

**Hierarchikus pénzosztási rendszerek néhány
problémájának megoldása véletlen vektorok és a
multidimenzionális skálázás alkalmazásával**

Készítette:
Takács Szabolcs

A doktori iskola vezetője:
Dr. Laczkovich Miklós
A matematika tudományok doktora

A doktori program vezetője:
Dr. Michaletzky György
A matematika tudományok doktora

Témavezető:
Dr. Vizvári Béla
A matematika tudományok kandidátusa



*Eötvös Loránd Tudományegyetem
Természettudományi Kar
Matematika Doktori Iskola
Alkalmazott Matematika Doktori Program*

Budapest, 2013.

Tartalomjegyzék

1. Bevezető	9
2. Pszeudo-véletlen egész vektorok	10
2.1. Nem természettudományos megközelítések	10
2.2. A véletlen számokról, vektorokról általánosságban	11
2.2.1. A „véletlenszerűség” meghatározásainak lehetséges változatai	13
2.2.2. Matematikatörténeti kitekintés	14
2.2.3. Gyakorlati megközelítések különböző területeken	15
2.2.4. Matematikai területek megközelítései a tapasztalt jelenségre	16
2.2.5. Pszeudo-véletlen sorozatokat generáló algoritmusok	17
2.3. Az általunk alkalmazott tesztek ismertetése	19
2.3.1. Az egyenletesség tesztelése	20
2.3.2. Az összefüggőség tesztelése	21
2.4. A vizsgált algoritmus ismertetése	25
2.4.1. Az algoritmus eredeti változata	25
2.4.2. Tapasztalatok utáni javítások, fejlesztések	26
2.4.3. Módosítás az algoritmuson	27
2.5. Az algoritmussal elért eredményeink a különböző statisztikai teszteken	28
2.5.1. Kolmogorov–Szmirnov-statisztika (KS) alkalmazása	28
2.5.2. A χ^2 -teszt eredményei	30
2.5.3. A Kendall-féle teszt eredményei	30
2.5.4. Az átrendezésteszt és a sorozateszt eredményei	31
3. Multidimenziós skálázás (MDS)	33
3.1. A klasszikus MDS-elemzés elméleti háttere	33
3.1.1. A klasszikus MDS algoritmikus megoldása	35
3.1.2. A klasszikus megoldás optimalitási tulajdonsága	36
3.1.3. Shepard–Kruskal-algoritmus	37
3.2. Az MDS néhány alkalmazási területe	39
3.3. A többdimenziós skálázás más megközelítése	41
3.3.1. Optimalizálási feladatként való felírás	42
3.3.2. Tesztelési eredmények	47
3.3.3. Első tesztelési/kísérleti feladatok	47
3.3.4. Második tesztelési/kísérleti feladatok	54
3.4. Futásidő-javításra tett kísérlet – specifikálás	55
3.4.1. Az l_1 és l_∞ modellek specifikálása, linearizálása	55
3.5. Az MDS alkalmazása: ellenőrzés, visszakövetkeztetés	58
3.5.1. A cikk filozófiai célkitűzése	58
3.5.2. Az l_1 norma speciális esete	59
3.5.3. Visszakövetkeztetésben tett tapasztalati eredményeink	61
3.6. Az általánosított MDS-eljárás tesztelési eredményei	64

3.6.1.	Az elvégzett tesztek és eredményeik	65
3.6.2.	Az általánosított MDS-eljáráson elvégzett számítások tapasztalatai	67
3.7.	Az MDS-eljárás egy másik alkalmazása – munkaügyi kirendeltségek osztályozása	68
3.7.1.	Csak két koordináta használata – könnyebb interpretáció	71
3.8.	Konklúzió, célkitűzés	73
4.	Hierarchikus pénzosztási feladat – Alkalmazás	74
4.1.	Kérdésfelvetés	74
4.2.	Specifikumok, a közigazgatási környezet	75
4.3.	A matematikai modell alakja	79
4.3.1.	Korlátok	79
4.3.2.	Az optimalizálandó célfüggvény	79
4.3.3.	A feladat megfogalmazása	79
4.3.4.	Formális felírás	80
4.3.5.	Gazdasági interpretáció	84
4.4.	Strukturális tulajdonságok	85
4.4.1.	Robusztus optimalizálás	87
4.4.2.	A megoldandó feladat vázlatos szerkezete	88
4.4.3.	A véletlenítés gazdasági interpretációja	90
4.4.4.	A robusztus megoldás gazdasági interpretációja	91
4.5.	A lehetséges szimulációs pontok	94
4.6.	A különböző megoldások összehasonlítása	96
4.6.1.	A robusztus megoldás	96
4.6.2.	Kevert robusztus megoldás	97
4.6.3.	A feladat megvalósításának bemutatása	98
4.6.4.	Szimulációink eredményei	104
4.6.5.	A robusztus megoldással való összehasonlítás	109
4.7.	Felhasználói tapasztalatok, összegzés	109
5.	Összefoglalás	110
5.1.	Pseudo-véletlen generátorban elért eredmények	111
5.1.1.	Tesztadaptáció	111
5.1.2.	Teszteredmények	111
5.1.3.	Publikáció	112
5.2.	Az MDS általánosításában, vizsgálatában elért eredmények	112
5.2.1.	Az általánosítás és új területen való alkalmazás eredményei	112
5.2.2.	Interpretációs eredmények	113
5.2.3.	Publikációk	113
5.3.	Alkalmazás-fejlesztés	114
5.3.1.	A matematikai vizsgálat	114
5.3.2.	Az alkalmazás bemutatása, publikációk	116
5.4.	Összesítés	116
5.5.	Melléklet - NYILATKOZAT	117

6. Összefoglaló	122
7. Summary: Solving a problem of a hierarchical money-distribution system with pseudo-random integer vectors and multidimensional scaling	123

Ábrák jegyzéke

1.	MDS-teszteredmények, körvonalon véletlenszerű 10 pontra – MATLAB [©]	49
2.	MDS-teszteredmények, körvonalon véletlenszerű 15 pontra – MATLAB [©]	50
3.	MDS-teszteredmények, körvonalon véletlenszerű 20 pontra – MATLAB [©]	51
4.	Exponenciális illesztés a pontok száma alapján az algoritmus futásidejére – EXCEL [©]	53
5.	MDS-teszteredmények 9 rácspontra – EXCEL [©]	54
6.	MDS-teszteredmények 16 rácspontra – EXCEL [©]	54
7.	MDS-teszteredmények 25 rácspontra – EXCEL [©]	55
8.	A Had14 feladat megoldása l_1 normában – EXCEL [©]	61
9.	A Kra30a feladat megoldása 3 dimenzióban, l_1 normában – MATLAB [©]	62
10.	A Kra30a feladat megoldása 3 dimenzióban, l_2 normában – MATLAB [©]	62
11.	A Kra30a feladat megoldása 2 dimenzióban, l_1 normában – EXCEL [©]	62
12.	A Kra30a feladat megoldása 2 dimenzióban, l_2 normában – EXCEL [©]	63
13.	A Rou12 feladat megoldása l_1 normában – EXCEL [©]	63
14.	Az MDS alkalmazása munkaügyi adatokra – SPSS [©]	70
15.	Az MDS alkalmazása két koordinátára – SPSS [©]	72
16.	A vizsgált közigazgatási rendszer vázlatos szerkezete – POWERPOINT [©]	84
17.	Az alkalmazás nyitófelülete – EXCEL [©]	99
18.	Az alkalmazási felület véletlenítési lehetőségének beállítása – EXCEL [©]	99
19.	Létszámközepek és politikai korlátok – EXCEL [©]	100
20.	A megoldó beállításai, feltételek – EXCEL [©]	101
21.	A megoldó beállításai, a futtatás jellemzői – EXCEL [©]	102
22.	Az eredmények listázása kirendeltségenként, eszközönként – EXCEL [©]	103
23.	Az eredmények listázása: a célfüggvény jellemzői – EXCEL [©]	103
24.	A kirendeltségi szerepek ábrázolása az MDS-eljárás segítségével – SPSS [©]	105
25.	A kirendeltségi létszámeredmények szerinti osztályozás MDS-eljárás segítségével – SPSS [©]	107
26.	A szimulációs létszamarányok ábrázolása MDS-eljárás segítségével – SPSS [©]	108
27.	A Munkaügyi Központ nyilatkozata	117

Táblázatok jegyzéke

1.	A MATLAB generátorának eredményei a Kolmogorov–Szmirnov-teszten	29
2.	Saját algoritmusunk eredményei a Kolmogorov–Szmirnov-teszten	29
3.	A Kendall-teszt eredményei	30
4.	A sorozatteszt futásainak száma	31
5.	A sorozattesztben megfogott futások aránya	32
6.	Futásidők a különböző normákban, másodpercben	52
7.	Futásidők a különböző normákban, másodpercben	54
8.	Testfeladatok listája az általánosított MDS-eljáráshoz	65
9.	Az (l_1, l_∞) rekonstrukciós modell eredményei	65
10.	Az (l_1, l_1) rekonstrukciós modell eredményei	66
11.	Az (l_2, l_2) rekonstrukciós modell eredményei	67
12.	A mérlegelési szempontok értékeinek standardizált táblázata, kirendeltségenként	69
13.	A fővárosi munkaügyi kirendeltségek kódjai, megnevezései és illetékességei	76
14.	Az aktív eszközök ára, sugara a robusztus feladatban	96
15.	Vállalt létszámok kirendeltségenként	97
16.	Irányítandó létszámok és keretek a robusztus feladat megoldásában	97
17.	Irányítandó létszámok és szerepek a kevert robusztus megoldásban	98
18.	A kirendeltségi szerepek alakulása a szimulációs eljárásban	104
19.	A szimulációs létszámeredmények átlagai, kerekítve	106
20.	A szimulációs eredmények létszámarányai eszközönként, kirendeltségenként	108

Köszönetnyilvánítás

Köszönettel tartozom mindenekelőtt témavezetőmnek, Dr. Vizvári Bélának, amiért végtelen türelemmel és inspiráló ötleteivel, javaslataival ösztönzött – nem csak az e tézisben feldolgozott témákban. Közös munkánk során úgy az alkalmazott matematikában, mint az élet más területein is példát mutatott – amiért nem tudok elég hálás lenni.

Szintén köszönettel tartozom Dr. Vargha Andrásnak, a Károli Gáspár Református Egyetem Pszichológiai Intézete vezetőjének atyai és baráti tanácsaiért, figyelméért – mely lényegében doktori tanulmányaim kezdete óta kísér. Javaslatai, beszélgetéseink során felmerült ötletei nagyban segítettek, inspiráltak – és inspirálnak, jóval túlmutatva e dolgozat keretein.

Nem hagyható ki e felsorolásból a tézis témájaként is szolgáló projekt miatt a jelenleg Budapest Főváros Kormányhivatala Munkaügyi Központjának (korábban Fővárosi Munkaügyi Központnak) két vezetője sem, Mátrainé Bartus Éva és Vass István. Részint a téma felvetésében, részint a munkakörülményeim megteremtésében minden emberileg lehetséges segítséget megadtak és megadnak – így a szervezetet érő folyamatos változások közepette is nyugodt körülmények között dolgozhattam, dolgozom. Köszönöm továbbá e helyen közvetlen kollégáimnak, vezetőimnek is a folyamatos támogatást, segítséget és érdeklődést, amivel munkámat segítették.

Köszönöm továbbá Dr. Csizmadia Zsoltnak a szünni nem akaró lelkesedését és konstruktív, objektív meglátásait, mellyel a dolgozat elkészítése közben felmerült problémáimra reagált. Szintén ehelyütt tartozom köszönettel Fenyőné Piro Krisztának, aki nyelvhelyességi ellenőrzéssel, stilisztikai megjegyzésekkel tette könnyebben olvashatóvá a dolgozatot.

Hálás szívvel gondolok barátaimra, akik tanulmányaim alatt és azóta is érdeklődésükkel, támogatásukkal tartották bennem a lelket, illetve biztatásukkal segítettek át a nehezebb pillanataimon.

Természetesen – és egyáltalán nem utolsó sorban – köszönettel tartozom családomnak és legfőképpen feleségemnek, Juditnak. Megteremtette és fenntartotta a munkáimhoz elengedhetetlen nyugalmat, továbbá a fokozott intenzitású munkafolyamatok során nagymértékben tehermentesített.

