G. 1984.
- [32] Fichtner, J. (1986) On deriving priority vectors from matrices of pairwise comparisons. *Socio-Economic Planning Sciences* 20(6):341–345
- [33] Golany, B., Kress, M. (1993) A multicriteria evaluation of methods for obtaining weights from ratio-scale matrices. *European Journal of Operational Research* 69(2):210–220
- [34] Jensen, R., E., (1984): An alternative scaling method for priorities in hierarchical structures. *Journal of Mathematical Psychology*. 28 (3):317-332.
- [35] Kéri, G. (2005): Kritériumok páros összeasonlítás mátrixokra, *Sigma*, XXXVI(3-4):139–148
- [36] Koczkodaj, W. W. (1993): A new definition of consistency of pairwise comparisons. *Mathematical and Computer Modelling*, 18(7):79-84.
- [37] Komáromi, É. (2013): A Kullback-Leibler relatív entrópia függvény alkalmazása páros összehasonlítás mátrix egy prioritásvektora meghatározására, *Sigma*, XLIV(1-2):1–19
- [38] Kou, G., Lin, C. (2014) A cosine maximization method for the priority vector derivation in AHP. *European Journal of Operational Research* 235(1):225–232
- [39] Lin, C.-C. (2007) A revised framework for deriving preference values from pairwise comparison matrices. *European Journal of Operational Research* 176(2):1145–1150
- [40] Lundy, M., Siraj, S., Greco, S. (2017) The mathematical equivalence of the „spanning tree” and row geometric mean preference vectors and its implications for preference analysis. *European Journal of Operational Research* 257(1):197–208
- [41] Poesz, A. (2017): Inkonzisztencia a döntéshozatalban, Ph.D. értekezés, Budapesti Corvinus Egyetem.
- [42] Rabinowitz, G. (1976) Some comments on measuring world influence. *Journal of Peace Science* 2(1):49–55
- [43] Rapcsák, T. (2007) Többszemponútú döntési problémák, egyetemi jegyzet, Budapesti Corvinus Egyetem MTA SZTAKI-ba kihelyezett Gazdasági Döntések Tanszék. http://www.oplab.sztaki.hu/tanszek/download/I_Tobbsz_dont_modsz.pdf
- [44] Saaty, T.L. (1977) A scaling method for priorities in hierarchical structures. *Journal of Mathematical Psychology* 15(3):234–281
- [45] Simon, H.A. (1957): *Models of Man*, Wiley, New York.
- [46] Simon, H.A. (1979): *Models of Thought*, Yale University Press, New Haven.
- [47] Siraj, S., Mikhailov, L., Keane, J.A. (2012) Enumerating all spanning trees for pairwise comparisons. *Computers & Operations Research* 39(2):191–199
- [48] Siraj, S., Mikhailov, L., Keane, J.A. (2012) Corrigendum to „Enumerating all spanning trees for pairwise comparisons [Comput. Oper. Res. 39 (2012) 191-199]”. *Computers & Operations Research* 39(9) page 2265

- [49] Taji, K., Mizuno, T.: On properties of Pareto optimal weights from pairwise comparison matrices based on graph, International Symposium on the Analytic Hierarchy Process (2018).
- [50] Temesi, J. (2002): A döntéselmélet alapjai, Aula Kiadó
- [51] Tsyganok, V. (2000) Combinatorial method of pairwise comparisons with feedback. Data Recording, Storage & Processing 2:92–102 (in Ukrainian)
- [52] Tsyganok, V. (2010) Investigation of the aggregation effectiveness of expert estimates obtained by the pairwise comparison method. Mathematical and Computer Modelling 52(3-4):538–544

6. fejezet

Melléklet

6.1. Hatékonyság

6.1.1. Példa. Tekintsük a következő két páros összehasonlítás mátrixot!

$$\mathbf{A} = \begin{pmatrix} 1 & 2 & 4 \\ 1/2 & 1 & 2 \\ 1/4 & 1/2 & 1 \end{pmatrix} \quad \mathbf{A}' = \begin{pmatrix} 1 & 2 & 3 \\ 1/2 & 1 & 2 \\ 1/3 & 1/2 & 1 \end{pmatrix}$$

A baloldali \mathbf{A} mátrix konzisztens, mert az egyetlen i, j, k hármásra, ahol ezek különbözőek teljesül a kardinális tranzitivitás: $a_{12}a_{23} = a_{13}$, azaz $2 \cdot 2 = 4$.

A jobboldali, \mathbf{A}' mátrix inkonzisztens, mivel ugyanezen egyetlen hármásra nem teljesül ez a feltétel, mivel $a_{13} = 3$.

