

**Nem teljesen kitöltött
páros összehasonlítás mátrixok
a többszemponútú döntésekben**

Diplomamunka

Írta: Ábele-Nagy Kristóf

Alkalmazott matematikus szak

Témavezető:

Kovács Gergely, főiskolai docens
Operációkutatási Tanszék



Eötvös Loránd Tudományegyetem
Természettudományi Kar

2010

Köszönetnyilvánítás

Köszönöm Kovács Gergelynek, hogy elvállalta a témavezetői feladatot, és hogy bármikor fordulhattam hozzá segítségért. Alapos ellenőrzése sokat segített a dolgozat megírásában.

Hálás köszönettel tartozom Bozóki Sándornak, aki felkeltette érdeklődésemet a döntésanalízis iránt; javasolta számomra a témát, és rendelkezésemre bocsátotta a legfrissebb kutatási eredményeit. Köszönöm továbbá, hogy a munka során mindvégig számíthattam gyors és segítőkész támogatására.

Tartalomjegyzék

1. Bevezetés	3
2. Súlyozás és páros összehasonlítás	5
2.1. Páros összehasonlítás mátrixok	5
2.2. Inkonzisztencia	7
3. Nem teljesen kitöltött páros összehasonlítás mátrixok	9
4. A λ_{max} parciális deriváltjai	11
5. Létezés és egyértelműség	14
5.1. Átskálázás	14
5.2. Gráf reprezentáció	15
5.3. Egyértelműség	16
6. Algoritmus az optimális kitöltésre	17
6.1. Az általános algoritmus	17
6.2. A Newton-módszer alkalmazása az egyváltozós optimalizálásra	18
6.3. Ciklizálás	21
7. Többváltozós Newton-módszer	24
8. Számítási eredmények	29
8.1. Példák az algoritmusok működésére	29
8.2. Véletlengenerált tesztek	36
8.3. A többváltozós módszer stabilitása és a γ szerepe	39

9. Konklúzió	42
Irodalomjegyzék	43

1. fejezet

Bevezetés

A mindennapi életben sokszor nézünk szembe döntési problémákkal. Ezek többsége rendszerint nagyon egyszerű; gyakran pedig nincs időnk elemezni a problémát. Ilyenkor rövid mérlegelés után gyors döntést hozunk. Sokszor azonban komolyabb, nagyobb hatású döntési problémák kerülnek elénk, ahol körültekintéssel kell meghoznunk a döntést. Általában több szempontunk és több alternatívánk is van. Tipikus példa erre a közbeszerzési eljárás. A mindennapi életből is lehet példát meríteni, például: tegyük fel, hogy autót szeretnénk vásárolni. A piacon sokféle autó kapható, és mindegyiknek vannak olyan tulajdonságai amelyek alapján összehasonlíthatjuk őket. Például: végsebesség, fogyasztás, ár, de akár olyan szubjektív szempontok is, mint a kényelem vagy a szín. Ezeknek a szempontoknak és a rendelkezésre álló alternatíváknak az elemzése hasznos lehet, hogy végül a lehető leginkább elégedettek legyünk a döntésünkkel.

A döntéshozótól származó szubjektív információ mennyisége nem mindig elégséges a legjobb döntés meghozatalához, a döntést ennek ellenére meg kell hozni. Ezért az lesz a célunk, hogy ilyen körülmények között is, a rendelkezésre álló információ alapján – a hiányzó információt áthidalva – optimális döntést hozzunk.

A második fejezetben bemutatjuk a többszemponú döntésmélet egyik alapvető fogalmát, a páros összehasonlítás mátrixokat, és definiáljuk az inkonzisztenciát, és azt, hogy ez hogyan függ a mátrix legnagyobb sajátértékétől.

A harmadik fejezetben megismerkedünk a nem teljesen kitöltött páros összehasonlítás mátrixokkal, amelyekkel a hiányos információjú feladatokat tudjuk kezelni. A továbbiakban igyekszünk módszert adni arra, hogy egy nem teljesen kitöltött páros összehasonlítás mátrixot hogyan lehet az inkonzisztenciát minimalizálva kitölteni.

A következő fejezet egy, a későbbiekben felhasználásra kerülő eszközt ad a kezünkbe: képletet, amellyel kiszámolhatjuk egy páros összehasonlítás mátrix legnagyobb sajátértékének a mátrix elemei szerinti parciális deriváltjait .

Az ötödik fejezetben tárgyaljuk módszerünk korlátait, azaz a megoldás létezését és egyértelműségét.

A hatodik fejezetben rátérünk az új eredményekre, azaz arra, hogyan tudjuk az egyváltozós Newton-módszert alkalmazni a problémánk optimális megoldására, a hetedik fejezetben pedig további új eredményeket, a többváltozós Newton-módszer ugyanilyen célú alkalmazását mutatjuk be.

A nyolcadik fejezet konkrét számítási eredményeket mutat be, példákön és véletlen feladatokon végzett nagy számú tesztek eredményein egyaránt. Végül az utolsó fejezetben a tapasztalt eredmények összefoglalása történik.

2. fejezet

Súlyozás és páros összehasonlítás

Bemutatjuk a többszemponútú döntésméletben gyakran alkalmazott (és a Saaty-féle AHP módszer [8, 9] alappilléreinek tekinthető) páros összehasonlítás mátrixokat, és azt, hogy ezek segítségével hogyan dönthetünk a számunkra legjobb alternatíva mellett.

2.1. Páros összehasonlítás mátrixok

Ha szemponkjainkat sikerült mértékegységektől függetlenné tennünk, akkor természetesen adódik, hogy az alternatívákat a súlyozott tulajdonságaiknak megfelelően rangsoroljuk, azaz

$$S_i = \sum_j (w_j x_{ij})$$

alapján, ahol

w_j a j szemponthoz rendelt súly, $\sum_j (w_j) = 1$, $w_i \geq 0$,

x_{ij} az i alternatíva j szempontra szerinti értéke,

S_i pedig az i alternatíva súlyozott értékösszege.

Feltehető, hogy $w_i > 0$, mert a 0 súlyú szemponthoz elhagyhatjuk. Az S_i szerinti rangsor megadja a mi szubjektív preferenciáink szerinti rangsort, feltéve, hogy a súlyokat ennek megfelelően állítottuk be.

De hogyan határozzuk meg a súlyokat? Aligha kérhetjük meg rá a döntéshozót, hogy egy, az ő preferenciáit megfelelően tükröző súlyvektort szolgáltatson nekünk. A következő kérdés azonban sokkal könnyebben megválaszolható: Hányszor fontosabb az A szempont a B szempontnál? A gyakorlat azt mutatja, hogy erre a kérdésre általában viszonylag pontos választ lehet várni. Ha minden szempontpárra feltesszük ezt a kérdést, akkor ezekből közvetetten meghatározhatjuk a szempontok súlyait: a kérdésre kapott r_{ij} értékeket (azaz r_{ij} azt mondja meg, hányszor fontosabb az i szempont a j szempontnál) az R mátrixba rendezve, ún. *páros összehasonlítás mátrixot* kapunk, melynek elemei a következő tulajdonságokkal rendelkeznek:

$$r_{ij} = \frac{1}{r_{ji}}, \quad r_{ij} > 0, \quad r_{ii} = 1.$$

Itt az első tulajdonság azért igaz, mert a w_i súlyok az i szempont fontosságát adják; tehát azt, hogy egy szempont hányszor fontosabb egy másiknál, a két súly hányadosával állapíthatjuk meg, azaz $r_{ij} = \frac{w_i}{w_j}$. A második és a harmadik tulajdonság nyilvánvaló.

Egy páros összehasonlítás mátrixot *konzisztensnek* nevezünk, ha $r_{ij} = r_{ik}r_{kj}, \forall i, j, k$. Ideális esetben ez teljesül, hiszen

$$r_{ik}r_{kj} = \frac{w_i}{w_k} \frac{w_k}{w_j} = \frac{w_i}{w_j} = r_{ij}.$$

Tehát az r_{ij} elemeket az R mátrixba, a keresett w_i súlyokat a w vektorba gyűjtve, az

$$Rw = mw$$

összefüggés teljesül, ahol m az R mátrix egyetlen nemnulla sajátértéke (az R rangja láthatóan 1, mivel az összes sor az első skalárszorosa), w pedig a hozzá tartozó normált sajátvektor, ami megadja nekünk a keresett súlyokat. Itt m az R mátrix dimenziója, hiszen egy mátrix nyoma a sajátértékeinek összege, és mivel a főátlóban csupa 1-es van, a nyom éppen a dimenziószám lesz, és egy kivételével az összes sajátérték 0. A w súlyvektor ilyen módon való meghatározását nevezzük sajátérték módszernek [8, 9].

2.2. Inkonzisztencia

Egy élő döntéshozótól rendszerint nem várható el, hogy konzisztens mátrixot produkáljon, amikor meghatározza a fontosságok egymáshoz való arányát. A súlyok egy becslését ekkor is meg lehet határozni, az

$$Rw = \lambda_{max}w$$

sajátértékfeladat megoldásával. Itt λ_{max} az R mátrix legnagyobb sajátértéke, ami konzisztens esetben az egyetlen nemnulla sajátérték. A Perron–Frobenius-tétel szerint λ_{max} egyszeres, valós, és a hozzá tartozó sajátvektor elemei szigorúan pozitívak.

Saaty bebizonyította [9], hogy $\lambda_{max} \geq m$, ezért a két érték különbségét fel lehet használni egy *konzisztencia mérőszám* meghatározására [8, 9]:

$$CR = \frac{CI}{ACI},$$

ahol

$$CI = \frac{\lambda_{max} - m}{m - 1}.$$

Itt ACI véletlengenerált páros összehasonlítás mátrixok átlagos indexe, ez felírható így:

$$ACI = \frac{\bar{\lambda}_{max} - m}{m - 1},$$

ahol $\bar{\lambda}_{max}$ véletlengenerált páros összehasonlítás mátrixok átlagos legnagyobb sajátértéke. ACI értéke így minden m -re egy átlagos érték, ami táblázatba foglalható [12]:

m	3	4	5	6	7	8
ACI	0.523862	0.888663	1.107644	1.253422	1.339445	1.403563
9	10	11	12	13	14	15
1.452397	1.488691	1.515705	1.533726	1.548214	1.571806	1.584318

Saaty [8, 9] javaslata alapján a $CR = 0,1$ elfogadható küszöbszámnak

tekinthető. Látható, hogy az inkonzisztencia a λ_{max} lineáris függvénye, ezért a legnagyobb sajátértékre adott állítások általában átvihetők az inkonzisztenciára adott állításokba.

Néha nem csak az lehet a probléma, hogy az inkonzisztencia túl nagy, hanem esetleg nincs is elég információnk, azaz elég összehasonlításunk, hogy kitöltsük a mátrixot. A következőkben ennek a problémának a kezelésével foglalkozunk.

3. fejezet

Nem teljesen kitöltött páros összehasonlítás mátrixok

Definiáljuk a nem teljesen kitöltött páros összehasonlítás mátrixokat, ami alapvető fontosságú lesz a későbbiekben. A nem teljesen kitöltött páros összehasonlítás mátrixokat Harker vezette be először [3, 4].