1. Bevezető

A dolgozat egy olyan munkafolyamat ismertetése, mely egy véletlen vektorokat generáló algoritmus megszületésétől egészen annak felhasználásáig tart – kisebb kitérőket is beleértve, hogy az alkalmazás felhasználói oldala is teljesebbé válhasson. A köztes lépések leírásra kerülnek a „véletlen” definiálásától egészen az alkalmazásban bemutatott indokolt használatig.

A dolgozat során több helyen kisebb kitérőket teszünk alkalmazások, felhasznált rutinok irányába, melyek elméleti és gyakorlati szempontból lényegében tisztázottak és elfogadottak. A vizsgálatokat próbáltuk a lehető legegyszerűbb módokon megvalósítani, hogy a reprodukálhatóság lehetőségét mindinkább biztosítani tudjuk annak ellenére, hogy alapvetően a „véletlent” szándékoztuk vizsgálni.

A dolgozat három nagyobb blokkra tagolható. Az első blokk egy véletlen egész vektorokat generáló algoritmus köré szerveződik, annak vizsgálatáról szól. A vizsgált algoritmust kutatási riportban [51] és Ramirez könyvében [37] is megtalálhatjuk. Az algoritmus viselkedésének vizsgálatát elvégeztük, eredményeit cikkünk [49] mutatja be.

A második blokkban egy jól ismert statisztikai eljárást, a multidimenziós skálázást (MDS) vizsgáljuk. Az újdonság e blokk kapcsán az, hogy nem statisztikai, hanem optimalizálási feladatként írtuk fel, majd általánosítottuk az eljárást. Az általánosítást több módon is vizsgáltuk, eredményeink egy részét mutatja be cikkünk [31], illetve további tapasztalatainkról szól kutatási riportunk [32].

A harmadik blokk a dolgozat oksági alapját adó alkalmazást ismerteti, összekapcsolva a két első fejezetet. Ennek keretében egy munkafolyamatot, egy közigazgatási probléma egy lehetséges megoldását mutatjuk be, melyben a fenti két módszer hasznos eszköznek bizonyult. Eredményeinket a Munkaügyi Szemlében megjelent cikkünk [50] mutatja be.

A dolgozat első két blokkjában az alábbi tagolásokat követjük:

- Az első rész az adott fejezet matematikai hátterét mutatja be, amelyben (akár több különböző) forrásból általános, rövid ismertetéseket adunk a fejezet témájából – definíciókkal és eddig elért, a dolgozat szempontjából releváns eredményekkel.
- A második rész az adott fejezet tudományterületeken elért alkalmazásainak jegyzékébe ad betekintést: próbáltunk a lehetőségekhez képest széles körben keresni, felsorakoztatni tanulmányokat, könyveket vagy jegyzeteket – orvosi, biológiai, közgazdasági vagy akár humán tudományokból – annak érdekében, hogy látható legyen: a vizsgált módszerek általánosan elfogadottak, számos területen használtak.
- A harmadik rész minden fejezetben a vizsgált módszerekkel elért elméleti és/vagy gyakorlati eredményeinket, tapasztalatainkat ismerteti – részben a harmadik blokk alkalmazásához kapcsolódóan.

Az első két fejezet eredményei általánosan felhasználható eredmények, míg az eljárások, melyeket e két fejezetben ismertetünk, szervesen hozzájárulnak ahhoz, hogy az utolsóban ismertetendő alkalmazásunk eredményes legyen. A dolgozatban tehát lényegében egy munkafolyamat dokumentációját olvashatjuk: egy gyakorlati probléma kapcsán felmerült elméleti kérdésekre adott gyakorlati és elméleti válaszokat ismertetünk – mely elméleti eredmények önálló módon is megállják a helyüket, a harmadik fejezetben pedig ezek közvetlenül egy alkalmazásban is manifesztálódnak.

2. Pszeudo-véletlen egész vektorok

Arról, hogy mi a véletlen, vagy mit tekintünk véletlennek, érdemes talán több oldalról vizsgálatot indítani. Három eltérő nézőpontból ismertetem vázlatosan a lehetséges megfogalmazásokat, definíciókat – természetesen a teljesség igénye nélkül, mégis megpróbálva képet adni a helyzet bonyolultságáról.

A három nézőpont a nem természettudományos, az elméleti és a gyakorlati megfontolás lesz. Az első kettő nem feltétlenül különíthető el egymástól, mi most mégis megteesszük annak érdekében, hogy érzékeltessük: a téma csak részben természettudományos.

2.1. Nem természettudományos megközelítések

Erről a megközelítésről alapvetően azért érdemes megemlékezni pár mondatban, hogy látható legyen: a természettudományok számára is olykor nehezen kezelhető fogalom mennyire másként értelmezhető bizonyos nem természettudományos – vagy hagyományos értelemben akár nem is tudományos – megközelítésben.

A Pallas Nagy Lexikona definíciója szerint:

1. Definíció. *A véletlen ellentéte a szükségesnek, szándékosnak, lényegesnek. De minthogy ezen a világon minden szükségkép történik, mint okoknak okozata; minthogy voltaképp semmi se lényegtelen, amennyiben mindenesetre összefügg ezzel, amit lényegesnek tartunk: véletlen ebben az értelemben nem létezik.*

Innen is látszik, hogy a Pallas Nagy Lexikona – szemmel láthatóan matematikai precizitást nélkülöző – definíciója szerint véletlen nem létezik, minden okkal történik.

A fenti definíció első fele – nevezetesen, hogy a véletlen a szándékos ellentéte – matematikailag használhatatlan, hiszen matematikailag interpretálhatatlan a szándékosság.

Ettől nem sokban különbözik pl. az Új Magyar Lexikon definíciója – bár itt már némi kitekintésre is módunk lehet:

2. Definíció. *A véletlen a dialektika kategóriája a természet, a társadalom és a gondolkodás mozgásának, fejlődésének a szükségszerűvel ellentétes, de tőle elválaszthatatlan és azt kiegészítő objektív mozzanatának jelölésére. (...) A tudományos kutatás és elemzés számára azonban nélkülözhetetlen a szükségszerű és a véletlen dialektikus kapcsolatának ismerete és a folyamatokban lévő törvényszerűségek feltárása érdekében ez ismeret gyakorlati alkalmazása.*

Mindkét definíció második fele már természettudományosan is megfogható, így kiindulási alapként használható: „a világon minden szükségkép történik, mint okoknak okozata”. A definíció ezen része a természettudományok princípiumaként fogható fel: mindent modellezni szándékozunk, az általunk ismert és észlelhető világ jelenségeire matematikai modelleket próbálunk illeszteni, majd ezen illeszkedést igazolni szeretnénk részint elméleti, részint tapasztalati úton. Csak azok a modellek állják ki a természettudományok próbáját, melyek mind elméleti, mind gyakorlati oldalról bizonyítást nyertek.

Ilymódon a véletlenszerűség lehetséges definícióit mindig abban a közegben kell értelmeznünk, melyben használni szeretnénk: bonyolultság, előrejelezhetőség/kitalálhatóság, analitikus vagy számítógépes alkalmazások területe.

2.2. A véletlen számokról, vektorokról általánosságban

A most következő fejezet alapvetően Knuth munkája [19] alapján készült - így a teljesség igénye nélkül próbálunk egy általános képet mutatni. Érdekes felidézni Neumann János 1951-ben tett kijelentését, miszerint „bárki, aki számelméleti módszerekkel akar véletlen számjegyeket előállítani, egyfajta bünt követ el”.

Az egész elméleti megalapozást olyan struktúrában szeretnénk tárgyalni, hogy a gyakorlati megvalósítások, alkalmazások oldaláról lehessen vizsgálni a kérdéseket. Nagyon fontos kiemelni: nem véletlen, hanem „véletlenszerű” sorozatokat szeretnénk keresni, definiálni. Ennek tárgyalását két elméleti megfogalmazás vizsgálatával, javításával próbáljuk végrehajtani.

3. Definíció. (LEHMER, 1951) *A véletlen sorozat bizonytalan fogalmán olyan sorozatot értünk, melynek későbbi elemei nem megjósolhatók; jól vizsgáljuk „néhány” statisztikai próbán; a próbák megválasztása attól is függhet, hogy mire akarjuk a sorozatot felhasználni.*

4. Definíció. (FRANKLIN, 1962) *Egy számsorozat véletlen, ha rendelkezik minden olyan tulajdonsággal, mely egyetlen eloszlású, véletlen változókból álló független minták végtelen sorozatainak közös tulajdonsága.*

Vegyük észre, hogy Franklin definíciója szerint egy sorozatnak minden statisztikai próbán jól kell szerepelnie, tehát biztosan állíthatjuk, hogy ez alapján a definíció alapján véletlen sorozat nem létezik (nincsen olyan véletlen sorozat, mely minden statisztikai próbán minden körülmények között megfelelné).

Járható útnak tűnik, hogy Lehmer definícióját próbáljuk szigorítani – elérve ezzel egy vélhetően elég szigorú, de teljesíthető definícióig.

Ahhoz, hogy néhány, e fenti két definíción alapuló, valóban használható definícióját adjuk a véletlenszerű sorozatoknak, szükségünk van néhány alapvető definícióra és állításra. Ezeket pontokba szedjük – felvillantva az irányokat, és egyes esetekben a tulajdonságok szigorításának szükségességére példákat hozunk.

A véletlenszerűség megragadására hozott definíciókhoz szükségünk lesz az alábbi, még nem egészen a véletlenszerűséget tárgyaló segéddefinícióra, -fogalomra:

5. Definíció. *Legyen $S(n)$ egy állítás, mely $n \in \mathbb{N}$ és U_0, U_1, \dots sorozat elemeire vonatkozik. Azt mondjuk, hogy $P(S(n)) = \lambda$, ha*

$$\lim_{n \rightarrow \infty} \frac{\nu(n)}{n} = \lambda,$$

ahol $\nu(n)$ azon U_i -k száma, amelyekre $S(n)$ teljesül.

Más megfogalmazásban ez azt jelenti, hogy $S(n)$ igaz voltának valószínűsége akkor λ , ha a $\frac{\nu(n)}{n}$ relatív gyakoriság végtelenben vett határértéke éppen λ .

Tekintsünk egy példát annak érdekében, hogy az egyetlen eloszlású sorozatok esetén az egyenletesség még nem elegendő feltétele a véletlenszerűségnek.

1. Példa. Legyen U_0, U_1, \dots véletlenszerű $[0, 1)$ -en, legyen V_0, V_1, \dots szintén véletlenszerű $[0, 1)$ -en, továbbá legyen W_0, W_1, \dots is véletlenszerű $[0, 1)$ -en. Ekkor tudható, hogy

$$\begin{aligned} U_i^* &\sim \frac{1}{3}U_0, \frac{1}{3}U_1, \dots, \\ V_i^* &\sim \frac{1}{3} + \frac{1}{3}V_0, \frac{1}{3} + \frac{1}{3}V_1, \dots, \\ W_i^* &\sim \frac{2}{3} + \frac{1}{3}W_0, \frac{2}{3} + \frac{1}{3}W_1, \dots, \end{aligned}$$

sorozatok szintén egyenletesek rendre a $[0, \frac{1}{3})$, $[\frac{1}{3}, \frac{2}{3})$ és $[\frac{2}{3}, 1)$ intervallumokon, $i = 1, 2, \dots$

Továbbá az $U_0^*, V_0^*, W_0^*, U_1^*, V_1^*, W_1^*, \dots$ módon definiált, összefésült sorozat szintén egyenletes, azonban nem nevezhető véletlenszerűnek, hiszen 3 hosszú, szigorúan monoton növekvő részsorozat követi egymást.

E példa indukálhatja intuitíve a következő definíciót:

6. Definíció. Az U_0, U_1, \dots sorozatot k -egyenletesnek nevezzük, ha

$$P(u_1 \leq U_n \leq v_1; u_2 \leq U_{n+1} \leq v_2, \dots, u_k \leq U_{n+k-1} \leq v_k) = (v_1 - u_1)(v_2 - u_2) \dots (v_k - u_k)$$

az u_j és v_j számok minden olyan megválasztása mellett, ahol

$$0 \leq u_j < v_j \leq 1$$

és $1 \leq j \leq k$, továbbá $u_0, u_1, \dots \in [0, 1)$ és P a korábban már definiált, létező mértékkel megegyező.

Világos, hogy az egyenletes eloszlású sorozatok 1-egyenletesek, továbbá a k -egyenletesség hierarchikus abban az értelemben, hogy minden k -egyenletes sorozat $k-1$ -egyenletes is, hiszen $u_k = 0$ és $v_k = 1$ választás megengedett.