6.1.2. Példa. Tekintsük a következő \mathbf{A} mátrixot. A mátrix mellé írtuk a legnagyobb sajátértéket és az ebből számolt CR -t.

$$\mathbf{A} = \begin{pmatrix} 1 & 2 & 9 & 8 \\ 1/2 & 1 & 3 & 6 \\ 1/9 & 1/3 & 1 & 3 \\ 1/8 & 1/6 & 1/3 & 1 \end{pmatrix}, \quad \lambda_{\max} = 4.1263, \quad CR = 0.0476$$

A sajátvektor módszer által kapott súlyvektor a következő (\mathbf{w}^{EM})-mel jelölve:

$$\mathbf{w}^{EM} = \begin{pmatrix} 0.578100 \\ 0.277375 \\ 0.096350 \\ 0.048175 \end{pmatrix}$$

A mátrix elemeit tehát a súlyvektor koordinátáinak hányadosaival közelítettük. A konkrét értékei a hányadosoknak a következők lesznek.

$$\left[\frac{w_i^{EM}}{w_j^{EM}} \right] = \begin{pmatrix} 1 & 2.0842 & 6 & 12 \\ 0.4798 & 1 & 2.8788 & 5.7577 \\ 1/6 & 0.3474 & 1 & 2 \\ 1/12 & 0.1737 & 1/2 & 1 \end{pmatrix},$$

A súlyvektor nem hatékony, ugyanis ha a (pirossal kiemelt) 2. koordinátáját megnöveljük, akkor a következő, \mathbf{w}' súlyvektorral (ezt már normálás után kaptuk) jobban közelítjük a mátrix azon elemeit, amelyeket a hányadosokkal felírt mátrixban szintén pirossal jelöltünk.

Zölddel jelölve \mathbf{w}' javított koordinátáját:

$$\mathbf{w}' = \begin{pmatrix} 0.571429 \\ 0.285714 \\ 0.095238 \\ 0.047619 \end{pmatrix} = 0.988460 \cdot \begin{pmatrix} 0.578100 \\ 0.289050 \\ 0.096350 \\ 0.048175 \end{pmatrix},$$

A \mathbf{w}' súlyvektorhoz tartozó hányadosokból a következő mátrixot írhatjuk fel, ami valóban jobb közelítés – kék színnel kiemelve ahol tökéletesen közelítettük az eredeti mátrix megfelelő elemeit.

$$\left[\frac{w'_i}{w'_j} \right] = \begin{pmatrix} 1 & 2 & 6 & 12 \\ 1/2 & 1 & 3 & 6 \\ 1/6 & 1/3 & 1 & 2 \\ 1/12 & 1/6 & 1/2 & 1 \end{pmatrix},$$

6.2. Mátrixtesztelés

6.2.1. Példa.

#	Elements	λ_{\max}	CR	EM?	AMAST?	COS?	...
1	(2, 2, 2, 2, 2)	4	0	Y	Y	Y	...
2	(2, 2, 3, 2, 3, 2)	4.0606	0.02287	Y	Y	Y	...
3	(2, 2, 4, 2, 3, 2)	4.118	0.0445	Y	Y	Y	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮	...

...	\mathbf{w}_{EM}	\mathbf{w}_{ST}	...
...	(0.25, 0.25, 0.25, 0.25)	(0.25, 0.25, 0.25, 0.25)	...
...	(0.2894, 0.2894, 0.2463, 0.1751)	(0.2897, 0.2897, 0.246, 0.1746)	...
...	(0.3191, 0.281, 0.2428, 0.1571)	(0.3158, 0.2838, 0.2432, 0.1572)	...
	⋮	⋮	

...	\mathbf{w}_{COS}	\mathbf{w}'_{EM}	\mathbf{w}'_{ST}	\mathbf{w}'_{COS}
...	(0.25, 0.25, 0.25, 0.25)	-	-	-
...	(0.2881, 0.2881, 0.2475, 0.1762)	-	-	-
...	(0.3137, 0.28, 0.246, 0.1603)	-	-	-
	⋮	⋮	⋮	⋮