Előfordulhat, hogy nem áll rendelkezésünkre az összes páros összehasonlítás, vagy a mátrix olyan nagy, hogy nagyon körülményes lenne minden elemét kitölteni ($\frac{m(m-1)}{2}$ darab elemre van szükségünk), esetleg a döntéshozó nem tudja elég biztonsággal megadni a fontosság arányát. Ekkor olyan mátrixot kapunk, ami nincs teljesen kitöltve. Egy *nem teljesen kitöltött páros összehasonlítás mátrix* tehát az alábbihoz hasonló formát ölti, ahol a * hiányzó elemet jelöl:

$$\mathbf{R} = \begin{pmatrix} 1 & r_{12} & * & \dots & r_{1n} \\ 1/r_{12} & 1 & r_{23} & \dots & * \\ * & 1/r_{23} & 1 & \dots & r_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1/r_{1n} & * & 1/r_{3n} & \dots & 1 \end{pmatrix}$$

Természetesen a hiányzó elemek (a főátlót kivéve) bárhol lehetnek, és ha r_{ij} hiányzik, akkor r_{ji} is. Egy nem teljesen kitöltött mátrix tulajdonképpen nem is mátrix, de valamelyest kezelhetővé válik, ha a hiányzó elemekre mint vál-

tozókra gondolunk. E célból vezessük be az $x_1, x_2, \dots, x_d \in \mathbb{R}_+$ változókat az R felső háromszögből hiányzó elemekre. A reciprokaik, $1/x_1, 1/x_2, \dots, 1/x_d \in \mathbb{R}_+$ kerülnek az alsó háromszögbe, így a hiányzó elemek száma összesen $2d$. Jelölje

$$\mathbf{R}(x) = \mathbf{R}(x_1, x_2, \dots, x_d) = \begin{pmatrix} 1 & r_{12} & x_1 & \dots & r_{1n} \\ 1/r_{12} & 1 & r_{23} & \dots & x_d \\ 1/x_1 & 1/r_{23} & 1 & \dots & r_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1/r_{1n} & 1/x_d & 1/r_{3n} & \dots & 1 \end{pmatrix}$$

ahol $x = (x_1, x_2, \dots, x_d)^T \in \mathbb{R}_+^d$. Így akár úgy is tekinthetünk a nem teljesen kitöltött páros összehasonlítás mátrixokra, mint az $x \in \mathbb{R}_+^d$ által generált összes kitöltött mátrixok halmaza.

Shiraishi, Obata és Daigo nyomán [10, 11] a végső célunk az, hogy a mátrix *optimális kitöltését* adjuk meg, abban az értelemben, hogy az inkonzisztenciát (azaz ekvivalensen λ_{max} -ot) minimalizáljuk, tehát

$$\min_{x \in \mathbb{R}_+^d} \lambda_{max}(R(x))$$

értékét, és a minimum helyét (az optimális x vektort) keressük. A továbbiakban ezzel a feladattal foglalkozunk, de az optimum meghatározásához először is szükségünk lesz a λ_{max} -nak a mátrix elemei szerinti parciális deriváltjaira.

4. fejezet

A λ_{max} parciális deriváltjai

Hogy a legnagyobb sajátérték minimalizálásának problémáját meg tudjuk oldani, fel fogjuk használni a λ_{max} parciális deriváltjait. Nézzük ezek hogyan határozhatóak meg.

Harker bebizonyította [5], hogy egy A páros összehasonlítás mátrix legnagyobb sajátértékének (vagy más néven Perron-sajátértékének) léteznek az A elemei szerinti összes parciális deriváltjai, sőt ki is számolta az első- és másodrendűeket explicit formában.

Jelölje x az A jobb oldali Perron-sajátvektorát (azaz a legnagyobb sajátértékhez, λ_{max} -hoz tartozó jobb oldali sajátvektort):

$$Ax = \lambda_{max}x$$

y pedig jelölje a bal oldali Perron-sajátvektort, azaz:

$$y^T A = \lambda_{max}y^T.$$

A két sajátvektor pontosan akkor egymás elemenkénti reciproka, ha a mátrix konzisztens.

Fontos, hogy itt a sajátvektorokat nem a megszokott módon kell normálni, hanem úgy, hogy a skalárszorzatuk 1 legyen, azaz

$$y(B)^T x(B) = 1.$$

Ezekkel a jelölésekkel meghatározhatóak a $\lambda_{max}(A)$ első és második parciális deriváltjai, ami általánosabb mátrix osztályban is igaz, a négyzetes, valós, nemnegatív, irreducibilis mátrixok körében, ahol az irreducibilitás azt jelenti, hogy bármely $i \neq j$ indexpárra létezik egy nemnulla elemekből álló $a_{ii_1}, a_{i_1i_2}, \dots, a_{i_rj}$ lánc az A mátrixban.

$$D_1^A := \left(\frac{\partial \lambda_{max}(A)}{\partial_{ij}} \right) = yx^T,$$

$$D_2^A := \left(\frac{\partial^2 \lambda_{max}(A)}{\partial_{ij} \partial_{kl}} \right).$$

Itt

$$\frac{\partial^2 \lambda_{max}(A)}{\partial_{ij} \partial_{kl}} = (I - QQ^+)_{li} (Q^+)_{jk} + (I - QQ^+)_{jk} (Q^+)_{li},$$

ahol

$$Q = \lambda_{max}(A)I - A$$

és Q^+ a Q pszeudoinverze, azaz Q^+ teljesíti az alábbi három feltételt:

1. $QQ^+Q = Q$
2. $Q^+QQ^+ = Q^+$
3. $Q^+Q = QQ^+$

Nekünk azonban ennél az osztálynál speciálisabb mátrixaink vannak, nevezetesen páros összehasonlítás mátrixok, ahol kihasználhatjuk, hogy az alsó háromszögben lévő elemek a felsők függvényei, konkrétan ha $j > i$, azaz a_{ij} a felső háromszögben van, akkor $a_{ji}(a_{ij}) = 1/a_{ij}$. Sőt, azt is tudjuk, hogy a főátlóban lévő elemek értéke 1. Így csak a felső háromszögben lévő $n(n-1)/2$ elem szerinti deriváltak érdekesek.

Legyen

$$\tilde{D}_1^A = \left(\frac{\partial \lambda_{max}(A)}{\partial_{ij}} \Big|_{i > j} \right)$$

és

$$\tilde{D}_2^A = \left(\frac{\partial^2 \lambda_{max}(A)}{\partial_{ij} \partial_{kl}} \Big|_{i > j, k > l} \right).$$

A bevezetett jelölésekkel, Harker szerint:

$$\tilde{D}_1^A = \left([y(A)_i x(A)_j] - \frac{[y(A)_j x(A)_i]}{[a_{ij}]^2} \right),$$

ahol $x(A)$ és $y(A)$ rendre az A jobb és a bal oldali Perron-sajátvektora;

$$\tilde{D}_2^A = \left(\frac{\partial^2 \lambda_{max}(A)}{\partial_{ij} \partial_{kl}} \Big|_{i > j, k > l} \right)$$

pedig az alábbi módon határozható meg:

$$\frac{\partial^2 \lambda_{max}(A)}{\partial_{ij} \partial_{kl}} = \left\{ \begin{array}{l} (x(A)y(A)^T)_{li} Q_{jk}^+ + (x(A)y(A)^T)_{jk} Q_{li}^+ - \\ - \frac{(x(A)y(A)^T)_{ki} Q_{jl}^+ + (x(A)y(A)^T)_{jl} Q_{ki}^+}{[a_{kl}]^2} - \\ - \frac{(x(A)y(A)^T)_{lj} Q_{ik}^+ + (x(A)y(A)^T)_{ik} Q_{lj}^+}{[a_{ij}]^2} + \\ + \frac{(x(A)y(A)^T)_{kl} Q_{il}^+ + (x(A)y(A)^T)_{il} Q_{kj}^+}{[a_{ij}]^2 [a_{kl}]^2} \end{array} \right. \quad \text{ha } i \neq k \text{ vagy } j \neq l$$

$$\left\{ \begin{array}{l} \frac{2(x(A)y(A)^T)_{ij}}{[a_{ij}]^3} + 2(x(A)y(A)^T)_{ji} Q_{ii}^+ - \\ - 2 \frac{(x(A)y(A)^T)_{ii} Q_{jj}^+ + (x(A)y(A)^T)_{jj} Q_{ii}^+}{[a_{ij}]^2} + \\ + 2 \frac{(x(A)y(A)^T)_{ij} Q_{ij}^+}{[a_{ij}]^4} \end{array} \right. \quad \text{ha } i = k \text{ és } j = l$$

Ezekkel a képletekkel előállíthatjuk a mátrix elemei szerinti első és második parciális deriváltakat, és így készen állunk a Newton-módszer használatára, de előbb meg kell vizsgálnunk, hogy ezt milyen korlátok és körülmények között tehetjük meg.

5. fejezet

Létezés és egyértelműség

Az előző fejezetben ismertetett deriváltak segítségével már megkaptuk az eszközt a Newton-módszer alkalmazására. Meg kell azonban néznünk, hogy ennek mikor van egyáltalán értelme, azaz, hogy mikor létezik a feladatnak egyértelmű megoldása. Ebben a részben a Bozóki–Fülöp–Rónyai-cikk eredményei alapján haladunk [2].

5.1. Átskálázás

A λ_{max} minimum létezésének kulcsa, hogy a

$$\min_{x \in \mathbb{R}_+^d} \lambda_{max}(R(x))$$

feladatot át lehet transzformálni egy *konvex optimalizálási feladattá*, a következő módon:

Parametrizáljuk a nem teljesen kitöltött mátrixunkat, $A(x) = A(x_1, x_2, \dots, x_d)$ -t úgy, hogy $x_i = e^{y_i}$, ($i = 1, 2, \dots, d$). Így kapjuk a B mátrixot:

$$A(x) = B(y) = B(y_1, y_2, \dots, y_d) = A(e^{y_1}, e^{y_2}, \dots, e^{y_d}).$$

A parametrizált $B(y)$ nem teljesen kitöltött mátrixra a $\lambda_{max}(B(y))$ az y -nak logkonvex – ebből következően konvex – függvénye [1, 6, 2]. (*Logkonvexnek* nevezünk egy f függvényt, ha $\log f$ konvex függvény.)

Sőt, $\lambda_{max}(B(y))$ vagy szigorúan konvex, vagy konstans bármely egyenes mentén az y terében. Így a feladatunkat egy *szigorúan konvex optimalizálási feladattá* alakítottuk.

5.2. Gráf reprezentáció

Szeretnénk meghatározni, hogy a λ_{max} optimalizálására mikor van egyáltalán reményünk. Ehhez először definiáljuk a nem teljesen kitöltött páros összehasonlítás mátrixok *gráf reprezentációját*.

Legyen tehát adott egy A $n \times n$ -es nem teljesen kitöltött páros összehasonlítás mátrix (ez persze teljesen kitöltött mátrixokra is érvényes definíció, itt érdemes úgy gondolni a kitöltött mátrixokra, mint speciális nem teljesen kitöltött mátrixokra), ekkor a hozzá tartozó G *irányítatlan gráf* a következőképpen határozható meg:

$$G := (V, E), \text{ ahol } V = 1, 2, \dots, n$$

ahol az a pontok az összehasonlítandó szempontoknak felelnek meg, az élek pedig a mátrix kitöltött elemeinek:

$$E = \{e(i, j) | a_{ij} \text{ meg van adva (és ekkor persze } a_{ji} \text{ is adott), és } i \neq j\}.$$

Tehát két különböző szempontnak megfelelő pont között pontosan akkor megy él, ha a két szempont össze van hasonlítva, azaz az összehasonlításuknak megfelelő két elem a mátrixban ki van töltve.