7. Definíció. Egy sorozat ∞ -egyenletes, ha $\forall k \in \mathbb{Z}^+$ számra k -egyenletes.

E fenti definíciók könnyen általánosíthatók bármely tetszőleges egész sorozatra.

8. Definíció. Egy k -jegyű b -alapú szám alatt egy k egézből álló $x_1 x_2 \dots x_k$ jelsorozatot értünk, ahol $0 \leq x_j < b$, amennyiben $1 \leq j \leq k$.

9. Definíció. Egy b -alapú sorozat k -egyenletes, ha

$$P(X_n X_{n+1} \dots X_{n+k-1} = x_1 x_2 \dots x_k) = \frac{1}{b^k}$$

minden b -alapú $x_1 x_2 \dots x_k$ -ra.

A ∞ -egyenletesség ezen elv mentén ugyanígy általánosítható.

Az igazi véletlen sorozatokban azonban kell hogy legyenek lokális nem-véletlenszerűségek – például egy „fej vagy írás”-sorozatban kellően hosszú „fej”-sorozatoknak. Azonban ez bizonyos alkalmazások számára gondot jelenthet – így azt is megállapíthatjuk, hogy nincsen olyan véletlenszerű számokat generáló eljárás, mely minden alkalmazáshoz megfelelő lenne.

A ∞ -egyenletesség alkalmas lehet bizonyos definíciók felállítására, melyek segítségével eljuthatunk olyan definíciókig, melyek az alkalmazásokat tekintve megfelelőek – és rendelkeznek a „véletlenszerűség” egyfajta tulajdonságával.

2.2.1. A „véletlenszerűség” meghatározásainak lehetséges változatai

Ezen alfejezet egy tudománytörténeti rész, melynek pontosítását, illetve szakirodalmi áttekintését a rákövetkező alfejezetben ismertetjük. Álljanak most itt a fontosabb definíciók, melyek történeti szerepét tehát még pontosítjuk, illetve kifejtjük a későbbiekben.

10. Definíció. (VSZ1) Egy $[0, 1)$ -sorozatot akkor nevezünk véletlenszerűnek, ha ∞ -egyenletes.

E fenti definícióval az a gond, hogy igazolható az, hogy nagyon liberális (kicsit hasonlatosan Lehmer definíciójához). Azonban jól szigorítható.

11. Definíció. (VSZ2) Egy U_n $[0, 1)$ -sorozatot véletlenszerűnek nevezünk, ha minden olyan P tulajdonságra, melyre $P(V_n) = 1$ valószínűséggel teljesül egyenletes eloszlású véletlen változók független mintáinak V_n sorozatára, $P(U_n) = 1$ is igaz.

A most definiált változat viszont túlságosan szigorú (gondoljunk a hosszú „fej”-sorozatokra például). Ez tehát túl sok – a VSZ1 viszont kevésnek bizonyul, hiszen pl. ∞ -egyenletes sorozatot tudunk racionális számokból is készíteni $[0, 1)$ -en, ami azonban gondot jelenthet.

12. Definíció. (VSZ3) Egy $[0, 1)$ -sorozatot akkor mondunk véletlenszerűnek, ha minden végtelen részsorozata ∞ -egyenletes.

A definíció érezhetően egyre pontosabb, azonban itt is van egy gyorsan észrevehető gond: minden sorozatnak van végtelen monoton részsorozata – ez pedig nem megfelelő számunkra. Továbbá e fenti definíció ráadásul túl erős is: Knuth bizonyítékai alapján [19] nem létezik olyan pszeudo-véletlen számsorozat, mely e fenti definíció alapján véletlenszerű lehetne.

Így több ok is mutatkozik arra, hogy javításokat eszközöljünk.

13. Definíció. (VSZ4) Az U_n $[0, 1)$ sorozat akkor véletlenszerű, ha különböző s_n természetes számok végtelen sorozatát előállító tetszőleges, effektív algoritmusra a megfelelő U_{s_0}, U_{s_1}, \dots részsorozat ∞ -egyenletes.

Igazolható, hogy ez a meghatározás sem megfelelő, mert pl. ha az effektív algoritmusunkban döntést kellene hozni, hogy adott érték $\frac{1}{3}$ alatt vagy felett van-e, akkor algoritmikus szempontból a 0,3333... végtelen számjegyet kéne ellenőrizni. Így az is belátható, hogy számunkra elegendő lenne, ha valamilyen módon az első n számjegyből döntést tudnánk hozni.

14. Definíció. Egy R részsorozat kiszámítható, ha (n, x_1, \dots, x_n) számokból $f_n(x_1, \dots, x_n)$ kiszámítható.

Azaz: az inputadatokból valamely algoritmussal az f_n érték meghatározható (a fenti, $\frac{1}{3}$ példát nézve, a döntés meghozható végtelen sok számjegy vizsgálata nélkül). Ez a definíció és a fenti megjegyzés indukálja a következő definíciót, melyet b -alapú sorozatokra mondunk ki:

15. Definíció. (VSZ5) Egy b -alapú sorozatot véletlenszerűnek nevezünk, ha minden kiszámítható részsorozat-szabállyal kijelölt részsorozat ∞ -egyenletes.

Ezzel a definícióval tudunk eljutni a legutolsó definícióig, melyet még teljesíteni tudunk egyszerű algoritmusok által előállított generátumok segítségével is, illetve kellően szigorú ahhoz, hogy ne tudja bármely generátum teljesíteni.

16. Definíció. (VSZ6) Az X_n b -alapú sorozatot véletlenszerűnek nevezünk, ha minden olyan effektív algoritmusra, mely n és $X_{s_0}, \dots, X_{s_{n-1}}$ függvényeként különböző s_n számok végtelen sorozatát jelöli ki, az algoritmusnak megfelelő X_{s_n} részsorozat az előző definíció szerinti értelemben véletlenszerű.

Az U_n $[0, 1)$ -sorozatot véletlenszerűnek nevezünk, ha $\lfloor bU_n \rfloor$ b -alapú sorozat véletlenszerű minden $b \geq 2$ egész számra.

Knuth igazolta [19], hogy míg a (VSZ3) definíció teljesíthetetlen követelményeket támaszt, addig a rákövetkező (VSZ4), (VSZ5) és (VSZ6) definíciók már – ugyanazon alapjelleg felhasználásával – jó irányt mutatnak. A (VSZ6) definíciót kellően liberálisnak tartja ahhoz, hogy teljesíteni lehessen követelményeit, de nem túlságosan megengedő.

2.2.2. Matematikatörténeti kitekintés

A témának lényegében átláthatatlanul nagy szakirodalma van, így csak összefoglaló jelleggel, koncentrálván a fenti pár definícióra szeretnénk néhány szerzőt megemlíteni – nyilvánvalóan a teljesség igénye nélkül. Egyes eredeti műveket nem sikerült felderíteni, így ott általánosságban Knuth összefoglaló művére [19] hagytam.

A ∞ -egyenletes sorozatokat N. M. Korobov munkatársaival tanulmányban [21] 1948-ban publikálta először, illetve tőlük függetlenül hasonló eredményekre jutott J. N. Franklin 1963-ban, melyet dolgozatban közölt [9].

Kuipers és Niederreiter közös könyvükben [22] többek között különböző fajta k -egyenletes sorozatokat vizsgáltak és publikáltak 1974-ben. Már ők is látták, hogy a ∞ -egyenletes sorozatok nem „elegendően kuszák”. Érdekes módon 1919-ben R. von Mises dolgozatában [28] már a fenti (VSZ5)-ös definícióhoz igen hasonlatos definíciót vizsált, de túl szigorúnak találta.

1928-ban A. H. Copeland dolgozatában [7] ezt a definíciót gyengíti úgynevezett „elfogadható számok” vagy Bernoulli-sorozatokkal való kombinálás segítségével. Ez lényegében egy (VSZ1)-re való visszatérés volt, ahol pl. egy adott érték alatti vagy feletti generátumokat 1-nek vagy 0-nak definiált.

1937-ben Wald Ábrahám tanulmányban [53] vizsgálta szintén ezt a definíciót, és lényegesen kevesebb gyengítést is elegendőnek ítélt (ráadásul egy algoritmust is definiált, melynek gyakorlati jelentősége nem sok van, azonban a (VSZ6)-nak megfelelő számsorozat előállítására alkalmas).

1940-ben A. Church kijavította Wald hibáit [6], továbbá ehhez kapcsolódik, hogy az algoritmust 1963-ban A. N. Kolmogorov javítási javaslata alapján – Knuth könyve [19] szerint – olyan állapotra tudták hozni, hogy már a (VSZ6) szerinti sorozatokat tudott generálni.

Szintén ide tartozik, hogy Kolmogorov ekkor javasolt véges véletlen sorozatokra is definíciót a fenti terminológiába való illesztés segítségével. Fontos megemlíteni, hogy jóval ezt megelőzően, már 1934-ben S. Besiovitc is olyan definíciót javasolt cikkében [2], mely véges sorozatokra vonatkozott. Szintén ezen – egymástól olykor teljesen független – kutatások részének tekinthető, hogy 1975-ben J. V. Howard a von Mises–Wald–Church-féle definíciót finomította tovább.

1966-ban D. W. Loveland dolgozatában [24] a (VSZ4)–(VSZ5)–(VSZ6) kapcsolatait kezdte vizsgálni. Szintén ehhez a vonulathoz tartozik, hogy 1976-ban Thomas Herzog és James C. Owings, Jr. olyan sorozatot konstruált, mely (VSZ4)-et teljesíti, de (VSZ6)-ot nem [14].

Másfajta elgondolásokkal próbálkozott többek között Karl Popper is, aki filozófiai megközelítést alkalmazott 1934-ben [34]. Kolmogorov 1965-ben információtartalommal próbálta leírni a véletlenszerűséget [20]. 1966-ban D. W. Loveland már hivatkozott dolgozatában [24], és 1969-ben, illetve 1971-ben C. P. Schnorr olyan műveket publikáltak [42], melyekben rekurzív függvényekkel összekapcsolva vizsgálták a véletlenszerűséget. 1968-ban Chaitin és Martin-Löf szabálytalanságra alkotott definíciót [26].

Jól látható, hogy nagyon sokan és relatíve már nagyon korán elkezdték vizsgálni a fenti kérdéskört, hiszen alkalmazási szempontból nagy jelentősége van egy jó véletlenszerű számsorozat vagy általában sorozat generálásának.

2.2.3. Gyakorlati megközelítések különböző területeken

Gyakorlati szempontból vizsgáltuk a véletlenszerű számsorozatokat – ezek vizsgálatából kiderül, hogy mit is kell gyakorlatban egy számsorozatnak teljesíteni, hogy véletlennek gondoljuk.

Az elmúlt időszakból két tanulmányt is említhetünk, melyekben közvetlenül véletlen vektorok generálásának problémájára, illetve alkalmazására hívják fel a szerzők a figyelmet. Onn és Weissman tanulmányukban [33] egy adott szimplexbe generálnak folytonos, egyenletes eloszlású véletlen vektorokat, és a generátor minőségi tulajdonságait is vizsgálják – az alkalmazáshoz igazítva. Az eredményeket általában geometriai módszerekkel szemlélik – ellentétben az általunk használt algoritmus vizsgálatával.

Hu és Cui cikkükben [15] azzal a problémával foglalkoznak, hogy a legtöbb programnyelvben véletlen vektorokat oly módon hoznak létre, hogy véletlen számokat generálva azokat vektorokba rendezik. Ennek problematikájára a későbbiekben mi is kitérünk.

Mint az a történeti áttekintésből és a jelenkor két frissebb tanulmányából is látszik, a téma feldolgozása nem lehet univerzális. Alkalmazásokhoz, alkalmazási területekhez kell igazítani a generátorokat – és ezzel együtt olykor a definíciókat is.

Esetünkben ez az alkalmazás azt indokolná – mint az tapasztalható az utolsó fejezetben – hogy valamely robusztus optimalizálási eljárást vessünk be a feladat megoldására.

Ennek lehetséges módozatairól összefoglaló leírásokat olvashatunk Ben-Tal és Nemirovski 2002-es tanulmányában [1], vagy 2009-es könyvében [3]. Ezek mindegyikében számtalan eszközt és problémátípust találhatunk. Hasonlóan átfogó egyetemi jegyzetek állnak rendelkezésünkre a témában, pl.: az M. I. T. egyetem azonos nevű kiadványában Sim [44] átfogó módon értekezik a témáról. Goh és Sim tanulmányukban [11] a robusztus optimalizálást lineáris programozási feladatok esetére alkalmazzák, specifikálva a problémát (azaz a megoldást az adott problémára adaptálva, nem általánosítva magát az eljárást).