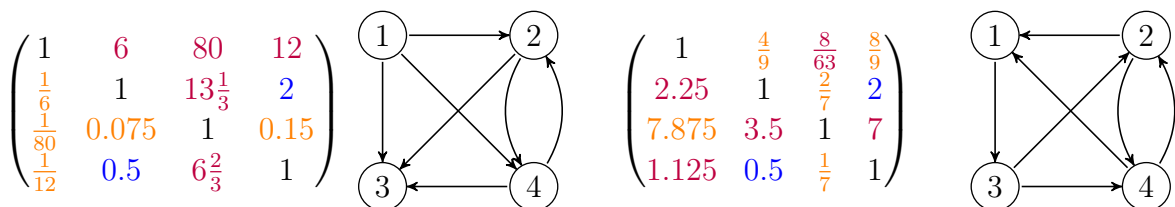
ahol az „Elements” oszlopban a mátrix szigorú felső háromszög részének elemei vannak, valamint \mathbf{w}'_{EM} jelöli a (3.2) lineáris programozási feladat megoldását, ha az abban a sorban lévő \mathbf{w}_{EM} nem volt hatékony.

6.3. Erősen összefüggővé tevő algoritmus

6.3.1. Példa. Tekintsük a következő mátrixot és súlyvektorokat:

$$\mathbf{A} = \begin{pmatrix} 1 & 2 & \frac{1}{9} & 1 \\ \frac{1}{2} & 1 & 3 & 2 \\ 9 & \frac{1}{3} & 1 & 6 \\ 1 & \frac{1}{2} & \frac{1}{6} & 1 \end{pmatrix} \quad \hat{\mathbf{w}} = \left(120, 20, \frac{3}{2}, 10 \right) \quad \tilde{\mathbf{w}} = \left(\frac{8}{9}, 2, 7, 1 \right)$$

Az alábbi 6.1 ábrán a $\hat{\mathbf{w}}$ és a $\tilde{\mathbf{w}}$ súlyvektorok közelítései láthatóak a 3.2.1. definíció alapján a belőlük konstruált irányított gráfokkal. A mátrixelemeket kiszíneztük narancssárgával, ha szigorúan alul-, lilával, ha szigorúan felülbecsültük, valamint késsel, ha tökéletesen sikerült közelítenünk. A lila elemek reprezentálják az oda-élt, a narancssárga a vissza-élt, a kékek pedig az oda-vissza élt.



(a) $\hat{\mathbf{w}}$ súlyvektor általi közelítése a mátrixelemeknek, és a hozzá tartozó irányított gráf

(b) $\tilde{\mathbf{w}}$ súlyvektor általi közelítése a mátrixelemeknek, és a hozzá tartozó irányított gráf

6.1. ábra. Az \mathbf{A} mátrix elemeinek közelítése a súlyvektorok által

Ahogy az látható, $\hat{\mathbf{w}}$ nem hatékony, ugyanis a 6.1a gráf nem erősen összefüggő (a 3-as csúcs nyelő, azaz belőle nem indul ki irányított él), míg a $\tilde{\mathbf{w}}$ hatékony, ugyanis a 6.1b gráf erősen összefüggő (az $1 \rightarrow 3 \rightarrow 4 \rightarrow 2 \rightarrow 1$ csúcsok irányított Hamilton kört alkotnak).

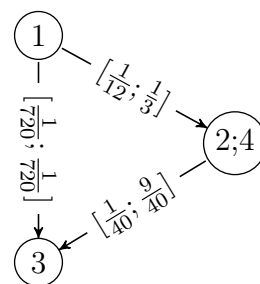
Nézzük az algoritmus futását a $\hat{\mathbf{w}}$ súlyvektor inputra. A gráf erősen összefüggő komponenseit a 6.2 ábra szemlélteti, továbbá minden $u \rightarrow v$ élre ráírtuk az $[l[u, v]; u[u, v]]$ értékeket. Az $1 \rightarrow \{2; 4\}$ élhez tartozó alsó korlátot az eredeti gráf $1 \rightarrow 4$ éle adta – ott az érték a következő:

$$\frac{w_4 \cdot a_{14}}{w_1} = \frac{10 \cdot 1}{120} = \frac{1}{12}$$

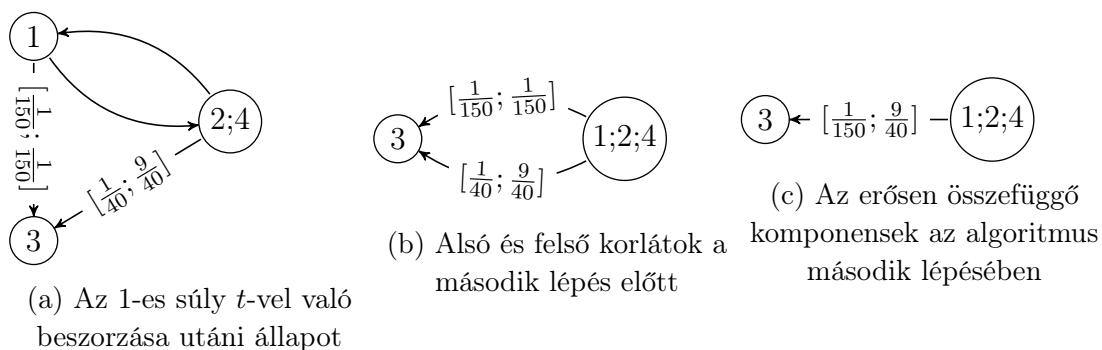
Hasonlóképpen a felső korlát az $1 \rightarrow 2$ élből származik:

$$\frac{w_2 \cdot a_{12}}{w_1} = \frac{20 \cdot 2}{120} = \frac{40}{120} = \frac{1}{3}$$

A többi élre ez ugyanígy elvégezhető. Válasszunk két tetszőleges komponenset, pl. az 1 és a $\{2; 4\}$ csúcsokat tartalmazókat és olvassuk össze őket. Válasszuk a stratégiák közül az alsó és a felső korlát átlagával módosítót, így tehát a t értéke $(\frac{1}{12} + \frac{1}{3})/2 = \frac{5}{24}$. Egyszerűbb



6.2. ábra. Az erősen összefüggő komponensek az algoritmus első lépésében



6.3. ábra. Az algoritmus lépései

az 1-es súlyt, és a hozzá tartozó élek alsó és felső korlátait megváltoztatni – így kapjuk a $\hat{\mathbf{w}}' = (25, 20, 1.5, 10)$ súlyvektort, és a 6.3a ábrán látható gráfot.

Az új erősen összefüggő komponensek a 3-as és az $\{1; 2; 4\}$ -es csúcsokból álló komponensek lesznek, de ez előtt a lépés előtt, az alsó és felső korlátokat is módosítani kell. Ebben az esetben csak két komponens olvadt össze, tehát az új komponens alsó és felső korlátja csak azon két él minimum/maximuma közül kerül ki, ahogy az látható a 6.3b és a 6.3c ábrákon.

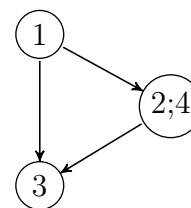
A második lépésben a $t = (\frac{1}{150} + \frac{9}{40})/2 = \frac{139}{1200}$ értékkel módosítjuk a 3-as koordinátáját a $\hat{\mathbf{w}}'$ -nek. Mivel az él oda mutat, ezért ennek reciprokával kell beszoroznunk. Így kapjuk $\hat{\mathbf{w}}' = (25, 20, 12.9496, 10)$, és miután újra lefuttatjuk a komponensek keresését, azt kapjuk, hogy a gráf erősen összefüggő. Megállhatunk és visszaadhatjuk $\hat{\mathbf{w}}'$ -t.

Újra tekintve a 3.2.4. állítást, az előző két súlyvektorra alapozva bizonyíthatjuk:

6.3.2. Állítás. *Egy \mathbf{w} súlyvektor bemenet esetén a t választásaitól és a komponensek összeolvasztásának sorrendjétől függetlenül az algoritmus nem feltétlenül képes megtalálni akár-melyik öt domináló hatékony súlyvektort.*

Bizonyítás:

Ha a $\hat{\mathbf{w}}$ nem hatékony súlyvektorra futtatjuk az algoritmust, akkor ahogy azt már láttuk, a 6.4 ábrán látható gráfot kapjuk. Ezután több lehetőségünk van: összeolvaszthatjuk az 1. és 3. komponenst, ekkor viszont ekkor az 1 és 3 között menő oda és vissza él már nem szűnhet az algoritmus további futása alatt, és így biztosan nem kapjuk meg $\tilde{\mathbf{w}}$ -t (6.1b ábra gráfját), ami nem tartalmazza ezt az oda-vissza élet.



6.4. ábra

Ekkor kénytelenek vagyunk a $\{2; 4\}$ komponenst összeolvasztani vagy az 1-es vagy pedig a 3-as csúccsal. Bármelyik csúccsal is tesszük ezt meg, utána már a közöttük lévő élek véglegesek lesznek. Ekkor viszont nem kaphatjuk meg a 6.1b ábra gráfját, ugyanis ott mind az $\{1, 2, 4\}$, mind pedig a $\{2, 3, 4\}$ csúcsok által feszített részgráf nem erősen összefüggő, viszont az algoritmus ezen lépése után pontosan az egyik biztosan erősen összefüggő részgráfot alkot majd, és azok az élek már nem változnak az algoritmus azutáni részében.

□