A páros összehasonlítás mátrixhoz rendelhetünk egy *irányított gráfot* is, a következőképpen:

$\vec{G} := (V, \vec{E})$, ahol V ugyanaz, mint az irányítatlan gráf esetében, viszont az élek irányítottak, és minden összehasonlításához két él tartozik, de minden ponthoz tartozik egy hurokél is, az alábbiak szerint:

$$\vec{E} = \{\overrightarrow{e(i, j)} | a_{ij} \text{ meg van adva és } i \neq j\} \cup \{\overrightarrow{e(i, i)} | i = 1, 2, \dots, n\}.$$

Azaz pontosan akkor megy él i -ből j -be, ha a mátrix a_{ij} eleme ki van töltve (ügyelve arra, hogy a hurokélek egyszerezsek). Ez a reprezentáció azért jó, mert a gráf éleihez hozzárendelve a megfelelő arányt, könnyedén visszakaphatjuk a mátrixot.

5.3. Egyértelműség

Az irányítatlan gráf reprezentációval könnyen karakterizálhatjuk az egyértelműséget [2]:

A $\lambda_{max}(A(x))$ minimalizálási feladatnak pontosan akkor egyértelmű a megoldása, ha az A nem teljesen kitöltött páros összehasonlítás mátrixhoz tartozó G irányítatlan gráf összefüggő.

Sőt, az az általánosabb tétel is igaz, hogy az átparaméterezett $B(y)$ mátrixra a $\lambda_{max}(B(y))$ függvény felveszi a minimumát \mathbb{R}^d -n, és az optimális megoldások $(s-1)$ dimenziós affin halmazát alkotják \mathbb{R}^d -nek, ahol s az összefüggő komponensek száma G -ben.

Ezen elméleti háttér, és a korábban bemutatott deriváltak ismeretében minden akadály elhárult az elől, hogy algoritmust adjunk az optimális kitöltésre.

6. fejezet

Algoritmus az optimális kitöltésre

A sajátérték optimalizálásának feladatát a korábban ismertetett parciális deriváltak segítségével már meg tudjuk oldani, feltéve, hogy az előző fejezetben ismertetett feltételek teljesülnek a megoldás egyértelmű létezésére. Az itt bemutatott algoritmus a Bozóki–Rónyai–Fülöp-cikkben leírt algoritmuson alapszik [2], de a függvényoptimalizálást itt az analitikus formában megadott parciális deriváltak segítségével felírt Newton-módszerrel végezzük.

6.1. Az általános algoritmus

Az algoritmus egyváltozós problémákat old meg, egyszerre mindig egy változó szabad, a szabad változóban optimalizálunk, majd a következő változóra ugunk, így végigmegyünk az összes hiányzó elemen, majd iteráljuk az algoritmust.

Jelölje d az M nem teljesen kitöltött páros összehasonlítás mátrix felső háromszögéből hiányzó elemek számát. Fontos, hogy az M gráfja összefüggő legyen. Írjuk az x_1, x_2, \dots, x_d változókat a felső háromszögből hiányzó elemek helyére, és $1/x_1, 1/x_2, \dots, 1/x_d$ -t az alsó háromszögből hiányzó elemek helyére (természetesen úgy, hogy ha x_k az M mátrix ij -dik pozíciójába került, akkor $1/x_k$ a ji -dikbe kerül). Legyenek $x_i^{(0)}, i = 1, 2, \dots, d$ tetszőleges pozitív számok, ezek lesznek az induló értékeink. Minden iteráció d lépésből áll.

Az első iteráció elején legyen x_1 a szabad változónk, a többi változó pedig

legyen rögzített az $x_i = x_i^{(0)}$, $i = 2, 3, \dots, d$ értékekkel. Most optimalizáljuk a λ_{max} -ot az x_1 egyváltozós függvényeként (a konkrét alkalmazásban ezt fogjuk majd Newton-módszerrel csinálni). Mivel a többváltozós feladat áttranszformálható egy többváltozós konvex optimalizálási feladattá, akkor is konvex marad, ha egy változóra szorítjuk meg.

Legyen az egyváltozós optimum $x_1^{(1)}$. Az első iteráció második lépésében x_2 szabad, a többi változó pedig rögzített $x_1 = x_1^{(1)}$, $x_i = x_i^{(0)}$, $i = 3, 4, \dots, d$ módon. Ennek az optimuma legyen $x_2^{(1)}$.

Értelemszerű folytatással kapjuk a többi $x_i^{(1)}$ értéket. Az utolsó lépésben a feladat λ_{max} -ot minimalizálni x_d -ben, úgy, hogy $x_i = x_i^{(1)}$, $i = 1, 2, \dots, d-1$. Ennek az optimális megoldása legyen $x_d^{(1)}$, és ezzel befejeződik az algoritmus első iterációja.

A k -dik iterációban kiindulóértékként az előző iterációban számított $x_i^{(k-1)}$ értékeket használjuk, azaz

$$x_i^{(k)} = \arg \min_{x_i} \lambda_{max}(M(x_1^{(k)}, \dots, x_{i-1}^{(k)}, x_i, x_{i+1}^{(k-1)}, \dots, x_d^{(k-1)})), \quad i = 1, 2, \dots, d.$$

A megállási szabályt sokféleképpen meg lehet határozni; mi egy 4 tizedesjegy pontosságú kritériumot használunk, azaz: az algoritmus megáll a k -dik iteráció végén, ha k a legkisebb olyan egész, amire

$$\max_{i=1,2,\dots,d} \|x_i^{(k)} - x_i^{(k-1)}\| < T,$$

ahol T a toleranciaküszöb, a mi esetünkben ez 10^{-4} .

A ciklikus koordinátákkal történő optimalizálás globális konvergenciája be van bizonyítva [7].

6.2. A Newton-módszer alkalmazása az egyváltozós optimalizálásra

Nézzük, hogy a rendelkezésünkre álló parciális deriváltakat hogyan tudjuk felhasználni az optimalizálásra. Az ismert szélsőérték keresésre alkalmazható

Newton-módszerből, azaz az

$$x_{n+1} = x_n - \frac{f'(x_n)}{f''(x_n)}$$

iterációból indulunk ki, azonban az átsákálzás miatt nem használhatjuk tisztán ezt a módszert.

Mivel egyszerre csak egy változó szerint minimalizálunk, tekinthetjük a többi változó értékét adottnak, mint ahogy az algoritmus konkrét futásakor azok is: ha a k -dik iterációban vagyunk, és az x_i változó szerint optimalizálunk, akkor $x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_d$ rögzítve van az

$$x_1 = x_1^{(k)}, \dots, x_{i-1} = x_{i-1}^{(k)}, x_{i+1} = x_{i+1}^{(k-1)}, \dots, x_d = x_d^{(k-1)}$$

konkrét értékekre, és csak az x_i a tényleges változó. Ezért feltehetjük, hogy x_i az egyetlen változó, a következőkben ezért index nélkül x -el jelöljük.

Harker alapján ismertek a $\frac{\partial \lambda_{max}(x)}{\partial x}$ és $\frac{\partial^2 \lambda_{max}(x)}{(\partial x)^2}$ kifejezések.

Legyen $x = e^t$ és $L(t) = \lambda_{max}(e^t)$. Határozzuk meg az $\frac{\partial L(t)}{\partial t}$ és $\frac{\partial^2 L(t)}{(\partial t)^2}$ deriváltakat:

$$\frac{\partial L(t)}{\partial t} = \frac{\partial \lambda_{max}(e^t)}{\partial t} = \frac{\partial \lambda_{max}(x)}{\partial x} \cdot \frac{\partial e^t}{\partial t} = \frac{\partial \lambda_{max}(x)}{\partial x} \cdot e^t.$$

Hasonlóan,

$$\frac{\partial^2 L(t)}{(\partial t)^2} = \frac{\partial^2 \lambda_{max}(e^t)}{(\partial t)^2} = \frac{\frac{\partial \lambda_{max}(x)}{\partial x} \cdot e^t}{\partial t} = \frac{\frac{\partial \lambda_{max}(x)}{\partial x}}{\partial t} \cdot e^t + \frac{\partial \lambda_{max}(x)}{\partial x} \cdot \frac{\partial e^t}{\partial t}.$$

Az első együtthatót átírhatjuk:

$$\frac{\frac{\partial \lambda_{max}(x)}{\partial x}}{\partial t} = \frac{\frac{\partial \lambda_{max}(x)}{\partial x}}{\partial x} \cdot \frac{\partial x}{\partial t} = \frac{\partial^2 \lambda_{max}(x)}{(\partial x)^2} \cdot e^t.$$

Kapjuk tehát, hogy

$$\frac{\partial^2 L(t)}{(\partial t)^2} = \frac{\partial^2 \lambda_{max}(x)}{(\partial x)^2} \cdot e^{2t} + \frac{\partial \lambda_{max}(x)}{\partial x} \cdot e^t.$$

Mivel az $L'(t) = 0$ egyenlet megoldását keressük, a Newton-iteráció így írható fel:

$$t_{n+1} = t_n - \frac{L'(t_n)}{L''(t_n)} = t_n - \frac{\frac{\partial \lambda_{max}(x)}{\partial x} \cdot e^{t_n}}{\frac{\partial^2 \lambda_{max}(x)}{(\partial x)^2} \cdot e^{2t_n} + \frac{\partial \lambda_{max}(x)}{\partial x} \cdot e^{t_n}} = t_n - \frac{\frac{\partial \lambda_{max}(x)}{\partial x}}{\frac{\partial^2 \lambda_{max}(x)}{(\partial x)^2} \cdot e^{t_n} + \frac{\partial \lambda_{max}(x)}{\partial x}}.$$

Mivel a változónk, x , szerinti parciális deriváltak valójában az x -nek a mátrixban elfoglalt pozíciójának, (i, j) -nek a függvénye, jelölje

$$d^1(i, j) = \frac{\partial \lambda_{max}(x)}{\partial x}.$$

A másodrendű deriváltak négy indextől függenek, (i, j, k, l) -től, ahol (i, j) az egyik elem (ami szerint deriválunk) pozíciója a mátrixban, (k, l) a másiké. Azonban az iterációban mindkétszer ugyanazon elem (x) szerinti deriváltat kell venni, azaz csupán két indextől függ, mert most $(i, j) = (k, l)$, vagyis értelmes a jelölés

$$d^2(i, j) = \frac{\partial^2 \lambda_{max}(x)}{(\partial x)^2}.$$

Mindkét jelölés esetében (i, j) az x pozíciója.