Hasonló eseteket tanulmányok, konferenciák összefoglaló tanulmányköteteiben is találhatunk, például Schied tanulmánykötetében [41] a Pueblában 2004-ben megrendezett 8. Valószínűségelméleti és sztochasztikus eljárások szimpóziumának előadásait foglalja össze.

A robusztus optimalizálás lehetséges alkalmazása indokolta, hogy jó minőségű véletlen egézsékből álló vektorokat generáló algoritmusokat kezdjünk el vizsgálni – amihez persze tisztázni kellett, hogy mit is értünk a matematikán belül véletlennek.

2.2.4. Matematikai területek megközelítései a tapasztalt jelenségre

Semmiképpen sem vállalkoznánk arra, hogy minden tekintetben teljes, átfogó képet adjunk a tudományos élet valamennyi részterületének „véletlenről” alkotott fogalmi rendszeréről. Azt már láthattuk, hogy alkalmazásokat szintén igen változatos területeken lehet találni.

Most néhány matematikai ág segítségével próbálunk képet mutatni arról, hogy e látszólag egyszerű (bár mostanra talán már érezhetően bonyolult) fogalom hány különböző formában létezhet akár egy tudományterületen belül is:

- **Valószínűségelmélet**

Egyik oldalról a valószínűség-elméleti megközelítés azt mondja, hogy ne legyen előre jósolható, kitalálható a sorozatunk következő eleme az előzmények ismeretében. Ez egybevág azzal a megközelítéssel, mint amit a társasjátékok során tapasztalunk: nagyban megváltozna a dobókockával játszott, stratégiát igénylő játékok eredménye, ha a játék során, annak első felében végrehajtott dobások eredménye befolyásolná a játék második felében történő kockadobások kimenetelét.

- **Kombinatorika – bonyolultság-elmélet**

A bonyolultság-elméletben használatos elvárás, hogy ha egy külső szemlélőnek átadjuk a sorozatot, akkor az ne tudja tömöríteni (azaz ha ezt algoritmus csinálta is, a sorozatból ne legyen könnyen kitalálható az algoritmus). Más megfogalmazásban: a sorozatunk információtartalma a sorozat hosszával összemérhető legyen. Világos, hogy pl. egy $1 - 0 - 1 - 0 - 1 - 0 \dots$ sorozat ezt nem tudja garantálni.

- **Dinamikai rendszerek**

Dinamikai rendszereket vizsgálva a kaotikus rendszerek rendelkeznek azzal a tulajdonsággal, hogy – miután tetszőleges pontossággal a gyakorlatban sosem adható meg minden bemeneti paraméter (pl. gondoljunk arra, hogy az egyik bemeneti paraméterünk irracionális) – a kaotikus rendszerek kiszámíthatatlanoknak, így véletlennek is tekinthetők.

E három megközelítést, illetve ezek filozófiai mondanivalóját fogjuk felhasználni olyan módon, hogy milyen gyakorlati kritériumok szerint tudjuk eldönteni, hogy e területek által definiált vagy használt „véletlenszerű” jelenségekhez közeli sorozattal dolgozunk-e.

2.2.5. Pseudo-véletlen sorozatokat generáló algoritmusok

A fejezet ezen része az Alkalmazott Matematikai Lapokban megjelent [49] cikkünkben jelent meg. Vizsgálatainkat az EXCEL[©] programban valósítottuk meg, mert az alkalmazást, amiben ezt az algoritmust használtuk, EXCEL[©]-környezetben tudtuk létrehozni. Referenciaként pedig a számunkra rendelkezésre álló MATLAB[©] programot használtuk – mely programmal a felhasználóink azonban nem rendelkeznek.

Jó minőségű pseudo-véletlen számokra, vektorokra számos alkalmazásban szükség van. A vektorrendszer generálásához használt algoritmust számos kritérium alapján kell vizsgálnunk, melyek egyik része általánosan elfogadott kritérium, amit egy algoritmussal, véletlen számokat/vektorokat generáló rutinnal szemben követelményként meg kell fogalmaznunk. Másik oldalról a felhasználás során esetleges specifikus igények is megfogalmazódhatnak egyes eljárásokkal szemben.

A tézisben vizsgált algoritmus korábban már publikálásra került két cikkünkben [49], [51], illetve Ramirez könyvében [37]. A vizsgálatok során, hogy az eljárásunk megfelelő minőségű legyen, bizonyos módosításokat hajtottunk végre.

Knuth és Marsaglia által megfogalmazott [18], [27], véletlen számsorozatokra vonatkozó kritériumokat általánosítottunk, illetve adaptáltunk véletlen vektorok minőségének tesztelésére. Eljárásunkat a MATLAB[©] véletlen egészeket generáló rutinjával hasonlítottuk össze, illetve önmagában is vizsgáltuk az algoritmusunk egyes általános tulajdonságait.

Mindezen vizsgálatokból kiderült, hogy az algoritmus megfelel egy általánosan pseudo-véletlen vektorokat generáló rutin elé támasztott követelményrendszernek – továbbá egy általunk relevánsnak vélt alkalmazásban is megállta a helyét.

Álljon most itt néhány lehetséges vizsgálati eszköz, melyek segítségével pseudo-véletlen sorozatokat készítő algoritmusokat tesztelhetünk – természetesen a teljesség igénye nélkül.

1. GYAKORISÁGVIZSGÁLAT

Például egy megfelelő χ^2 -próba vagy a Kolmogorov-teszt alapján annak eldöntése, hogy az $\{a_i\}$ számsorozat egyenletes eloszlásúnak tekinthető-e az $[a, b]$ intervallumon.

Természetesen a Kolmogorov-teszt nem tekinthető valódi gyakoriságvizsgálatnak, így Mogyoródi [29] szóhasználatával ellentétben talán az eloszlásvizsgálat kifejezés találhatóbb lenne.

Nagyon sok alkalmazható teszt található Prékopa könyvében [36] is, melyek adaptálhatók, illetve bevethetők e pontján a vizsgálatoknak.

2. SZÁMJEGYVIZSGÁLAT

Vagy számjegygyakoriság-vizsgálat. Amennyiben egy „fej vagy írás”-játékhoz hasonlóan kettes számrendszerben felírt számokként képzeljük el a generátumunkat, úgy jogos feltételezni, hogy 0 és 1 számok relatív gyakorisága $\frac{1}{2}$ körül ingadozik.

Természetesen ezt a tesztet más számrendszerekben felírva a számjegyek relatív gyakorisága szintén tesztelhető.

3. KOMBINÁCIÓVIZSGÁLAT

Ebben a tesztben azt tudjuk megvizsgálni, hogy az egyes számokban előforduló számjegyek különböző kombinációi milyen relatív gyakoriságokkal fordulnak elő – az elméletben elvárhatóhoz képest.

Lényegében e kombinációvizsgálat számtalan speciális esete található meg Marsaglia tesztjeiben [27], illetve Knuth számos tesztje [19] is e vizsgálatok közé sorolható.

Álljon most itt Mogyoródi [29] egy konkrét példája egy ilyen elméleti kombinációvizsgálat megvalósítására.

4. HÉZAGOK VIZSGÁLATA

Legyen $0 \leq \alpha < \beta \leq 1$. Ha $a_j, a_{j+1}, \dots, a_{j+k-1}$ nem eleme $[\alpha, \beta]$ -nak, de $\alpha_{j+k} \in [\alpha, \beta]$, akkor azt mondjuk, hogy ebben a sorozatban egy k hosszúságú hézag van.

Egy k hosszúságú hézag elméleti valószínűsége (egyenletes eloszlást feltételezve)

$$p_k = (\beta - \alpha) (1 - (\beta - \alpha))^k.$$

Ezt hasonlítjuk össze az $\{a_i\}$ sorozatból kapott tapasztalati relatív gyakorisággal (pl. χ^2 -próba segítségével).

Az eddigi tesztek alapvetően a generátumok eloszlásának vizsgálatára voltak alkalmasak. Mogyoródi [29] jegyzete egy általánosabb javaslatot tesz egy korrelációs együttható definiálásának segítségével összefüggőségek vizsgálatára.

Ezen kívül a már említett Marsaglia [27] és Knuth [18] is javasolnak bőségesen konkrét példákat a sorozatban fellelhető rejtett összefüggések kiderítésére – illetve mi magunk is adaptáltunk egyéb területekről származó összefüggőségi mutatókat e speciális vizsgálatához. Néhány példát mutatunk arra vonatkozóan, hogy számsorozatok esetén általában milyen eljárások vehetők be – majd utána bemutatásra kerülnek azok az eljárások, melyeket a véletlen egész vektorokat generáló algoritmusunkhoz általánosítottunk, illetve adaptáltunk.

17. Definíció. Az $\{a_i\}_{i=1}^n$ számsorozat széria-korrelációs együtthatója:

$$r = \frac{n(a_1 a_2 + a_2 a_3 + \dots + a_{n-1} a_n + a_n a_1) - (a_1 + \dots + a_n)^2}{n(a_1^2 + \dots + a_n^2) - (a_1 + \dots + a_n)^2},$$

ahol azt feltételezzük, hogy a nevező nem 0.

Megjegyezzük, hogy e fenti mutató azt méri, hogy a_{j+1} mennyire függ a_j -től (azaz lényegében egy 1 lépéses emlékezetet tudunk vizsgálni ezen mutató segítségével).

Annak tesztelésére, hogy mennyire jó egy generátum ezen mutatója, tudnunk kell a fenti mutató elméleti viselkedését (ahogy egy Pearson-féle korrelációs együttható vizsgálatánál is ismernünk kell annak elméleti paramétereit).

1. Lemma. *Maximális periódusú lineáris kongruencia sorozatra*

$$r \approx \frac{1}{a} \left(1 - 6 \frac{c}{m} + 6 \left(\frac{c}{m} \right)^2 \right),$$

ahol a közelítés hibája nem haladja meg az $\frac{a+6}{m}$ -et.

1. Megjegyzés. *A statisztikai becslések esetén fontos szempont lehet, hogy egy-egy becsült paraméter hibáját is becsülni tudjuk. Itt nagyon leegyszerűsítve arról van szó, hogy e mutató véletlen mintavételezésből fakadó elméleti hibájának felső mértékét meg tudjuk adni, így adott esetben egy „legszélesebb”, adott valószínűség melletti konfidencia-intervallumot tudunk köré szerkeszteni.*

Ennek egy általánosítását tettük akkor, amikor a későbbiekben ismertetésre kerülő Kendall-féle mutatókat szimulációs technika segítségével becsültük.

Továbbá e technikák tanulmányozása és gyakorlati különbségei találhatóak meg cikkünkben [48], melyben egy egészen más helyzetben, de e hibabecslés fontosságára hívjuk fel a figyelmet.

2.3. Az általunk alkalmazott tesztek ismertetése

Az általunk használt algoritmus teszteléséhez előre meghatároztuk, hogy mely teszteken szeretnénk, ha jól szerepelne. Igaz ugyanis az az állítás, hogy nem létezik olyan pseudo-véletlen számokat/vektorokat generáló algoritmus, melyhez ne lehetne találni egy tesztet, mely kimutatja annak algoritmikus voltát (nevezetesen, hogy nem véletlen) – lásd Knuth munkáját [19].

A tesztek egy része a generált vektorrendszerünk egyenletességét volt hivatott ellenőrizni, míg a tesztek másik része a vektorok közötti esetleges összefüggőséget vizsgálta. Mindkét szempont nagyon fontos: amennyiben a generált vektorok nem egyenletesek, úgy az azt jelentené, hogy sűrűsödési pontjai vannak a generátumnak, bizonyos területekre gyakrabban tesz pontokat – ezzel lényegében alkalmazhatatlanná teszi az algoritmust.

E sűrűsödési pontok szintén alkalmasak lehetnek különféle összefüggőségi vizsgálatokra, mint ahogy tettük azt új klasszifikációs módszereket taglaló, pszichometriai cikkünkben [46].

Az összefüggőség tehát már egy bonyolultabb kérdés: itt elsősorban azt teszteltük, hogy már generált pontokból, vektorokból kitalálhatóak-e a következő vektorok vagy sem. Illetve, ami a vektorok esetén szintén fontos szempont, hogy a vektorok egy-egy koordinátájából, vagy akár több koordináta figyelembe vételével tudunk-e más koordinátákra, pontokra előrejelzést adni.