Ezekkel a jelölésekkel tehát az eredeti Newton-iteráció így nézne ki:

$$x_{n+1} = x_n - \frac{d_n^1(i, j)}{d_n^2(i, j)},$$

ahol $d_n^1(i, j)$ és $d_n^2(i, j)$ értelemszerűen a Newton-iterációban számolt x_n értékével behelyettesített érték, azaz

$$d_n^1(i, j) = \frac{\partial \lambda_{max}(x_n)}{\partial x_n}, \quad d_n^2(i, j) = \frac{\partial^2 \lambda_{max}(x_n)}{(\partial x_n)^2}.$$

Most legyen $x = e^t$, azaz $t = \ln x$. Ekkor az előző levezetés alapján, és a jelöléseket használva, az iteráció ezt az alakot ölti:

$$t_{n+1} = t_n - \frac{d_n^1(i, j)}{d_n^2(i, j) \cdot e^{t_n} + d_n^1(i, j)}.$$

Feltettük, hogy csak egy változónk van (mivel egyszerre csak egy változó

szerint minimalizálunk), ezért az A mátrix így néz ki:

$$A = \begin{pmatrix} 1 & \dots & a_{1i} & \dots & a_{1j} & \dots & a_{1n} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{i1} & \dots & 1 & \dots & x & \dots & a_{in} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{j1} & \dots & 1/x & \dots & 1 & \dots & a_{jn} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{ni} & \dots & a_{nj} & \dots & 1 \end{pmatrix}$$

Itt tehát x az (i, j) pozícióban van, persze így $1/x$ a (j, i) pozícióban. Az x -et be kell állítanunk egy kezdőértékre, azaz, kezdetben legyen $x = x_0$ (például $x_0 = 1$). Az x helyére minden lépésben beírjuk x_n -et, kezdetben x_0 -t.

Egy teljes Newton-iterációs lépés tehát a következőkből áll:

1. $t_n := \ln a_{ij} = \ln x_n$
2. Kiszámoljuk $d_n^1(i, j)$ -t és $d_n^2(i, j)$ -t
3. $t_{n+1} = t_n - \frac{d_n^1(i, j)}{d_n^2(i, j) \cdot e^{t_n} + d_n^1(i, j)}$
4. $a_{ij} = x_{n+1} := e^{t_{n+1}}$, $a_{ji} = 1/x_{n+1} := e^{-t_{n+1}}$

Ezt a négy lépést iteráljuk, például egy előre meghatározott számú alkalommal. Tapasztalat alapján a 25 egy jó iterációs számnak bizonyult. Az iterációs eljárás végén kapott x az optimális kitöltés, az x értékét behelyettesítve kapott mátrix legnagyobb sajátértéke pedig az optimális λ_{max} , ha a többi elem fix. Azaz: ha esetleg csak egy elem hiányzik a felső háromszögben, akkor készen is vagyunk.

6.3. Ciklizálás

Ha több hiányzó elemünk is van, akkor ciklizálnunk kell őket, és külön-külön – az éppen aktuális változót kivéve mindnek az értékét rögzítve – futtatni rájuk a Newton-iterációt. Gyűjtsük tehát az x vektorba a hiányzó ele-

meket, $x = (x_1, \dots, x_d)$. Legyen a 3. fejezetben bevezetett jelölést használva

$$A(x_1, x_2, \dots, x_d) = \begin{pmatrix} 1 & a_{12} & x_1 & \dots & a_{1n} \\ 1/a_{12} & 1 & a_{23} & \dots & x_d \\ 1/x_1 & 1/a_{23} & 1 & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1/a_{1n} & 1/x_d & 1/a_{3n} & \dots & 1 \end{pmatrix}$$

Természetesen az x_i hiányzó elemeket jelölő változók a felső háromszög bármely pozíciójában lehetnek. A Newton-módszert szubrutinként használva jelölje $x_i^N(A)$ az x_i változó Newton-iteráció által számolt optimális értékét az A mátrixban. Kiindulási értékeket is rögzítenünk kell: legyen $x_i = x_i^{(0)}$, $i = 1, 2, \dots, d$ (lehet például $x_i^{(0)} = 1$, $i = 1, 2, \dots, d$). Így az algoritmus $k+1$ -dik iterációs lépése az x_i változóra a következő formát ölti:

$$x_i^{(k+1)} = x_i^N(A(x_1^{(k+1)}, \dots, x_{i-1}^{(k+1)}, x_i^{(k)}, \dots, x_d^{(k)})).$$

Az i indexet végigfuttatjuk 1-től d -ig minden k -ra.

Addig iteráljuk ezt k -ra, amíg el nem érünk valamilyen megállási kritériumot, például, ha már nem változik sokat az x vektor; azaz, mint már említettük az általános algoritmusnál

$$\max_{i=1,2,\dots,d} \|x_i^{(k)} - x_i^{(k-1)}\| < T,$$

ahol T a toleranciaküszöb.

Az egész algoritmus tehát összefoglalható így:

$$x_i := x_i^{(0)} \quad \forall i = 1, \dots, d \quad \text{és } k := 1$$

$$\text{do while } \max_{i=1,2,\dots,d} \|x_i^{(k)} - x_i^{(k-1)}\| > T$$

for $i = 1, \dots, d$

$$x_i^{(k)} = x_i^N(A(x_1^{(k)}, \dots, x_{i-1}^{(k)}, x_i^{(k-1)}, \dots, x_d^{(k-1)}))$$

$k := k + 1$

Figyeljük meg, hogy három darab iterációnk is van az algoritmusban, hiszen x_i^N -et is iterációval számoljuk, sőt, tulajdonképpen az az egész eljárás lényege.

Felmerül a gondolat, hogy ha sikerült az egyváltozós Newton-módszert adaptálnunk a feladatra, hogyan lehetne a többváltozósat alkalmazni?

7. fejezet

Többváltozós Newton-módszer

Az egyváltozós ciklizált Newton-módszerrel egy jó és alkalmas algoritmust kaptunk; nézzük meg ezután, milyen eredménnyel alkalmazható a többváltozós Newton-módszer a probléma megoldására.

Legyen $\underline{x} = (x_1, \dots, x_d)$ a d dimenziós vektorváltozónk, melynek elemei – az eddig is szokásos módon – az A páros összehasonlítás mátrix felső háromszögének hiányzó elemeinek felelnek meg.

Kiindulunk tehát a többváltozós Newton-iteráció képletéből:

$$\underline{x}_{n+1} = \underline{x}_n - [Hf(\underline{x}_n)]^{-1} \nabla f(\underline{x}_n),$$

ahol $\nabla f(\underline{x})$ az f gradiens vektora, azaz

$$\nabla f(\underline{x}) = \left(\frac{\partial f(\underline{x})}{\partial x_1}, \dots, \frac{\partial f(\underline{x})}{\partial x_d} \right)$$

$Hf(\underline{x})$ pedig az f Hesse-mátrixa, azaz

$$Hf(\underline{x}) = \begin{pmatrix} \frac{\partial^2 f(\underline{x})}{\partial x_1^2} & \frac{\partial^2 f(\underline{x})}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f(\underline{x})}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(\underline{x})}{\partial x_2 \partial x_1} & \frac{\partial^2 f(\underline{x})}{\partial x_2^2} & \cdots & \frac{\partial^2 f(\underline{x})}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f(\underline{x})}{\partial x_n \partial x_1} & \frac{\partial^2 f(\underline{x})}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f(\underline{x})}{\partial x_n^2} \end{pmatrix}$$

Látható a párhuzam az egyváltozós módszerrel: a gradiens veszi át az első

derivált szerepét, a Hesse-mátrix inverze pedig a második deriválttal való osztás szerepét.

Az egyváltozós eset analógiájára itt is át kell skáláznunk a változóinkat, és ennek megfelelően ezt a képletet is át kell alakítanunk. Most minden x_i helyére írjuk be az e^{t_i} -t, így

$$\underline{x} = (x_1, x_2, \dots, x_d) = (e^{t_1}, e^{t_2}, \dots, e^{t_d}).$$

Az A mátrix így a következő formát ölti:

$$A(\underline{x}) = A(e^{t_1}, e^{t_2}, \dots, e^{t_d}) = \begin{pmatrix} 1 & a_{12} & e^{t_1} & \dots & a_{1n} \\ 1/a_{12} & 1 & a_{23} & \dots & e^{t_d} \\ e^{-t_1} & 1/a_{23} & 1 & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1/a_{1n} & e^{-t_d} & 1/a_{3n} & \dots & 1 \end{pmatrix}$$

Legyen $\underline{t} = (t_1, \dots, t_d)$, és $L(\underline{t}) = L(t_1, \dots, t_d) = \lambda_{max}(e^{t_1}, e^{t_2}, \dots, e^{t_d})$.

Szükségünk van az L a Hesse-mátrixára és a gradiensére. Kezdjük a gradienssel:

$$\nabla L(\underline{t}) = \left(\frac{\partial L(\underline{t})}{\partial t_1}, \dots, \frac{\partial L(\underline{t})}{\partial t_d} \right).$$

Tehát ki kell számolnunk $\frac{\partial L(\underline{t})}{\partial t_i}$ -t $\forall i = 1, \dots, d$ -re. Nézzük, hogyan lehet ezt átalakítani.

$$\begin{aligned} \frac{\partial L(\underline{t})}{\partial t_i} &= \frac{\partial \lambda_{max}(e^{t_1}, \dots, e^{t_d})}{\partial t_i} = \frac{\partial \lambda_{max}(x_1, \dots, x_d)}{\partial t_i} = \\ &= \frac{\partial \lambda_{max}(x_1, \dots, x_d)}{\partial x_i} \cdot \frac{\partial x_i}{\partial t_i} = \frac{\partial \lambda_{max}(\underline{x})}{\partial x_i} \cdot e^{t_i}. \end{aligned}$$

Itt a $\frac{\partial \lambda_{max}(\underline{x})}{\partial x_i}$ érték számolható Harker alapján, mert ugyan a képlet csak egy változóra szól, és nekünk most d darab változónk van, de az $n + 1$ -dik iterációban már rendelkezésünkre állnak az előző (n -dik) iterációban számolt $x_j^{(n)}$ értékek. Ezeket az $x_j^{(n)}$ értékeket helyettesítjük be a $\frac{\partial \lambda_{max}(x_1, \dots, x_d)}{\partial x_i}$ -be. Azaz a derivált függvényt egyszerre csak egy pontban számoljuk.

Mivel a deriválásnál igazából nem számít, hogy melyik változó a k -dik (akár át is indexelhetnénk őket), csupán a mátrixban elfoglalt pozíciójuktól függ a derivált, ezért legyen ismét

$$d^1(i, j) = \frac{\partial \lambda_{max}(\underline{x})}{\partial x_k} = \frac{\partial \lambda_{max}(x_1, \dots, x_d)}{\partial x_k},$$

ahol (i, j) az x_k -nak az A mátrixban elfoglalt pozíciója.

Tekintsük most a Hesse-mátrixot

$$HL(\underline{t}) = \begin{pmatrix} \frac{\partial^2 L(\underline{t})}{\partial t_1^2} & \frac{\partial^2 L(\underline{t})}{\partial t_1 \partial t_2} & \cdots & \frac{\partial^2 L(\underline{t})}{\partial t_1 \partial t_n} \\ \frac{\partial^2 L(\underline{t})}{\partial t_2 \partial t_1} & \frac{\partial^2 L(\underline{t})}{\partial t_2^2} & \cdots & \frac{\partial^2 L(\underline{t})}{\partial t_2 \partial t_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 L(\underline{t})}{\partial t_n \partial t_1} & \frac{\partial^2 L(\underline{t})}{\partial t_n \partial t_2} & \cdots & \frac{\partial^2 L(\underline{t})}{\partial t_n^2} \end{pmatrix}$$

Látható, hogy $\frac{\partial^2 L(\underline{t})}{\partial t_i \partial t_j}$ -t kell kiszámolnunk $\forall i, j = 1, \dots, d$ -re. Nézzük tehát, hogyan alakítható ez át:

$$\frac{\partial^2 L(\underline{t})}{\partial t_i \partial t_j} = \frac{\partial^2 \lambda_{max}(e^{t_1}, \dots, e^{t_d})}{\partial t_i \partial t_j} = \frac{\partial \left(\frac{\partial \lambda_{max}(e^{t_1}, \dots, e^{t_d})}{\partial t_j} \right)}{\partial t_i} = \quad (*)$$

a belső derivált az előző alapján egyenlő $\frac{\partial \lambda_{max}(\underline{x})}{\partial x_i} \cdot e^{t_i}$ -vel, így az átalakítás továbbvihető:

$$= \frac{\partial \left(\frac{\partial \lambda_{max}(\underline{x})}{\partial x_j} \cdot e^{t_j} \right)}{\partial t_i} = \frac{\partial \left(\frac{\partial \lambda_{max}(\underline{x})}{\partial x_j} \right)}{\partial t_i} \cdot e^{t_j} + \frac{\partial \lambda_{max}(\underline{x})}{\partial x_j} \cdot \frac{\partial e^{t_j}}{\partial t_i}.$$

A második tagban a $\frac{\partial e^{t_j}}{\partial t_i}$ tényező 0, ha $i \neq j$, és e^{t_j} , ha $i = j$. Másképpen felírva,

$$\frac{\partial e^{t_j}}{\partial t_i} = e^{t_j} \cdot \chi_{\{i=j\}},$$

ahol

$$\chi_{\{i=j\}} = \begin{cases} 1 & \text{ha } i = j \\ 0 & \text{ha } i \neq j \end{cases}$$

Az első tagban

$$\frac{\partial \left(\frac{\partial \lambda_{max}(\underline{x})}{\partial x_j} \right)}{\partial t_i} = \frac{\partial \left(\frac{\partial \lambda_{max}(\underline{x})}{\partial x_j} \right)}{\partial x_i} \cdot \frac{\partial x_i}{\partial t_i} = \frac{\partial \left(\frac{\partial \lambda_{max}(\underline{x})}{\partial x_j} \right)}{\partial t_i} \cdot \frac{\partial e^{t_i}}{\partial t_i} = \frac{\partial^2 \lambda_{max}(\underline{x})}{\partial x_i \partial x_j} \cdot e^{t_i}.$$

Ezeket visszaírva (*)-ba kapjuk:

$$\frac{\partial^2 L(\underline{t})}{\partial t_i \partial t_j} = \frac{\partial^2 \lambda_{max}(\underline{x})}{\partial x_i \partial x_j} \cdot e^{t_i+t_j} + \frac{\partial \lambda_{max}(\underline{x})}{\partial x_i} \cdot e^{t_i} \cdot \chi_{\{i=j\}}.$$

Figyeljük meg, hogy ez speciális esetként tartalmazza az egyváltozós esetet, ugyanis egy változóra szükségképpen $i = j$, a vektorjelölések eltűnnek, és ekkor ugyanazt a képletet kapjuk a második deriváltra, mint amit az egyváltozós esetben kaptunk. Sajnos azonban itt nem hajthatunk végre egyszerűsítéseket, mert ha ezeket az értékeket beírjuk a mátrixba, nem lesz olyan tényező, ami bármely koordinátán lévő elemből kiemelhető lenne.

A másodrendű deriváltakat, azaz $\frac{\partial^2 \lambda_{max}(\underline{x})}{\partial x_i \partial x_j}$ -ket is tudjuk számolni az elsőrendű deriváltakhoz hasonló módon Harker képlete alapján. Vegyük ismét észre, hogy ezek a deriváltak is csupán az x_k -k mátrixbeli elhelyezkedésétől függenek, nem azok konkrét indexétől, hiszen akár át is indexelhetnénk őket. Így itt is jogos a jelölés:

$$d^2(i, j, k, l) = \frac{\partial^2 \lambda_{max}(\underline{x})}{\partial x_p \partial x_q},$$

ahol (i, j) az x_p , (k, l) pedig az x_q helyének koordinátái a mátrixban. Az egyváltozós esettől eltérően itt az összes másodrendű deriváltra szükségünk van, ezért nem csak azt az esetet kell néznünk, amikor p azonos q -val, így $d^2(i, j, k, l)$ valódi négyindexes függvény marad. Jelöljük ezért most t_{ij} -vel t_p -t, ahol persze (i, j) a t_p (ekvivalensen az x_p) helye a mátrixban.

Ezekkel a jelölésekkel

$$\frac{\partial L(\underline{t})}{\partial t_{ij}} = d^1(i, j) \cdot e^{t_{ij}}$$

és

$$\frac{\partial^2 L(\underline{t})}{\partial t_{ij} \partial t_{kl}} = \begin{cases} d^2(i, j, k, l) \cdot e^{t_{ij} + t_{kl}} & \text{ha } i \neq k \text{ vagy } j \neq l \\ d^2(i, j, i, j) \cdot e^{2t_{ij}} + d^1(i, j) \cdot e^{t_{ij}} & \text{ha } i = k \text{ és } j = l \end{cases}$$

Így, mivel d^1 és d^2 Harker képletei alapján számolhatóak, meg tudjuk határozni mind a gradiens vektor, mind a Hesse-mátrix összes elemét. Ezután már felírhatjuk a \underline{t} vektorra a többváltozós Newton-iterációt:

$$\underline{t}_{n+1} = \underline{t}_n - [HL(\underline{t}_n)]^{-1} \nabla L(\underline{t}_n).$$

A többváltozós Newton-módszerben még szoktak használni egy γ lépésköz faktort is:

$$\underline{t}_{n+1} = \underline{t}_n - \gamma [HL(\underline{t}_n)]^{-1} \nabla L(\underline{t}_n).$$

Ha megadjuk a \underline{t} vektor kezdőértékét, \underline{t}_0 -t, már számolható az iteráció. A megállási kritérium itt is ugyanaz, mint az egyváltozós esetben, azaz

$$\|\underline{x}_n - \underline{x}_{n-1}\| < T,$$

ahol

$$\underline{x}_n = (x_1, \dots, x_d) = (e^{t_1}, \dots, e^{t_d}).$$

Azért az \underline{x} -re fogalmazzuk meg a megállási kritériumot, mert mivel $x_i = e^{t_i}$, a \underline{t} -ben bekövetkező apró változás még könnyen okozhat nagy változást az \underline{x} -ben.

8. fejezet

Számítási eredmények

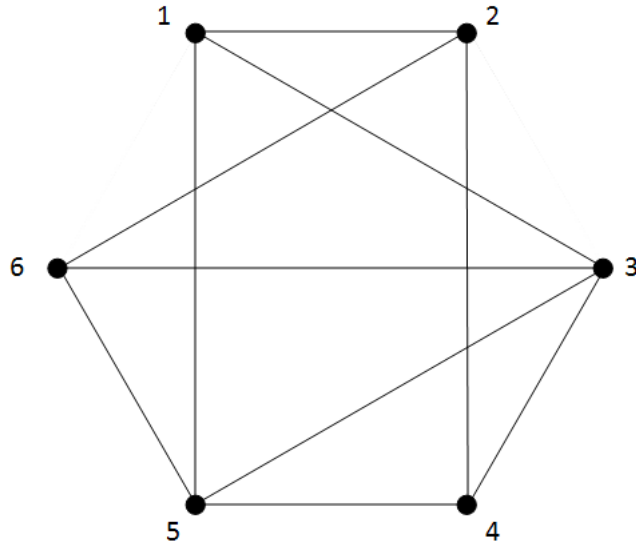
Rendelkezésünkre áll tehát az ismertett egyváltozós és többváltozós módszer. A következőkben egy példán mutatjuk be a két módszer működését, majd ismertetjük a nagy mintaelemszámú teszt eredményét. A módszereket összehasonlítjuk egymással, és egy harmadik – a Bozóki–Fülöp–Rónyai-cikkben bemutatott [2] – algoritmussal is, amely nagymértékben hasonlít az egyváltozósra, de nem a Newton-módszert használja.

8.1. Példák az algoritmusok működésére

Az egyváltozós és a többváltozós Newton-módszert használó algoritmusainkat először a következő példamátrixon fogjuk bemutatni:

$$A = \begin{pmatrix} 1 & 7 & 1/2 & * & 3 & * \\ 1/7 & 1 & * & 1/5 & * & 1/4 \\ 2 & * & 1 & 6 & 9 & 7 \\ * & 5 & 1/6 & 1 & 2 & * \\ 1/3 & * & 1/9 & 1/2 & 1 & 3 \\ * & 4 & 1/7 & * & 1/3 & 1 \end{pmatrix}$$

Nézzük az A mátrix gráf reprezentációját:



8.1. ábra. Az A mátrix gráfja

Látható, hogy ez összefüggő, vagyis a feladatnak van egyértelmű megoldása. Következő lépésként írjuk be a változókat a hiányzó elemek helyére. Oszlop-folytonosan indexeltük a változókat, hogy összhangban legyünk a programmal, mert ott technikai okokból így volt kézenfekvőbb.

$$A(\underline{x}) = \begin{pmatrix} 1 & 7 & 1/2 & x_2 & 3 & x_4 \\ 1/7 & 1 & x_1 & 1/5 & x_3 & 1/4 \\ 2 & 1/x_1 & 1 & 6 & 9 & 7 \\ 1/x_2 & 5 & 1/6 & 1 & 2 & x_5 \\ 1/3 & 1/x_3 & 1/9 & 1/2 & 1 & 3 \\ 1/x_4 & 4 & 1/7 & 1/x_5 & 1/3 & 1 \end{pmatrix}$$

Itt tehát a dimenzió $m = 6$, a hiányzó elemek száma $d = 5$. A többváltozós módszerben $\gamma = 1$ lesz. A megállási kritériumot négy tizedesjegy pontosságban határozzuk meg, azaz $T = 10^{-4}$. A kezdőértékek 1-re vannak beállítva, azaz $x_i^{(0)} = 1 \quad \forall i = 1, \dots, 5$.

A két módszer mellett még bemutatjuk a Bozóki–Fülöp–Rónyai-cikkben [2] ismertetett módszert, ami lényegében ugyanaz, mint az egyváltozós cik-

lizált módszer, de a problémára adaptált Newton-iteráció helyett ez a Matlab beépített *fminbnd* függvényét használja, ami egy adott intervallumon egy egyváltozós folytonos valós függvény lokális minimumát keresi meg. A használt intervallum $t \in (-10, 10)$, azaz $x \in (e^{-10}, e^{10})$. A gyakorlatban előforduló mátrixok esetén x bőven benne van ebben az intervallumban. Mivel megmutattuk, hogy az átskálázás után a probléma szigorúan konvex, ezért (akárcsak a Newton-módszer esetén) a lokális minimum itt is globális lesz.