Ennek érdekében páros összehasonlításokat végeztünk, illetve a Knuth által javasolt [18], teljes összefüggőség mérésére használt tesztet is adaptáltuk a vektor-változók esetére.

2.3.1. Az egyenletesség tesztelése

Kolmogorov–Szmirnov-statisztika

Az első teszt, amit alkalmaztunk, a nagy elemszámra való tekintettel a Kolmogorov–Szmirnov-próba volt. Ennek során először a peremeloszlásokat teszteltük. Majd minden perem esetén azokat 2, 4 és 5 részre bontottuk, és az ezen vágások mentén kialakult téglákban vizsgáltuk a többi koordináta egyenletességét.

2. Megjegyzés. Ezen a ponton nyer majd jelentőséget az a tény, hogy a szigorú pozitív kvadránsba generáltunk vektorokat, ugyanis így valóban értelmes osztópontokat hozhattunk létre – nem 50 – 51 pont került az egyes vágásokba, hanem 50 – 50, vagy 25 – 25 – 25 – 25, stb.

A Kolmogorov–Szmirnov-tesztek közül az egyváltozós esetek előfeltétele, hogy folytonos F eloszlást vizsgálunk. Ismert, hogy a kétváltozós esetnek is az a feltétele, hogy mindkét valószínűségi változó egy-egy folytonos F és G eloszlásból származzon, majd ezen F és G eloszlásokat vizsgáljuk egyenlőség szempontjából.

A Kolmogorov–Szmirnov-tesztek kétváltozós esetében a véletlen egész vektorokat úgy kezeljük, mintha folytonos eloszlásból származnának. (Például IQ-tesztek esetén is úgy feltételezzük, hogy folytonos eloszlás húzódik meg a háttérben, csak a mérőeszközünk nem tudja ezt mérni.)

Így az alábbi eljárást fogjuk elvégezni:

1. Generálunk egy V $n \times m$ -es vektorrendszert, azaz m darab n dimenziós vektort. Azt vizsgáljuk, hogy az m darab vektor minden egyes peremeloszlása egyenletes-e.

Válasszuk ki például V első oszlopát (ami most így egy m hosszú vektor lett) tesztelésre, jelölje ezt V_1 .

Legyen Max és Min V_1 maximális és minimális eleme.

2. Vesszünk egy n dimenziós w vektort, mely Max és Min között minden egész értéket felvesz, ráadásul mindet egyenlő mértékben. Ez azt jelenti, hogy minden értékből

$$\left\lfloor \frac{n}{Max - Min + 1} \right\rfloor$$

darabot veszünk.

Ezzel elérjük, hogy a V_1 vektorral majdnem megegyező hosszúságú, annak legnagyobb és legkisebb eleme között egyenletes eloszlásból származó w vektorunk legyen.

3. Most w és V_1 eloszlását a fent nevezett illeszkedés-vizsgálati eszközökkel összehasonlítjuk. Így azt feltételezhetjük, hogy mindkettő ugyanazon mérőeszközzel készült, egyenletes eloszlásból származó minta – továbbá a tesztnek az elemszámok nagyon eltérő volta miatti gyengülését mindenképpen kizárjuk, hiszen majdnem megegyező méretű vektorokat hasonlítottunk össze.

χ^2 -statisztika

A második teszt – a fenti téglák kombinációján – annak tesztelése volt, hogy minden téglában egyenletesen vannak-e jelen a generált rácpontok. Ez persze 2^n , illetve 4^n , 5^n téglát jelentett, melyekre χ^2 -próbát alkalmaztunk.

Ennek feltételét könnyebb volt garantálni, mint a Kolmogorov–Szmirnov-statisztika esetén elvárt folytonossági kritériumon, ugyanis itt csak a téglák kellő elméleti gyakoriságán kellett túllépnünk – amit azonban a generált vektorok számával azonnal teljesíteni tudtunk.

Így a fenti felosztás mellett már csak a szokásos,

$$\chi^2 = \sum_{i=1}^M \frac{(n_i - \nu_i)^2}{\nu_i}$$

Pearson-féle χ^2 -statisztikát kellett kiszámítani, ahol M az éppen aktuális felosztás számát jelenti (az oldalak felosztása alapján ennyi darab téglá keletkezett).

2.3.2. Az összefüggőség tesztelése

A Kendall-féle monotonitási együttható

Tegyük fel, hogy adott $A = (X_1, Y_1)$ és $B = (X_2, Y_2)$ pontpár. Amennyiben $X_1 > X_2$ és $Y_1 > Y_2$, úgy azt mondjuk, hogy A és B konkordáns viszonyban vannak egymással (pozitív irányú az összefüggés, monoton növekvő kapcsolat van közöttük). Amennyiben $X_1 > X_2$, de $Y_1 < Y_2$, úgy azt mondjuk, hogy A és B diszkordáns viszonyban vannak egymással (negatív irányú az összefüggés közöttük, monoton csökkenő kapcsolatot mérünk).

Világos, hogy ha egy ponthalmazban a konkordáns párok p_+ száma a nagyobb, úgy azt mondhatjuk, hogy a ponthalmaz összességében monoton növekvő, a pontok egymáshoz képest monoton növekednek. Amennyiben a diszkordáns párok p_- száma a nagyobb, úgy ennek ellenkezőjét állíthatjuk – azaz monoton csökkenő viszonyt vélelmezhetünk a ponthalmazunk esetén. Tehát

$$p_+ = \# \{ (A(X_1, Y_1), B(X_2, Y_2)) \mid (X_1 < X_2) \wedge (Y_1 < Y_2) \},$$

$$p_- = \# \{ (A(X_1, Y_1), B(X_2, Y_2)) \mid (X_1 < X_2) \wedge (Y_1 > Y_2) \}.$$

Az is világos, hogy nem minden pontpárt tudunk összehasonlítani – hiszen lehetnek egyező X vagy Y koordináták. Erre az esetre olyan korrekciót hajtunk végre, hogy p_+ és p_- számok különbségét nem az összes létező párhoz viszonyítjuk, hanem az összehasonlítható párokra. Az így képezett monotonitási együtthatók:

18. Definíció. A Kendall τ -ként az alábbi mennyiséget definiáljuk a fentiek alapján:

$$\tau = p_+ - p_-.$$

E fenti mutatót – miután bár sok értéket felvehető vektorokon teszteljük, a fent már említett, összehasonlíthatatlan párok miatt érdemes módosítani az alábbi módon:

$$\gamma = \frac{p_+ - p_-}{p_+ + p_-}.$$

3. Megjegyzés. *Technikai kérdés, hogy a pontpárok összehasonlítását miként végezzük. Ha sorban haladva a már összehasonlítottakat kihagyjuk, vagy rosszabb szervezés miatt minden pontot minden ponttal akár többször is összehasonlítottunk, az arányokon, illetve azok hányadosán ez a részlet nem fog változtatni. Fontos azt is kiemelni, hogy folytonos eloszlások esetén e két mutató megegyezik. Az előforduló – ugyan csak kis számú – egyezés miatt mi a γ -val jelzett mutatót részesítettük előnyben.*

Jackknife-módszer segítségével mérjük az így elkészített monotonitási együtthatók 95%-os konfidencia-intervallumát. Ezt úgy tesszük, hogy a generált vektorok közül mindig kihagyva a megfelelőt, pszeudo-statisztikákat nyerünk a fenti monotonitási mértékre, majd utána trimmelés segítségével (adott számú legnagyobb és legkisebb elem elhagyásával) meghatározzuk a kívánt szélességű intervallumot.

A Jackknife-eljárásról bővebben pl. Efron cikkében [8] olvashatunk. A trimmelés során nem teszünk mást, mint a nagyság szerint sorbarendezett elemek első és utolsó darabjaitól (egy előre meghatározott arányban, szimmetrikusan) megszabadulunk.

A cikkünkben [48] megtalálhatók különböző szimulációs módszerek, illetve a Jackknife-eljárás egy összehasonlító esettanulmánya, melynek alapján hivatkozhatunk arra, hogy ezen terminológia erre a helyzetre számunkra teljesen kielégítő, hiszen pl. egy bootstrap szimuláció újabb véletlenszerűséget vinne a mintánkba, melynek alkalmazása ehelyütt teljességgel felesleges.

A Jackknife-módszer alkalmazási feltételei fennállnak ahhoz, hogy torzítatlan becslést nyerjünk belőle a számítandó paraméter konfidencia-intervallumára. Amennyiben ez az intervallum a pont-halmazunk esetén tartalmazza a 0 értéket, úgy azt mondhatjuk, hogy a pont-halmazunk lényegében azonos arányban tartalmaz monoton növekedő és csökkenő pontpárosokat.

Átrendezés- és sorozatteszt

Knuth könyve [18] alapján a sorozattesztet, illetve annak átdolgozását alkalmaztuk.

1. Átrendezésteszt

Az eljárásunkat abból a szempontból vizsgáljuk, hogy a monoton növő és csökkenő részsorozatok tulajdonságai átrendezés hatására megváltoznak-e. Ugyanis ha az átrendezés változtat az eredményeken, az azt jelenti, hogy menet közben valamit kihasználtunk – vagy elvesztettünk, és ennek hatása van a sorozatunk viselkedésére.

Az átrendezést az alábbi módon értjük: tekintsük az adott, generált V vektorrendszerünket. A V vektorrendszert egy permutáció segítségével rendezzük át, a permutáció által adott sorrendben felsorolva újra a vektorokat, legyen ez V^* . Amennyiben a monoton növő és csökkenő részsorozatok aránya, száma V és V^* esetén szignifikánsan eltérő, úgy azt mondhatjuk, hogy az eredeti sorozatunkban jelentősége volt a vektorok sorrendjének, az nem egészen volt véletlenszerű.

2. Sorozatteszt

A sorozatteszt Knuth könyvében [18] megtalálható, melynek lényege, hogy a véletlen számok generálásakor vizsgálható, hogy hány monoton növe és csökkenő részsorozatot generál az algoritmus. Ezek természetesen nem függetlenek egymástól (így pl. az esetszám tesztelésére egyszerű χ^2 -próba nem alkalmazható). A függetlenség azért sérül, mert teljesen nyilvánvaló módon egy növekedő sorozatot mindenképpen egy csökkenő kell hogy kövessen.

Lényegében annyi történik, hogy összeszámoljuk: a generált vektorrendszerünkön lexikografikus rendezést végzünk. A rendezés után minden vektor helyett a rangsorban betöltött helyét vizsgáljuk tovább: a kérdés az lesz, hogy hány darab, milyen hosszú monoton növe részsorozat keletkezett. Ezeket utána a fenti [18] hivatkozáson megadott módon statisztikai vizsgálatnak vetjük alá.

A sorozatteszt vagy más néven futampróba az alábbi módon írható át véletlen vektorokra szintén Knuth könyve [18] alapján.

Amennyiben adott egy W generált vektorrendszerünk, úgy tekintjük annak V – lexikografikusan rendezett – párját, és minden eredeti W -beli vektor helyett nézzük annak v_i , V rendezésben elfoglalt rangját.

Ezzel a W vektorrendszert visszavezettük egy 1 dimenziós véletlen sorozatra, melyben értelmezhetőek a szigorúan monoton növe és szigorúan monoton csökkenő részsorozatok.

4. Megjegyzés. *Annak a valószínűsége, hogy kapcsolt rangok jöjjenek létre, nagyon alacsony, pláne ritkítás mellett (akkor jöhet létre, ha ugyanazon csúcspontba érünk vissza).*

A monoton növe és csökkenő sorozatokat osztályozzuk hosszuk szerint – legyen tehát $s(i)$ az i hosszúságú növe sorozatok száma. Azaz: egy monoton csökkenő sorozatot több, 1 hosszú monoton növe sorozatként fogunk tárolni $s(i)$ -ben.

Emlékeztetnénk rá, hogy $s(i)$ eloszlására nem alkalmazható közvetlenül a χ^2 -próba, hiszen $s(i)$ elemei nem lehetnek egymástól függetlenek (egy monoton növe sorozatot mindenképpen egy csökkenő sorozat követ, ráadásul egy hosszú sorozat után mindenképpen rövid sorozatnak kell következnie).

Így világos az is, hogy a monoton növe és csökkenő sorozatok szerepe szimmetrikus, azaz tesztelésként elegendő az egyiket figyelni csak (persze az is tesztként szolgál, hogy a monoton növe és csökkenő sorozatok eloszlása megegyezik-e egymással).

A sorozattesztet oly módon építettük fel, hogy 1, ..., 5 hosszú sorozatokat számoltunk, illetve a 6 vagy annál hosszabb sorozatokat összevontuk, tehát az utolsó, $s(6)$ érték definíció szerint a 6 vagy annál hosszabb monoton növe sorozatok számát jelöli.

Ezután az alábbi összegzést végeztük el:

$$C = \frac{1}{K} \sum_{1 \leq i, j \leq 6} (s(i) - nb_i)(s(j) - nb_j) a_{i,j},$$

ahol K generált W vektorok száma, $s(i)$ a fent definiált, monoton növe, adott hosszúságú részsorozatok száma, míg $a_{i,j}$ és b_i számokat az alábbiak szerint definiáltuk:

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & a_{36} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} & a_{56} \\ a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & a_{66} \end{pmatrix} = \begin{pmatrix} 4529,4 & 9044,9 & 13568 & 18091 & 22615 & 27892 \\ 9044,9 & 18097 & 27139 & 36187 & 45234 & 55789 \\ 13568 & 27139 & 40721 & 54281 & 67852 & 83685 \\ 18091 & 36187 & 54281 & 72414 & 90470 & 111580 \\ 22615 & 45234 & 67852 & 90470 & 113262 & 139476 \\ 27892 & 55789 & 83685 & 111580 & 139476 & 172860 \end{pmatrix},$$

míg a b_i számokat:

$$\left(b_1 \quad b_2 \quad b_3 \quad b_4 \quad b_5 \quad b_6 \right) = \left(\frac{1}{6} \quad \frac{5}{24} \quad \frac{11}{120} \quad \frac{19}{720} \quad \frac{29}{5040} \quad \frac{1}{840} \right).$$

E fenti mátrixban kerekített értékek vannak, melyek képlete Knuth munkájában [18] megtalálható. A fenti C statisztika χ^2 -eloszlást követ 5 szabadsági fokkal – az alkalmazáshoz legalább 4000 vektor szükségeltetik (az elméleti gyakoriságok kellő nagyságának garantálása érdekében).

2.4. A vizsgált algoritmus ismertetése

2.4.1. Az algoritmus eredeti változata

Az algoritmus egész vektorok egy véges, rögzített halmazából véletlenszerűen kiválasztott vektorokat ad össze, és az eredményt behúzza egy meglehetősen hosszú élű kockába (általánosításban: bármely tartományba). Lépéseit a következőképpen fogalmazhatjuk meg:

1. Inicializálás

$$A \in \mathbb{Z}_+^{n \times m}, \quad n \ll m, \quad n, m \in \mathbb{Z}_+$$
$$\xi_j \sim U(m) \in \mathbb{Z}_+, \quad j \in [1, \dots, K], \quad D \subset \mathbb{Z}_+^n, \quad K \in \mathbb{Z}_+,$$

ahol

- A mátrix összes $n \times n$ -es részmátrixának determinánsaira teljesül, hogy a legnagyobb közös osztójuk 1;
- ξ_j véletlen, egyenletes eloszlású valószínűségi változók;
- D a pozitív ortáns adott d élhosszúságú kockája (természetesen itt bármilyen tartomány választható);
- K számú véletlen egész vektort szeretnénk generálni.

Azaz egy D tartomány – jelen leírásban a d élhosszúságú, pozitív ortánsban elhelyezkedő n dimenziós kocka egész rácspontjaira szeretnénk véletlen vektorokat generálni.

2. Ciklus és levágás

$$\mu_0 = \mathbf{0},$$
$$\nu := \xi,$$
$$\mu_i = \mu_{i-1} + a_\nu,$$
$$i = 1 \dots K.$$

Ha $\mu_{i,j} > d$ (vagy $\mu_{i,j} < 1$), akkor $\mu_{i,j} := \mu_{i,j}^*$, ahol

$$\mu_{i,j} \equiv \mu_{i,j}^* \pmod{d},$$
$$1 \leq \mu_{i,j}^* \leq d,$$
$$j = 1, \dots, n.$$

Azaz a $\mathbf{0}$ vektorból indulva az A mátrix oszlopainak véletlen sorrendben – egyenletes eloszlású valószínűségi változók realizációinak segítségével – történő egységnyi együtthatójú lineáris kombinációját vesszük, ezek adják a véletlen vektorokat.

2. Tétel. Amennyiben $\{a_1, \dots, a_m\} \subset \mathbb{Z}^n$ vektorrendszer tartalmazza az \mathbb{R}^n egy lineáris bázisát, a determinánsokra vonatkozó feltétel fennáll, és $\exists t \in [1, m] : a_t = 0$, úgy:

$$\forall u \in D : \lim_{k \rightarrow \infty} P(\mu_k = u) = \frac{1}{\prod_{j=1}^n d_j} = d^{-n},$$

azaz minden u rácspont egyenlő valószínűséggel vétetik fel, amennyiben a generált vektorok száma a végtelenhez tart, lásd Vizvári kutatási riportját [52].

5. Megjegyzés. A tételben szereplő d_j értékek most egyenlőek, hiszen a d élhosszúságú kocka rácspontjaira generálunk. Általánosságban pedig $D = (d_1, \dots, d_n)$ pozitív egész élhosszak által meghatározott téglába állítjuk elő a vektorokat.

Bizonyítás: A tétel bizonyítása Vizvári [52] hivatkozáson megtalálható. ■

3. Következmény. A tételből az is következik, hogy a generátorunk ciklushossza nem a szubrutin-ként alkalmazott véletlenszám ciklushosszától függ elsősorban, hanem a D tartomány rácspontjaitól, ugyanis ezek mindegyike pozitív valószínűséggel előáll – így elvárható, hogy e pozitív valószínűségek be is következzenek. Ezzel tehát az algoritmusunk egy érdekes tulajdonsággal bír: a ciklushossza a felhasznált tartomány nagyságától fog függni – **ily módon tetszőlegesen növelhetővé válik a generátorunk ciklizálásának várható ideje.**

2.4.2. Tapasztalatok utáni javítások, fejlesztések

6. Megjegyzés. Az általunk alkalmazott algoritmusban seed beállítás is szerepelt, ami azt jelenti, hogy amennyiben K vektorra volt szükségünk, úgy a generált μ vektorokat csak egy bizonyos érték után kezdtük el egy mátrixba feltölteni (a seed érték eléréséig szabadon generált a program, nem történt kiíratás). Ez random-seedet eredményezett számunkra, hiszen így az adott D tartomány egy pseudo-véletlen pontjából indítottuk el az algoritmust.

7. Megjegyzés. Az eredeti algoritmus leírásában látszik, hogy egyetlen ponton van döntési lehetőségünk: az A mátrixunk megválasztásában. Az algoritmushoz olyan A mátrixra van szükségünk, melyre igaz, hogy az $n \times n$ -es al-determinánsok legnagyobb közös osztója 1.

Amennyiben ez nem teljesül, úgy az algoritmus során sávokat generálnánk, nem tudnánk kitölteni az egész, előre meghatározott, kitöltendő D tartományt.

Az A mátrixot első körben az alábbi technikával generáltuk: egy $n \times n$ -es egységmátrixon végrehajtottunk adott számú (n^2) Gauss eliminációs eljárást fordított irányban (ezzel elértük, hogy a kapott mátrix determinánsa továbbra is 1 maradjon). A fordított irány azt jelentette, hogy most az egységmátrix sorainak véletlen számszorosát hozzáadtuk/kivontuk egymásból.

Ezt az eljárást alkalmaztuk egymás után többször. Így garantáltuk, hogy minden $n \times n$ -es al-determináns legnagyobb közös osztója 1 maradt, hiszen tartalmazott több olyan rész-mátrixot is, melynek determinánsa 1.

Fontos azt is megemlíteni, hogy az algoritmus mindenképpen használ egy külső, véletlenszám-generátort, mellyel egyenletes eloszlásból származó pszeudo-véletlen számokat nyerünk a vektorok összegeinek előállításához, illetve ezt a generátort használjuk az A mátrix előállításához is.

Az algoritmus teszteléséhez különböző statisztikai eljárásokat alkalmaztunk. Egyik oldalról fontos szempont volt az eloszlás egyenletességének tesztelése. Erre részint χ^2 -próbát, részint Kolmogorov–Szmirnov-tesztet választottunk.

Az egyenletességet vizsgáló tesztek a 100 oldalhosszúságú, pozitív ortánsban elhelyezkedő, 2, 3 és 5 dimenziós kockába történő 2000 vektor generálására végeztük el.

A vektoraink függetlenségének tesztelésére a későbbiekben bemutatásra kerülő tesztekkel használtuk. Szintén a 100 oldalhosszúságú, pozitív ortánsban elhelyezkedő kockába generált véletlen egész vektorokkal dolgoztunk, azonban a megbízhatóság érdekében 5000 vektort generáltunk (a szakirodalomban a minimális 4000-es érték szerepel [18]).

2.4.3. Módosítás az algoritmuson

Kétfajta módosítást hajtottunk végre az algoritmuson. Az egyik az A mátrix generálása, a másik pedig egy ritkítás beállítása.

Generálás

Az algoritmus lelkét képező A mátrix megválasztása fontos lépés. A módosításban az alábbi lépéseket tettük:

1. Először a már ismertetett módon végrehajtottuk az A mátrix generálását, majd annak segítségével generáltunk egy megfelelően hosszú (2000 darabos) véletlen vektorrendszer-t, amit egy M mátrixba rendeztünk.
2. Az A mátrix elejéről megtartottuk a Gauss-elimináció segítségével generált $n \times n$ -es részt, így már garantálni tudtuk, hogy a determinánsok relatív prímek legyenek. Ezután az M mátrixból választottunk egy megfelelő nagyságú szeletet, azaz: az A mátrixba már előzetesen generált, ebből az eljárásból származó pszeudo-véletlen vektorok kerültek. (Ezzel azt szerettük volna elérni, hogy az algoritmus lényegében a környezettől független legyen – ne teljes egészében a MATLAB[®] generátorával dolgozzunk.)

Ritkítás

Már említettük, hogy beállítható ritkítás is az algoritmusban. Most a már említett seed mellett még ezt is alkalmaztuk. Így egy jóval nagyobb vektorrendszer-t használtunk – nevezetesen 2-, 4-, 8-, 16-szoros ritkítással dolgoztunk és teszteltünk.

8. Megjegyzés. *Nem alkalmaztunk ennél nagyobb ritkítást, mert bár teszteltük, de nem hozott érdemi javulást vagy romlást a 16-szoroshoz képest, viszont nagyban megnövelte a futásidőt.*

A ritkítással az egyenletesség tekintetében nem javultak mérhetően az eredményeink, azonban az összefüggőség esetén már más tapasztaltunk.

2.5. Az algoritmussal elért eredményeink a különböző statisztikai teszteken

A vizsgált algoritmust a MATLAB[®] véletlen egészeket generáló rutinjával hasonlítottuk össze.

2.5.1. Kolmogorov–Szmirnov-statisztika (KS) alkalmazása

A tesztek eredményeinek ismertetése előtt emlékeztetünk rá, hogy 2000 vektort generáltunk a szigorúan a pozitív ortánsban elhelyezkedő, 100-as oldalhosszúságú, adott dimenziós kockába.

9. Megjegyzés. Bár 100 futás történt, a táblázatokban tört értékek szerepelhetnek. Ugyanis 100 futás esetén, adott dimenzió és adott törés (kocka éleinek felosztása) miatt 100-nál természetesen több statisztikai próba készült.

Például 3 dimenzióban, 4 törés esetén (minden élen 4 részre bontunk) $4 \cdot 3 \cdot 2$ tesztelés történik: 4 törés az adott koordinátán, 3 dimenziót vizsgálunk végig, és a 2 kimaradó koordinátát kell ellenőrizni. (Általánosságban – ha van törés (t) – $\dim \cdot (\dim - 1) \cdot t$). Azaz a fenti példa esetén ez $24 \cdot 100 = 2400$ tesztet jelent.