A következő táblázat az $x_i^{(k)}$ változók értékét mutatja mindhárom módszerre (azaz az x elemeinek értékeit minden iterációban), amíg azok le nem állnak. Az „f” jelöli az *fminbnd*-vel működő módszert, „e” az egyváltozós Newton-módszert, „t” a többváltozós Newton-módszert.

k	$x_1^{(k)}$			$x_2^{(k)}$			$x_3^{(k)}$			$x_4^{(k)}$			$x_5^{(k)}$		
	f	e	t	f	e	t	f	e	t	f	e	t	f	e	t
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0.0577	0.0577	0.3643	1.5213	1.5213	1.5718	0.2735	0.2735	0.5790	3.1383	3.1383	2.1141	2.1081	2.1081	1.5201
2	0.0398	0.0398	0.3643	1.7432	1.7432	1.5718	0.2596	0.2596	0.5790	3.8376	3.8376	2.1141	2.1367	2.1367	1.5201
3	0.0393	0.0393	0.0736	1.8267	1.8267	1.8880	0.2593	0.2593	0.3400	3.8911	3.8911	3.7697	2.1199	2.1199	2.0629
4	0.0393	0.0393	0.0483	1.8365	1.8365	1.9185	0.2593	0.2593	0.2851	3.8894	3.8894	4.0359	2.1171	2.1171	2.1620
5	0.0393	0.0393	0.0397	1.8369	1.8369	1.8861	0.2593	0.2593	0.2616	3.8885	3.8885	4.0142	2.1169	2.1169	2.1606
6	0.0393	0.0393	0.0377	1.8369	1.8369	1.8481	0.2593	0.2593	0.2566	3.8884	3.8884	3.9223	2.1169	2.1169	2.1300
7	0.0393	0.0393	0.0385	1.8369	1.8369	1.8310	0.2593	0.2593	0.2579	3.8884	3.8884	3.8746	2.1169	2.1169	2.1127
8	-	-	0.0393	-	-	1.8316	-	-	0.2594	-	-	3.8743	-	-	2.1119
9	-	-	0.0395	-	-	1.8359	-	-	0.2597	-	-	3.8855	-	-	2.1157
10	-	-	0.0394	-	-	1.8377	-	-	0.2595	-	-	3.8903	-	-	2.1175
11	-	-	0.0393	-	-	1.8374	-	-	0.2593	-	-	3.8899	-	-	2.1174
12	-	-	0.0393	-	-	1.8369	-	-	0.2593	-	-	3.8886	-	-	2.1170
13	-	-	0.0393	-	-	1.8368	-	-	0.2593	-	-	3.8881	-	-	2.1168
14	-	-	0.0393	-	-	1.8368	-	-	0.2593	-	-	3.8882	-	-	2.1168

Az eredmények, amiket kaptunk (a negyedik tizedesjegytől eltekintve, ami kerekítési hibának tudható be), azonosak.

A mátrix optimális kitöltése

$$\underline{x} = (0.0393, 1.8369, 0.2593, 3.8884, 2.1169).$$

Az optimális sajátérték $\lambda_{max}(\underline{x}) = 6.2220$, így az optimálisan kitöltött mátrix inkonzisztenciája

$$CI = \frac{CI}{ACI} = \frac{\lambda_{max}(\underline{x}) - m}{m - 1} = \frac{\lambda_{max}(\underline{x}) - 6}{6 - 1} = \frac{6.222 - 6}{5} = 0.0354.$$

Tehát $CI < 0.1$, azaz az optimálisan kitöltött mátrix elfogadható inkonzisztenciát hordoz.

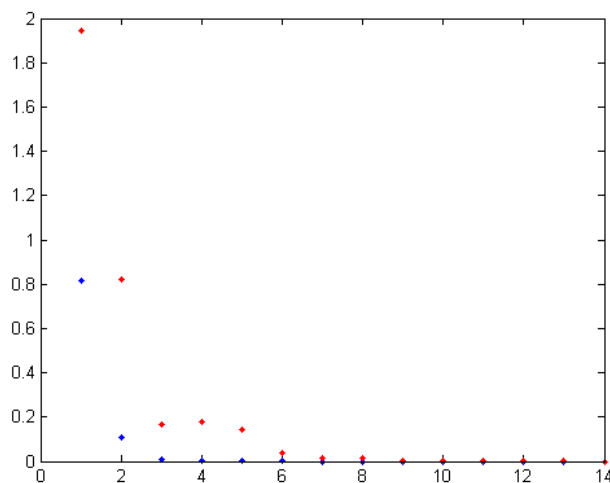
A $\lambda_{max}(\underline{x})$ -hez tartozó normált sajátvektor (azaz a súlyvektor):

$$w = (0.2058, 0.0206, 0.5239, 0.1119, 0.0822, 0.556).$$

A kitöltött mátrix így néz ki:

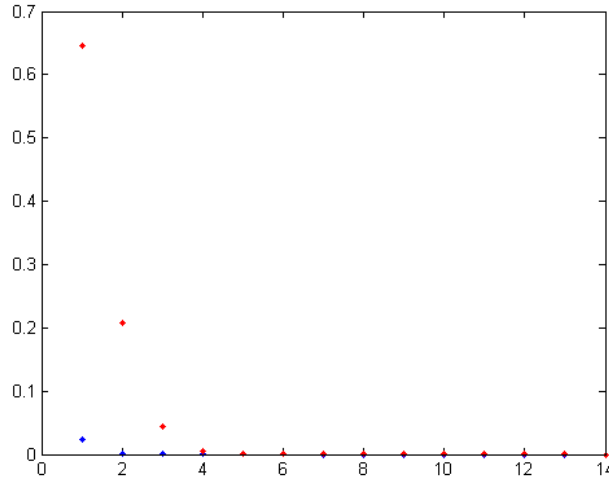
$$A(\underline{x}) = \begin{pmatrix} 1 & 7 & 1/2 & \mathbf{1.8369} & 3 & \mathbf{3.8884} \\ 1/7 & 1 & \mathbf{0.0393} & 1/5 & \mathbf{0.2593} & 1/4 \\ 2 & \mathbf{25.4286} & 1 & 6 & 9 & 7 \\ \mathbf{0.5444} & 5 & 1/6 & 1 & 2 & \mathbf{2.1169} \\ 1/3 & \mathbf{3.8559} & 1/9 & 1/2 & 1 & 3 \\ \mathbf{0.2572} & 4 & 1/7 & \mathbf{0.4724} & 1/3 & 1 \end{pmatrix}$$

Jelöljük \underline{x}^* -al a kapott optimális kitöltést, $\underline{x}^{(k)}$ -val a k -ik iterációban kapott \underline{x} vektort. Az alábbi 8.2 ábra mutatja az $\|\underline{x}^* - \underline{x}^{(k)}\|$ változását az egyes iterációkban. A kék pontok jelölik az egyváltozós Newton-módszerrel (és az *fminbnd*-vel) számolt értékeket, a pirosak a többváltozós Newtonnal számoltakat.



8.2. ábra. Az $\|\underline{x}^* - \underline{x}^{(k)}\|$ változása az A mátrixnál

Hasonlóan, jelöljük λ_{max}^* -al az algoritmus végén kapott optimális Perron-sajátértéket, $\lambda_{max}^{(k)}$ -val pedig a k -ik iterációban kapott mátrix legnagyobb sajátértékét. A kettő távolságát, $\lambda_{max}^{(k)} - \lambda_{max}^*$ -t az alábbi 8.3 ábrán követhetjük nyomon.



8.3. ábra. $\lambda_{max}^{(k)} - \lambda_{max}^*$ változása az A mátrixnál

Látható, hogy a két egyváltozós módszer – akár a Newton-módszert akár az *fminbnd*-t használjuk – nagyon hasonlóan viselkedik, olyannyira, hogy ebben a példában minden lépésben megegyeznek. Nincs feltétlenül mindig teljes egyezés lépésenként, de a két egyváltozós módszer valóban szinte egyformán viselkedik. Mindkét egyváltozós módszer leállt a 7. iteráció után.

A többváltozós módszer csak 14 iteráció után állt le. Nem jellemző tulajdonsága, hogy lassabb az egyváltozósnál, de előfordul.

Nézzünk egy másik példát: most a többváltozós módszer lesz a gyorsabb.

$$B(\underline{x}) = \begin{pmatrix} 1 & 5 & 3 & 7 & 6 & 6 \\ 1/5 & 1 & x_1 & 5 & x_3 & 3 \\ 1/3 & 1/x_1 & 1 & x_2 & 3 & x_5 \\ 1/7 & 1/5 & 1/x_2 & 1 & x_4 & 1/4 \\ 1/6 & 1/x_3 & 1/3 & 1/x_4 & 1 & x_6 \\ 1/6 & 1/3 & 1/x_5 & 4 & 1/x_6 & 1 \end{pmatrix}$$

A dimenzió, azaz a szempontok száma itt is $m = 6$, a hiányzó elemek száma $d = 6$. Itt is $\gamma = 1$ -et használunk a többváltozós módszerben, és a megállási kritérium is $T = 10^{-4}$, valamint a kezdőértékek is 1-re vannak beállítva, azaz $x_i^{(0)} = 1 \quad \forall i = 1, \dots, 6$.

k	$x_1^{(k)}$			$x_2^{(k)}$			$x_3^{(k)}$			$x_4^{(k)}$			$x_5^{(k)}$			$x_6^{(k)}$		
	f	e	t	f	e	t	f	e	t	f	e	t	f	e	t	f	e	t
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1.3528	1.3528	1.0473	3.21143	2.2115	2.6722	2.8944	2.8944	1.8645	0.58204	0.58204	0.64975	1.7839	1.7839	1.6844	0.7155	0.7155	0.84933
2	1.1031	1.1031	0.951684	4.40684	4.40684	3.8422	2.6095	2.6095	2.3292	0.57295	0.57296	0.55375	1.886	1.886	2.0116	0.74887	0.74887	0.81044
3	0.98043	0.98044	0.912024	4.62284	4.62284	4.8995	2.5021	2.5022	2.3709	0.55416	0.55417	0.53021	1.9873	1.9873	2.0941	0.77272	0.77271	0.80553
4	0.94687	0.94688	0.908424	4.76184	4.76184	4.926	2.4419	2.4419	2.3678	0.54268	0.54268	0.52932	2.0346	2.0346	2.0922	0.78596	0.78596	0.80246
5	0.92907	0.92907	0.908334	4.83624	4.83624	4.9261	2.4081	2.4081	2.3681	0.53654	0.53654	0.52925	2.0604	2.0604	2.0919	0.79337	0.79336	0.80235
6	0.91956	0.91956	0.908314	4.8769	4.877	4.9262	2.3898	2.3898	2.3681	0.53321	0.53321	0.52924	2.0747	2.0747	2.0918	0.79744	0.79744	0.80235
7	0.91443	0.91443	-	4.89924	4.8993	-	2.3799	2.3799	-	0.5314	0.5314	-	2.0825	2.0825	-	0.79967	0.79967	-
8	0.91164	0.91164	-	4.91154	4.9115	-	2.3745	2.3746	-	0.53042	0.53042	-	2.0867	2.0867	-	0.80088	0.80089	-
9	0.91013	0.91013	-	4.91814	4.9182	-	2.3716	2.3716	-	0.52989	0.52989	-	2.089	2.0891	-	0.80155	0.80155	-
10	0.90931	0.9093	-	4.92174	4.9218	-	2.37	2.37	-	0.52959	0.52959	-	2.0903	2.0903	-	0.80191	0.80191	-
11	0.90886	0.90885	-	4.92374	4.9238	-	2.3691	2.3692	-	0.52944	0.52944	-	2.091	2.091	-	0.80211	0.80211	-
12	0.90861	0.90861	-	4.92484	4.9249	-	2.3687	2.3687	-	0.52935	0.52935	-	2.0914	2.0914	-	0.80222	0.80222	-
13	0.90848	0.90847	-	4.92544	4.9255	-	2.3684	2.3684	-	0.5293	0.5293	-	2.0916	2.0916	-	0.80228	0.80228	-
14	0.9084	0.9084	-	4.92574	4.9258	-	2.3683	2.3683	-	0.52928	0.52928	-	2.0917	2.0917	-	0.80231	0.80231	-
15	0.90836	0.90836	-	4.9259	4.926	-	2.3682	2.3682	-	0.52926	0.52926	-	2.0918	2.0918	-	0.80233	0.80233	-
16	0.90834	0.90834	-	4.926	4.9261	-	2.3682	2.3682	-	0.52925	0.52925	-	2.0918	2.0918	-	0.80234	0.80234	-

A kapott eredmények itt is azonosak, kivéve az utolsó tizedesjegyet, a kapott optimális sajátértékek pedig teljesen azonosak, az optimális kitöltéshez tartozó Perron-sajátérték $\lambda_{max}(\underline{x}) = 6.2152$.