Az alkalmazott statisztika feltételei az alábbi módon írhatóak le. Adott X_i koordinátát szeletekre bontjuk, nevezetesen az alábbi átkódolásokat, töréseket vezetjük be:

$$T_{i,2} = \begin{cases} 1, & X_i \in [1, 50]; \\ 2, & X_i \in [51, 100]. \end{cases}$$

$$T_{i,4} = \begin{cases} 1, & X_i \in [1, 25]; \\ 2, & X_i \in [26, 50]; \\ 3, & X_i \in [51, 75]; \\ 4, & X_i \in [76, 100]. \end{cases}$$

$$T_{i,5} = \begin{cases} 1, & X_i \in [1, 20]; \\ 2, & X_i \in [21, 40]; \\ 3, & X_i \in [41, 60]; \\ 4, & X_i \in [61, 80]; \\ 5, & X_i \in [81, 100]. \end{cases}$$

A Kolmogorov–Szmirnov-statisztikát az egyenletesség tesztelésére alkalmazzuk $\forall j \neq i$, X_j koordinátára, minden lehetséges $T_{i,k}$ mentén, ahol $k \in \{2, 4, 5\}$.

Azaz $KS(T_{i,k}, X_j)$, ahol $1 \leq i, j \leq \dim$, $i \neq j$ és $k \in \{2, 4, 5\}$. A Kolmogorov–Szmirnov-statisztika első koordinátáján a bontásban résztvevő koordináta szerepel, míg a második koordináta mutatja, hogy mely koordináta egyenletességét szeretnénk tesztelni. A vektorrendszerünk dimenzióját most \dim jelöli. 100 futás százalékos eredményeit a következő táblázatok tartalmazzák.

MATLAB [®]					
		Perem	törés = 2	törés = 4	törés = 5
2D	Nincs ritkítás	0	0	0	0
2D	Ritkítás = 2	0	0	0	0
2D	Ritkítás = 4	0	0	0	0
2D	Ritkítás = 8	0	0	0.125	0
2D	Ritkítás = 16	0	0.25	0.125	0.3
3D	Nincs ritkítás	0	0	0	0.033
3D	Ritkítás = 2	0	0	0	0
3D	Ritkítás = 4	0	0	0	0.067
3D	Ritkítás = 8	0	0	0	0.033
3D	Ritkítás = 16	0	0	0	0
5D	Nincs ritkítás	0	0	0	0
5D	Ritkítás = 2	0	0	0	0.03
5D	Ritkítás = 4	0	0.025	0	0.01
5D	Ritkítás = 8	0	0.05	0	0
5D	Ritkítás = 16	0	0	0	0.01

1. táblázat. A MATLAB generátorának eredményei a Kolmogorov–Szmirnov-teszten

SAJÁT					
		Perem	törés = 2	törés = 4	törés = 5
2D	Nincs ritkítás	0	0.25	0	0
2D	Ritkítás = 2	0	0	0	0.2
2D	Ritkítás = 4	0	0.5	0.125	0.2
2D	Ritkítás = 8	0	0	0.125	0.1
2D	Ritkítás = 16	0	0.25	0	0
3D	Nincs ritkítás	0	0.583	0.292	0.367
3D	Ritkítás = 2	0	0.167	0.125	0.1
3D	Ritkítás = 4	0	0	0	0.1
3D	Ritkítás = 8	0	0.167	0	0
3D	Ritkítás = 16	0	0	0.125	0.033
5D	Nincs ritkítás	0.6	1.425	1.113	1.01
5D	Ritkítás = 2	1	0.4	0.125	0.06
5D	Ritkítás = 4	0.2	0.025	0.013	0.02
5D	Ritkítás = 8	0	0	0.013	0
5D	Ritkítás = 16	0	0	0.025	0.01

2. táblázat. Saját algoritmusunk eredményei a Kolmogorov–Szmirnov-teszten

10. Megjegyzés. A tesztek – mint az látható – elfogadható eredményre vezettek, ugyanis a célkitűzést, miszerint nagyságrendileg olyan jó eredményeket szeretnénk elérni, mint a MATLAB[®] által használt véletlen egész vektorokat generáló algoritmus, lényegében teljesítettük. (Természetesen nem értük el azt a fajta, lényegében 0 valószínűségű határt, amit a MATLAB[®] produkált, viszont általában az 1%-os küszöb alatt tudtunk maradni.)

Még az 5 dimenziós eset ritkításmentes értékei is a 95%-os határ alatt maradtak (alig lépték át az 1%-ot), bár ezen a ritkítás segített, és 4-es ritkítás után már a MATLAB[®]-hoz hasonló eredményeket tudtunk kimutatni.

2.5.2. A χ^2 -teszt eredményei

A χ^2 -próba eredményeit nem foglaljuk táblázatokba, ugyanis nem akadt fenn sem a MATLAB[®], sem az általunk alkalmazott algoritmus.

2.5.3. A Kendall-féle teszt eredményei

Annyi módosítást alkalmaztunk még, hogy nem vetettünk össze minden sorsolásban minden koordinátát minden más koordinátával, hanem bármely bontás esetén véletlenszerűen választottunk ki 1-1 koordinátapárt, amiket összehasonlítottunk egymással.

Például 5 dimenzió esetén, ha a második koordináta alapján 4 részre bontottuk a generáláshoz alkalmazott kockát, majd azon belül a 3. szeletet teszteltük, akkor hol az első és negyedik, hol a harmadik és ötödik koordináta monotonitását hasonlítottuk össze, és így tovább.

Minden esetben a fent már említett 95%-os szignifikancia-szint melletti konfidencia-intervallumot fogjuk a táblázatokban megmutatni. A konfidencia-intervallumot Jackknife-eljárás [8] segítségével becsültük meg 100 futás eredményei alapján. A ritkítási beállítások itt is a „nincs ritkítás”, 2, 4, 8 és 16 voltak.

Nincs ritkítás	MATLAB [®]	SAJÁT
2D	[-0.0417; 0.0423]	[-0.0445; 0.0503]
3D	[-0.0680; 0.0417]	[-0.0516; 0.0458]
5D	[-0.0413; 0.0372]	[-0.0516; 0.0505]
Ritkítás = 2		
2D	[-0.0407; 0.0463]	[-0.0569; 0.0404]
3D	[-0.0414; 0.0425]	[-0.0528; 0.0557]
5D	[-0.0387; 0.0360]	[-0.0845; 0.0793]
Ritkítás = 4		
2D	[-0.0490; 0.0480]	[-0.0510; 0.0379]
3D	[-0.0431; 0.0381]	[-0.0516; 0.0554]
5D	[-0.0472; 0.0483]	[-0.0491; 0.0586]
Ritkítás = 8		
2D	[-0.0458; 0.0342]	[-0.0476; 0.0470]
3D	[-0.0405; 0.0351]	[-0.0533; 0.0419]
5D	[-0.0664; 0.0417]	[-0.0603; 0.0471]
Ritkítás = 16		
2D	[-0.0418; 0.0384]	[-0.0504; 0.0514]
3D	[-0.0368; 0.0432]	[-0.0510; 0.0500]
5D	[-0.0451; 0.0417]	[-0.0453; 0.0394]

3. táblázat. A Kendall-teszt eredményei

11. Megjegyzés. Mindegyik konfidencia-intervallum tartalmazza a 0 értéket, azaz a vektorokat nézve, koordinátáinként sztochasztikusan domináns pár nem található, illetve szignifikánsan nem mutatható ki, hogy valamely koordinátapár mentén szisztematikusan monoton növekedést vagy csökkenést generálna az algoritmus.

Továbbá megállapítható az is, hogy a konfidencia-intervallumok hossza nem túl variábilis, azaz egyik dimenzió egyik ritkítésének esetén sem tapasztaltunk érdemileg hosszabb vagy rövidebb konfidencia-intervallumot.

2.5.4. Az átrendeztetést és a sorozatteszt eredményei

Az előzetes tesztek 32-szeres ritkítés mellett is elvégeztük, azonban e két teszt futásideje és számításigénye okán csak 8-szoros ritkítésig teszteltük magasabb esetszámon. Látható lesz, hogy a vizsgált dimenziók esetén már a 8-szoros ritkítés is elfogadhatóan jó eredményeket mutat.

Az átrendeztetést eredményei

A χ^2 -próba teszteredményeihez hasonlóan itt sem foglaljuk táblázatba az adatokat, mert minden generált vektorrendszer átment ezen a próbán.

Az átrendeztetés hatására nem változott semmi. Azt tapasztaltuk, hogy az eljárások – 100 futásból – egyszer sem fogtak el egy generált rendszert sem.

A sorozatteszt eredményei

Ez a teszt (lásd [18]) a ritkítási paraméter hatását volt hivatott ellenőrizni. Az eljárás során egy alapmátrixot generálva ritkításonként, több tesztet is végrehajtottunk.

Továbbra is 2, 3 és 5 dimenziós vektorokkal dolgoztunk. 5000 darab véletlen vektort generáltunk 1-1 sorozatban, a ritkítések mértékét változtatva.

Azt tapasztaltuk, hogy a teszten fennakadt futások aránya nem változik (sem a MATLAB[®], sem a mi generátorunk esetén) a ritkítés-paraméter növelésével. Az alábbi két táblázatban ismertetjük, hogy az adott dimenzióban hány sorozatot generáltunk, illetve az adott sorozatszám mellett milyen arányban fogta meg a sorozatteszt a tesztelt algoritmust.

Ritkítés mértéke	2D MATLAB [®]	2D Saját	3D MATLAB [®]	3D Saját	5D MATLAB [®]	5D Saját
Nincs ritkítés	10000	10000	10000	10000	10000	10000
Minden 2.	10000	10000	10000	10000	5000	5000
Minden 4.	5000	5000	1000	1000	1000	1000
Minden 8.	1000	1000	1000	1000	500	500

4. táblázat. A sorozatteszt futásainak száma

Ritkítás mértéke	2D MATLAB [©]	2D Saját	3D MATLAB [©]	3D Saját	5D MATLAB [©]	5D Saját
Nincs ritkítás	5,4%	5,3%	5,4%	5,4%	5,4%	5,2%
Minden 2.	5,4%	5,4%	5,6%	5,3%	5,3%	5,8%
Minden 4.	5,6%	5,2%	6,8%	4,8%	5%	5,1%
Minden 8.	5,7%	4,7%	4,1%	4,8%	4,6%	4,8%

5. táblázat. A sorozatesztben megfogott futások aránya

A táblázatokból kiderül, hogy 1-2 tizedes eltérés van a MATLAB[©] generátora és a saját generátorunk között – ráadásul mindegyik az 5%-os hibahatár környékén mozog (a 6%-ot egy esetben lépi át a MATLAB[©] generátora).

Ez azt jelenti, hogy az általunk alkalmazott generátor ez alapján a teszt alapján sem mutat összességében rosszabb képet, mint a MATLAB[©] által használt általánosan elfogadott véletlen egész vektorokat generáló algoritmus.

3. Multidimenzionális skálázás (MDS)

E fejezetben szeretnénk bemutatni, hogy a többdimenziós skálázás (továbbiakban MDS) klasszikus megoldása, illetve annak egyfajta robusztus változata milyen tulajdonságokkal bír.

E módszerek általános tárgyalása statisztikai megközelítésben mutatja be a problémát, amit mi szeretnénk másfajta, optimalizálási feladatként feltüntetve megoldani és kezelni.

3.1. A klasszikus MDS-elemzés elméleti háttere

Az eredeti feladat a következő módon fogalmazható meg: tegyük fel, hogy adott $X_1, \dots, X_n \in \mathbb{R}^p$ ponthalmaz, melyek között klasszikus értelemben távolságokat mérhetünk, kevésbé kötött módon e távolság hasonlóságként, esetleg különbözőségként is megfogalmazható.

A ponthalmazhoz keresünk olyan $Y_1, \dots, Y_n \in \mathbb{R}^q$ reprezentáns ponthalmazt, melyre teljesül, hogy X_i és X_j , illetve Y_i és Y_j , $1 \leq i, j \leq n$ pontok közötti távolságok rendre megegyeznek.

Feltehető, hogy $q \leq p$, hiszen p dimenzióban nyilván az $X_1, \dots, X_n \in \mathbb{R}^p$ pontok megfelelőek, illetve azok bármely, egybevágósági transzformációval módosított képe is megfelelő.

Fontos megemlíteni azt is, hogy a gyakorlatban felmerülő feladatok esetén az $X_1, \dots, X_n \in \mathbb{R}^p$ pontok koordinátái nem feltétlenül ismertek, gyakran csak a közöttük lévő távolságokat ismerjük, adott esetekben a dimenzió sem ismert. Ezért is kiemelt fontosságúak az alábbi definíciók:

19. Definíció. *Egy D , $n \times n$ -es mátrix távolságmátrix, ha szimmetrikus és*

$$\begin{aligned}d_{rr} &= 0 \\d_{rs} &\geq 0,\end{aligned}$$

$\forall 1 \leq r, s \leq n$ esetén.