Az optimális kitöltés

$$\underline{x} = (0.9083, 4.9261, 2.3682, 0.5293, 2.0918, 0.8023).$$

Az inkonzisztencia $CI = 0.0343$, azaz ez is elfogadható inkonzisztenciájú.

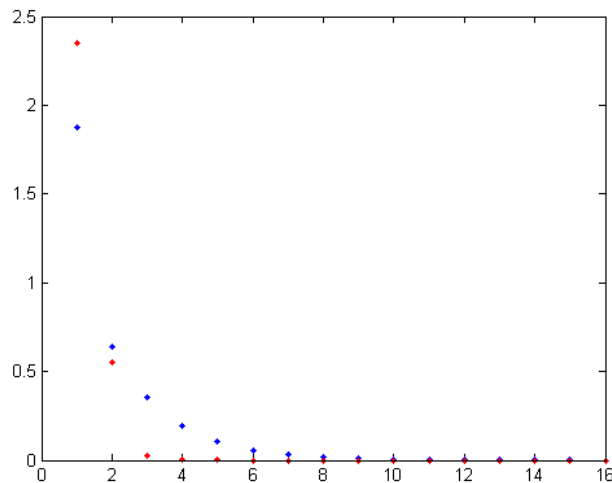
A kapott súlyvektor

$$w = (0.4778, 0.1625, 0.1717, 0.0368, 0.0659, 0.0853).$$

A kitöltött mátrix:

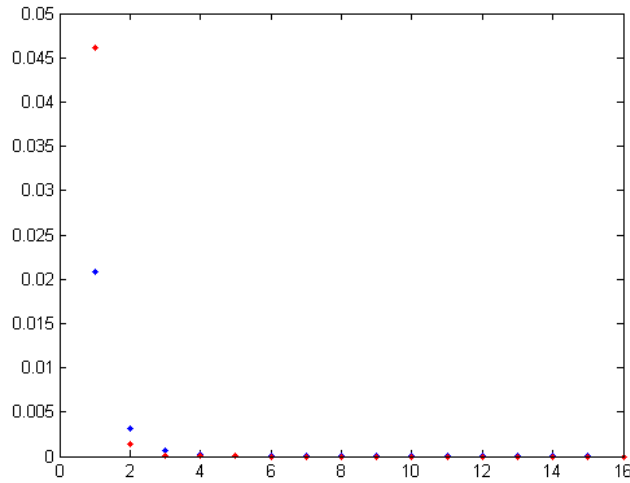
$$B(\underline{x}) = \begin{pmatrix} 1 & 5 & 3 & 7 & 6 & 6 \\ 1/5 & 1 & \mathbf{0.9083} & 5 & \mathbf{2.3682} & 3 \\ 1/3 & \mathbf{1.1009} & 1 & \mathbf{4.9261} & 3 & \mathbf{2.0918} \\ 1/7 & 1/5 & \mathbf{0.2030} & 1 & \mathbf{0.5293} & 1/4 \\ 1/6 & \mathbf{0.4223} & 1/3 & \mathbf{1.8895} & 1 & \mathbf{0.8023} \\ 1/6 & 1/3 & \mathbf{0.4781} & 4 & \mathbf{1.2464} & 1 \end{pmatrix}$$

Az előző mátrixnál alkalmazott jelölésekkel, az \underline{x} vektor, valamint a λ_{max} távolságát az optimumtól a következő (8.4 és 8.5) ábrákon követhetjük nyomon. Itt is a kék pontok jelölik az egyváltozós, pirosak a többváltozós módszerhez tartozó értékeket.



8.4. ábra. Az $\|\underline{x}^* - \underline{x}^{(k)}\|$ változása a B mátrixnál

A két egyváltozós módszerre a tapasztalat ugyanaz; a két módszer nagyon hasonlóan viselkedik. Itt a többváltozós módszer jóval gyorsabb volt, de mint az első példán láttuk, ez nincs mindig így. Sőt, a többváltozós módszerben még a γ választása is befolyásoló tényező.



8.5. ábra. $\lambda_{max}^{(k)} - \lambda_{max}^*$ változása a B mátrixnál

A konkrét példák szemügyre vétele után nézzük, mi az általános tapasztalat.

8.2. Véletlengenerált tesztek

A véletlen páros összehasonlítás mátrixok generálása úgy történik, hogy a felső háromszög minden pozíciójára az $1/9, 1/8, \dots, 1/2, 1, 2, 3, \dots, 9$ halmazból egyenletes eloszlás szerint választunk egy értéket, és az átellenes pozícióba beírjuk a reciprokát (a főátlót természetesen 1-esekkel töltjük ki).

Nem teljesen kitöltött páros összehasonlítás mátrixot többféleképpen lehet generálni. A nehézség az, hogy olyannak kell lennie, hogy a gráfja összefüggő legyen. Mivel egy kitöltött páros összehasonlítás mátrix gráfja teljes gráf, ezért egy pont foka $m - 1$. Ha tehát egy véletlen páros összehasonlítás mátrixból kitörlünk legfeljebb $m - 2$ elemet, akkor biztosan olyan nem teljesen kitöltött mátrixot kapunk, aminek a gráfja összefüggő. Mi a tesztlemegek során ezt a módszert alkalmazzuk.

Egy másik megközelítés, hogy ha egy üres gráfba húzunk be éleket addig, amíg az összefüggő nem lesz. Ennek a legegyszerűbb módja a csillag, azaz, ha kiválasztunk egy pontot, és az összes többi ponthoz húzunk onnan

egy élt. Ez a mátrix esetén úgy néz ki, hogy választunk véletlenszerűen egy számot $1, \dots, m$ -ből, és az annyiadik sort és oszlopot kitöltjük az előbbi módszer szerint választott véletlen számokkal. Ekkor azonban a mátrix kitölthető konzisztensen, így ez nem túl érdekes eset.

Sok más módon is lehet generálni véletlen, nem teljesen kitöltött páros összehasonlítás mátrixokat; mi az előbb ismertetett módon – azaz egy véletlen kitöltött páros összehasonlítás mátrixból legfeljebb $m - 2$ elemet törölve – hozzuk őket létre. Ezt úgy érjük el, hogy $m - 2$ -szer választunk két véletlen számot, i -t és j -t $1, \dots, m$ -ből, és az (i, j) és a (j, i) pozícióban lévő elemeket töröljük. Ez alól kivétel, ha $i = j$, vagy ha az adott koordinátájú elem már törölve lett. Így a törölt elemek száma legfeljebb $m - 2$, de lehet annál kevesebb is, és a pozíciójuk véletlen. A törlést úgy valósítjuk meg, hogy az adott elemet (és a reciprokát, ami átellenben van) egyszerűen 0-val helyettesítjük, hiszen egy páros összehasonlítás mátrixban nem fordulhat elő nulla, így ezzel egyértelműen jelölhetjük a hiányzó elemeket.

A tesztekben minden alkalommal négy tizedesjegy pontosság volt a megálási kritérium, azaz $T = 10^{-4}$, a kezdőértékek pedig mindhárom módszernél mindig $x_i^{(0)} = 1 \quad \forall i = 1, \dots, d$. Mint említettük, $d \leq m - 2$, bár a törlendő elemek helyének meghatározásából adódóan d tipikusan közel van $m - 2$ -höz, főleg ha m nagy. Ezért a hiányzó elemek számának átlagát, \bar{d} -t is mérjük. Minden tesztet 1000 darab véletlen mátrixra futtatunk le.

A módszereinket pontosság és iterációs szám alapján hasonlítottuk össze páronként. A táblázatban „i(f)”, „i(e)” és „i(t)” jelöli rendre az *fminbnd*, az egyváltozós Newton és a többváltozós Newton-módszer segítségével történő algoritmusok iterációs számát, „se(f)”, „se(e)” és „se(t)” pedig hasonlóan az optimális sajátértékeket. Mindegyik értékből a kisebb a jobb. A mérések során csak ezen értékek egymáshoz való viszonyát vizsgáltuk, nem azok konkrét értékeit. A táblázatok celláiban az adott méretű véletlen mátrixokon való 1000 darab futtatásból az adott oszlopban szereplő feltételnek megfelelő futások darabszáma látható, kivéve az első három oszlopot: az m a dimenzió, γ a többváltozós Newton-módszer lépésköze, \bar{d} pedig a hiányzó elemek átlagos száma.

Egyváltozós Newton vs. Többváltozós Newton

m	γ	\bar{d}	$se(f)=se(e)=se(t)$	$se(t)=se(e)$	$se(t)>se(e)$	$se(t)<se(e)$	$i(t)>i(e)$	$i(t)=i(e)$	$i(t)<i(e)$
6	0.1	3.07	652	652	348	0	907	27	66
7	1	3.949	982	982	18	0	346	351	303
8	1	4.844	955	955	45	0	279	379	342
9	1	5.796	918	918	82	0	244	359	397
10	1	6.704	921	921	79	0	192	428	380
15	1	11.529	993	993	7	0	79	634	287
20	1	16.378	998	998	2	0	32	789	179

 $fminbnd$ vs. Egyváltozós Newton

m	γ	\bar{d}	$se(f)=se(e)=se(t)$	$se(f)=se(e)$	$se(f)>se(e)$	$se(f)<se(e)$	$i(f)>i(e)$	$i(f)=i(e)$	$i(f)<i(e)$
6	0.1	3.07	652	1000	0	0	0	999	1
7	1	3.949	982	1000	0	0	0	1000	0
8	1	4.844	955	1000	0	0	0	1000	0
9	1	5.796	918	1000	0	0	0	994	6
10	1	6.704	921	1000	0	0	0	994	6
15	1	11.529	993	1000	0	0	0	1000	0
20	1	16.378	998	1000	0	0	0	1000	0

 $fminbnd$ vs. Többváltozós Newton

m	γ	\bar{d}	$se(f)=se(e)=se(t)$	$se(t)=se(f)$	$se(t)>se(f)$	$se(t)<se(f)$	$i(t)>i(f)$	$i(t)=i(f)$	$i(t)<i(f)$
6	0.1	3.07	652	652	348	0	907	27	66
7	1	3.949	982	982	18	0	346	351	303
8	1	4.844	955	955	45	0	279	379	342
9	1	5.796	918	918	82	0	246	359	395
10	1	6.704	921	921	79	0	192	429	379
15	1	11.529	993	993	7	0	79	634	287
20	1	16.378	998	998	2	0	32	789	179

A γ választásának stabilitási okai vannak, erre még külön visszatérünk később.

Látható a második táblázatból, hogy a két egyváltozós módszer gyakorlatilag tökéletesen azonosan működik, ebből kifolyólag az első és a harmadik táblázat szinte teljesen ugyanaz. A továbbiakban nem is foglalkozunk külön a két egyváltozós módszerrel, hanem egyként kezeljük őket. Nézzük tehát a megfigyeléseket, amiket az egyváltozós és a többváltozós módszer összehasonlításából, azaz az első (vagy a harmadik) táblázatból olvashatunk ki:

1. **Optimalitás:** Az esetek nagy többségében a két módszer által adott optimális sajátértékek megegyeznek, de amikor mégis eltérés van köztük, akkor mindig az egyváltozós a jobb. Az egyezések száma – úgy tűnik – m növekedésével növekszik.
2. **Iterációs szám:** A két módszer iterációs számának egymáshoz való viszonya nagy változatosságot mutat. A többváltozós gyakrabban gyorsabb

az egyváltozósnál, mint fordítva, de m növekedésével az egyezések száma válik dominánssá. Ha γ -t változtatjuk, az lényegesen befolyásolhatja a többváltozós Newton iterációs számát, erről a következő szakaszban lesz szó.

3. **Dimenzió:** Ha m növekszik, a két módszer határozottan egyre hasonlóbban viselkedik.

Ha egyelőre ragaszkodunk a $\gamma = 1$ választáshoz, akkor úgy tűnik, célravezetőbb az egyváltozós módszert használni, hiszen az sosem adott rosszabb eredményt. Ezt bizonyos mértékig árnyalja, hogy a többváltozós várhatóan valamivel kevesebb iterációval végez, ám ez csak egy várható lépésszám, nem egy szigorú becslés, hiszen néhányszor még lassabb is. Ha m nagy, akkor egyre kevésbé számít, hogy melyik módszert választjuk. Elképzelhető, hogy bizonyos m -ekre, ahol már a kétféle eredmény egyezése gyakorlatilag biztos, viszont az iterációs szám még kellő arányban különbözik, megéri többváltozós módszert alkalmazni; ez jövőbeni munkák témája lehet.

Az biztos, hogy az egyváltozós módszer megbízható, jó eredményeket produkál, ezért kiválóan alkalmas az adott probléma megoldására.

8.3. A többváltozós módszer stabilitása és a γ szerepe

A többváltozós módszer – a tapasztalatok alapján – néha hajlamos a divergenciára: a mátrixban végtelenbe tartó nagyságrendű elemek jelennek meg. Teljesen pontosan egyelőre nem sikerült karakterizálni, hogy mikor jöhet elő ilyen divergencia, de tapasztalati úton úgy tűnik, hogy az esélye a hiányzó és a kitöltött elemek arányától függ. Nézzünk egy példát (a példamátrix a Bozóki–Fülöp–Rónyai-cikkből származik [2]):

$$\begin{pmatrix} 1 & 5 & 3 & 7 & 6 & 6 & 1/3 & 1/4 \\ 1/5 & 1 & x_1 & 5 & x_2 & 3 & x_3 & 1/7 \\ 1/3 & 1/x_1 & 1 & x_4 & 3 & x_5 & 6 & x_6 \\ 1/7 & 1/5 & 1/x_4 & 1 & x_7 & 1/4 & x_8 & 1/8 \\ 1/6 & 1/x_2 & 1/3 & 1/x_7 & 1 & x_9 & 1/5 & x_{10} \\ 1/6 & 1/3 & 1/x_5 & 4 & 1/x_9 & 1 & x_{11} & 1/6 \\ 3 & 1/x_3 & 1/6 & 1/x_8 & 5 & 1/x_{11} & 1 & x_{12} \\ 4 & 7 & 1/x_6 & 8 & 1/x_{10} & 6 & 1/x_{12} & 1 \end{pmatrix}$$

Erre a mátrixra a többváltozós módszer divergál. Látható, hogy itt a mi tesztjeinknél arányaiban több hiányzó elem szerepel, $d = 12$, és $m = 8$. Ha azonban két hiányzó elem helyére beírjuk az arra az elemre (az egyváltozós módszerrel számolt) optimumot, akkor már működik a többváltozós módszer.

Felmerül, hogy esetleg a kezdőértékek jobb megválasztásával rendbehozható a többváltozós módszer. Azonban ez nem így van: ha az előbbi mátrixban a két elemet – ahelyett, hogy kitöltenénk – meghagyjuk változónak, de az optimumról indítjuk őket, akkor is divergál a többváltozós módszer.

A nagy elemszámú tesztekben kis m -ekre az derült ki, hogy m minél kisebb, annál nagyobb az esélye a divergenciának. Ez főleg $m = 4, 5, 6$ -ra fordul elő, $m = 7$ -re elég ritkán, $m = 8$ -ra pedig már egyáltalán nem fordult elő. Az előző példa viszont azt mutatja, hogy $m = 8$ -ra is elő tud állni ilyen helyzet, azonban d csökkentésével helyrehozható. A mi tesztjeinkben mindig $d \leq m - 2$. Ezért természetesen adódik a hipotézis, hogy a divergencia esélye a kitöltetlen elemek arányától függ. Ez már csak azért is összhangban van a tapasztalattal, mert

$$\frac{m-2}{\frac{m(m-1)}{2}} = 2 \cdot \frac{m-2}{m^2-m} \xrightarrow{m \rightarrow \infty} 0,$$

azaz ha $m \rightarrow \infty$ akkor a tesztekben kitöltetlen elemek aránya és ezzel hipotézisünk szerint a divergencia esélye tart a 0-hoz.

A példában szereplő 12 változós 8×8 -as mátrixra is tudjuk azonban sikerrel alkalmazni a többváltozós Newton-módszert, ennek kulcsa pedig a γ lépésköz faktor módosítása. Ha a példamátrixnál $\gamma = 0.6$, vagy kisebb, akkor már nem divergál rá a Newton-módszer, míg még például $\gamma = 0.7$ -re

divergál. A tapasztalat szerint ha γ -t csökkentjük, azzal stabilitást nyerünk az iterációs szám rovására. Probléma viszont, hogy az alkalmas γ mátrixfüggő. Elképzelhető, hogy lehet találni minden m -re és d -re olyan γ -t, hogy arra már nem divergál a többváltozós Newton-módszer, viszont az is lehet, hogy túlságosan függ a használható γ a konkrét mátrixtól ahhoz, hogy ilyet meg lehessen adni (az 1000 darabos tesztben $\gamma = 0.1$ -et választottunk, erre már nem divergált egy sem közülük).

Az is lehetséges, hogy nagy m -ekre, vagy esetleg kis d/m arányra, ahol a stabilitás már nem probléma, érdemes $\gamma > 1$ -et választani, hogy a sebességet növeljük, úgy, hogy a stabilitást is megtartsuk. Ezek a kérdések későbbi kutatások tárgyát képezhetik.

A konkrét program numerikus módosításaival is lehetne próbálkozni, hátha stabilabb eljárást tudunk nyerni.

9. fejezet

Konklúzió

A dolgozatban megnéztük, hogyan lehet egy többszemponútú döntési feladatból páros összehasonlítás mátrix segítségével a döntéshozó szubjektív preferenciáinak megfelelő optimális döntést meghozni. Ezután definiáltuk a nem teljesen kitöltött páros összehasonlítás mátrixokat, amik a hiányzó információjú döntési feladatok egy fajtáját reprezentálják, nevezetesen azt, ha nincs minden szempont összehasonlítva. Definiáltuk a nem teljesen kitöltött páros összehasonlítás mátrixok optimális kitöltését, ami azt a kitöltést jelentette, amire az inkonzisztencia, ekvivalensen a mátrix legnagyobb sajátértéke minimális. Fő feladatunknak ezért a λ_{max} minimalizálását tekintettük.

Megnéztük, hogyan lehet konvex optimalizálási feladattá átparaméterezni az eredeti feladatot, és azt is, hogy a feladatnak milyen körülmények között létezik egyértelmű megoldása. Az így tisztázott feladatra Harker képletei segítségével egy új módszert adtunk, egy Newton-módszert alkalmaztunk az átparaméterezett problémára, mind egy-, mind többváltozós formában.

Végül bemutattuk az új módszerek gyakorlati működését, összehasonlítottuk őket egymással, és egy már máshol [2] alkalmazott módszerrel is, és néhány új irányt adtunk jövőbeni lehetséges vizsgálatok számára. Az eredmények biztatóak: mindkét módszer működőképes, és (főleg az egyváltozós módszer esetén) nem rosszabb, mintha a már alkalmazott módszert használnánk.

Irodalomjegyzék

- [1] B. Aupetit and C. Genest, *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(1993), 263–268.
- [2] S. Bozóki, J. Fülöp, L. Rónyai, *On optimal completions of incomplete pairwise comparison matrices*. Mathematical and Computer Modelling 52(2010), 318–333.
- [3] P.T. Harker, *Alternative modes of questioning in the analytic hierarchy process*. Mathematical Modelling 9(3)(1987), 353–360.
- [4] P.T. Harker, *Incomplete pairwise comparisons in the analytic hierarchy process*. Mathematical Modelling 9(11)(1987), 837–848.
- [5] P.T. Harker, *Derivatives of the Perron root of a positive reciprocal matrix: with application to the Analytic Hierarchy Process*. Applied Mathematics and Computation 22(1987), 217–232.
- [6] J.F.C. Kingman, *A convexity property of positive matrices*. The Quarterly Journal of Mathematics 12(1961), 283–284.
- [7] D.G. Luenberger and Y. Ye, *Linear and Nonlinear Programming*. Series: International Series in Operations Research & Management Science, vol. 116. 3rd Edition (Springer, 2008).
- [8] T.L. Saaty, *A scaling method for priorities in hierarchical structures*. Journal of Mathematical Psychology 15(1977), 234–281.

- [9] T.L. Saaty, *The analytic hierarchy process* (McGraw-Hill, New York, 1980).
- [10] S. Shiraishi, T. Obata and M. Daigo, *Properties of a positive reciprocal matrix and their application to AHP*. Journal of the Operations Research Society of Japan 41(3)(1998), 404–414.
- [11] S. Shiraishi and T. Obata, *On a maximization problem arising from a positive reciprocal matrix in AHP*. Bulletin of Informatics and Cybernetics 34(2)(2002), 91–96.
- [12] V.M.R. Tummala and H. Ling, *A note on the Computation of the Mean Random Consistency Index of the Analytic Hierarchy Process (AHP)*. Theory and Decision 44(1998), 221–230.