A klasszikus MDS-probléma tárgyalása során egy másik fontos definíció az alábbi:

20. Definíció. *Egy D , $n \times n$ -es távolságmátrix euklideszi, ha valamely euklideszi térben van olyan ponthalmaz, melyben a pontok közötti távolságokat D adja meg.*

Más szóval: ha valamely q egész szám mellett vannak olyan $Y_1, \dots, Y_n \in \mathbb{R}^q$ pontok, hogy:

$$d_{rs}^2 = (Y_r - Y_s)^T (Y_r - Y_s)$$

minden $1 \leq r, s \leq n$ esetén.

Az alábbi tétel szolgáltatja a klasszikus MDS megoldásának az alapját.

4. Tétel. ([30], XIII. 2.3 tétel). Legyen $D_{n \times n}$ egy tetszőleges távolságmátrix. Legyen $B = HAH$, ahol

$$\begin{aligned} H &= H_n = I_n - \frac{1}{n} e_n e_n^T, \\ e_n &= (1, 1, \dots, 1)^T \in \mathbb{R}^n, \\ a_{rs} &= \left(-\frac{d_{rs}^2}{2} \right), \\ A &= (a_{rs}), \end{aligned}$$

ahol $1 \leq r, s \leq n$. D akkor és csak akkor euklideszi, ha B pozitív szemidefinit.

Nevezetesen igaz, hogy:

1. Legyen D egy $Z = (z_1, \dots, z_n)^T$ ponthalmaz euklideszi távolságmátrixa. Ekkor

$$\begin{aligned} \bar{z} &= \frac{1}{n} \sum_{k=1}^n z_k = \frac{1}{n} Z^T e_n, \\ b_{rs} &= (z_r - \bar{z})^T (z_s - \bar{z}) \quad r, s = 1, \dots, n, \\ \Rightarrow B &= (HZ)(HZ)^T, \end{aligned}$$

tehát B , amely Z ponthalmaz centrált skalármátrixa: pozitív szemidefinit.

2. **(Megfordítás)** Ha B pozitív szemidefinit mátrix, akkor egy megfelelő konfigurációt konstruálhatunk az alábbi módon.

Legyenek $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_p$ a B mátrix pozitív sajátértékei, $x_{(1)}, \dots, x_{(p)}$ a megfelelő, egyszeres sajátértékek esetén egyértelműen meghatározott ortogonális, alábbi módon normált sajátvektorok:

$$x_{(i)}^T x_{(i)} = \lambda_i.$$

Ekkor az $X = (x_{rs}) = (x_{(1)}, \dots, x_{(p)})$ jelöléssel, az $x_{(r)} = (x_{r1}, \dots, x_{rp})^T$ koordinátájú pontok távolságmátrixa D , továbbá a pontok súlypontja $\bar{X} = 0$, és skalárszorzatmátrixa éppen B .

12. Megjegyzés. Amennyiben B egy Z konfiguráció centrált skalárszorzatmátrixa, akkor $b_{rr}^{\frac{1}{2}}$ a \bar{Z} és z_r távolsága, továbbá $\frac{b_{rs}}{\sqrt{b_{rr} b_{ss}}}$ érték pedig a $(z_r - \bar{Z})$ és $(z_s - \bar{Z})$ vektorok által bezárt szög koszinusza.

Általánosabban is megfogalmazhatjuk az MDS-feladatot, amennyiben nem távolságmátrixok segítségével dolgozunk. Ráadásul a klasszikus értelemben vett megoldásnak ekkor is vannak optimalitási tulajdonságai – mint azt majd a későbbiekben látjuk is.

21. Definíció. Egy $C = (c_{rs})_{n \times n}$ hasonlóságmátrix, ha szimmetrikus és $0 \leq c_{rs} \leq c_{rr}$, $r, s = 1, 2, \dots, n$.

22. Definíció. Legyen C tetszőleges hasonlóságmátrix. Ekkor a $d_{rs} = (c_{rr} - 2c_{rs} + c_{ss})^{\frac{1}{2}}$ elemekből álló D mátrixot a C -ből standard transzformációval nyert D távolságmátrixnak nevezzük.

5. Tétel. ([30], XIII. 2.8) Ha C pozitív szemidefinit, akkor a standard transzformációval nyert D távolságmátrix euklideszi, mégpedig $B = HCH$ centrált skalárszorzatmátrixszal.

A gyakorlatban nagyon gyakran felmerül az a probléma, hogy a c_{rs} hasonlóságokat nem tudjuk természetes módon definiálni. Ekkor célszerűnek mutatkozik a

$$d_{rs} = \frac{K_1}{K_2 + c_{rs}}$$

transzformációval definiálni a távolságot ($d_{ss} = 0$ definícióval), ahol K_1 és K_2 alkalmas konstansok úgy, hogy K_2 olyan legyen, mely mellett $d_{rs} \geq 0$ teljesüljön.

3.1.1. A klasszikus MDS algoritmikus megoldása

A klasszikus MDS-megoldást a következő algoritmus segítségével tudjuk megkonstruálni:

1. D távolságmátrixból, melyről most nem tesszük fel, hogy euklideszi lenne, meghatározzuk az

$$A = (a_{rs}) = \left(-\frac{d_{rs}^2}{2} \right)$$

mátrixot.

2. Definiáljuk a $b_{rs} = a_{rs} - \bar{a}_{r\bullet} - \bar{a}_{\bullet s} + \bar{a}_{\bullet\bullet}$ mennyiségeket, ahol:

$$\begin{aligned} \bar{a}_{r\bullet} &= \frac{1}{n} \sum_{s=1}^n a_{rs}, \\ \bar{a}_{\bullet s} &= \frac{1}{n} \sum_{r=1}^n a_{rs}, \\ \bar{a}_{\bullet\bullet} &= \frac{1}{n^2} \sum_{r,s=1}^n a_{rs}. \end{aligned}$$

3. Megkeressük B k -számú legnagyobb $\lambda_1 \geq \dots \geq \lambda_k$ sajátértékét (a pozitívításuk feltehető, mert ha ez nem teljesül, akkor nincsen k -dimenziós klasszikus megoldás).

Meghatározzuk a hozzájuk tartozó $\mathbf{x}_{(i)}^T \mathbf{x}_{(i)} = \lambda_i$ szerint normált $\mathbf{x}_{(1)}, \dots, \mathbf{x}_{(k)}$ sajátvektorokat.

4. A P_r pontok koordinátái az $X = (\mathbf{x}_{(1)}, \dots, \mathbf{x}_{(k)})$ mátrix $\mathbf{x}_{(r)} = (x_{r1}, \dots, x_{rk})^T$ sorainak elemei, $r = 1, 2, \dots, k$.

Az előzetes tételben szerepeltetett B mátrixnak nem kell hogy ne legyenek negatív sajátértékei. Csak azt várjuk el, hogy legyen legalább k számú pozitív sajátértéke.

Továbbá kimondhatjuk azt is, hogy a $k = q$ feltétel nem szükséges. Persze alapesetben igen, de a gyakorlatban az algoritmus megoldható oly módon, hogy $\forall k \leq q$ megoldást vázolunk. Erről szólnak majd a következő részfejezet optimalitási tételei.

3.1.2. A klasszikus megoldás optimalitási tulajdonsága

Nem tettük fel tehát, hogy a távolságmátrix euklideszi lenne. Már megjegyeztük, hogy nem euklideszi esetben is rendelkezik a klasszikus megoldás egyfajta optimalitási tulajdonsággal.

Ennek megfelelően tegyük fel, hogy D távolságmátrix nem euklideszi. Ekkor a megfelelő B mátrix nem pozitív szemidefinit, μ_i pozitív sajátértékeinek száma legyen q . Tegyük fel tehát, hogy $\mu_1 \geq \mu_q$ azon megfelelő sajátértékek, melyek segítségével definiálhatjuk λ_i skalárokat az alábbi módon:

$$\lambda_i = \mu_i \quad 1 \leq i \leq q,$$

és minden további λ_i legyen 0.

Amennyiben a távolságmátrix euklideszi mivolta fennáll, úgy az alábbiak mondhatók el: tegyük fel, hogy az $X_{n \times q}$ adatmátrixnak D távolságmátrixa euklideszi, tekintsük ekkor az MDS-probléma k -dimenziós klasszikus megoldását.

Tegyük fel, hogy adott egy tetszőleges ponthalmaz, melyek távolságmátrixa $\widehat{D} = (\widehat{d}_{rs})$. Legyen ekkor

$$\Phi = \sum_{r,s=1}^k (d_{rs} - \widehat{d}_{rs})^2.$$

6. Tétel. ([30], XIII. 4.4) Ha D egy X -hez tartozó p -dimenziós ponthalmaznak megfelelő euklideszi távolságmátrix és k fix, egész, $1 \leq k \leq p$, akkor X -nek az \mathbb{R}^p tér k dimenziós altéréiben lévő összes vetülete közül a klasszikus MDS segítségével megtalált ponthalmaz által adott \widehat{D} távolságmátrix elemei minimalizálják Φ -t.

7. Tétel. ([30], XIII. 4.6) Legyen $B = HAH$ eredeti ponthalmaz segítségével definiált, míg \widetilde{B} a reprezentáns ponthalmaz segítségével definiált mátrixok. Legyen

$$\Psi = \sum_{r,s=1}^n (b_{rs} - \widetilde{b}_{rs})^2 = \text{tr} (B - \widetilde{B})^2.$$

Ekkor – D tetszőleges távolságmátrix (nem feltétlenül euklideszi) – fix k mellett a k -dimenziós klasszikus megoldás minimalizálja Ψ -t.

Ebből az látszik, hogy nem euklideszi távolságok esetén, de euklideszi térben ábrázolva a klasszikus megoldás – négyzetes eltéréseket véve, az eredeti távolságok és a reprezentáló ponthalmaz távolságait nézve – optimális megoldásnak tekinthető.

3.1.3. Shepard–Kruskal-algoritmus

Amennyiben nem feltételezzük az eredeti konfiguráció euklideszi mivoltát, sőt, az ábrázolásban is csak azt követeljük meg, hogy a „nagyobb távolságoknak az eredetiben nagyobb távolságok feleljenek meg a reprezentáns konfiguráció esetén is”, akkor a Shepard–Kruskal-algoritmus segítségével adhatunk egy lehetséges megoldást.

Tételezzük fel, hogy létezik egy ponthalmaz δ_{rs} euklideszi távolságokkal, de amit mérni tudunk, az:

$$d_{rs} = \delta_{rs} + \epsilon_{rs},$$

ahol ϵ_{rs} hibátag. Feltehető az is, hogy

$$d_{rs} = f(\delta_{rs}, \epsilon_{rs}),$$

ahol f monoton függvény.

23. Definíció. Legyenek adottak x_1^*, \dots, x_m^* számok. Ezek monoton kapcsolatban vannak x_1, \dots, x_m számokkal (jelölés: $\{x_i\} \underset{\text{mon}}{\sim} \{x_i^*\}$), ha bármely i, j párosra, $1 \leq i, j \leq m$ esetén teljesül, hogy

$$x_i < x_j \Rightarrow x_i^* < x_j^*.$$

13. Megjegyzés. Tegyük fel, hogy $\{x_i\} \underset{\text{mon}}{\sim} \{x_i^*\}$. Ekkor e viszony szimmetrikus, azaz

$$x_i^* < x_j^* \Rightarrow x_i < x_j.$$

24. Definíció. Legyenek x és y valószínűségi változók. Az x azon $g(x)$ monoton nem csökkenő függvényét, mely x összes monoton nem csökkenő függvénye közül az y legjobb becslését adja úgy, hogy

$$E(y - g(x))^2 \rightarrow \min,$$

az y -nak az x -re vonatkoztatott monoton regressziós függvényének nevezzük.

Ezzel összhangban azt mondjuk, hogy az $\{x_i^*\}$ ($i = 1, 2, \dots, m$) számsorozat az $\{y_i\}$ számsorozatnak az $\{x_i\}$ számsorozatra vonatkoztatott legkisebb négyzetes monoton regresszióját adja, ha minimalizálja a

$$\sum_{i=1}^m (y_i - z_i)^2$$

kifejezést azon $\{z_i\}$ számsorozatok közül, melyekre $\{z_i\} \underset{\text{mon}}{\sim} \{x_i\}$.

Az algoritmus leírása

14. Megjegyzés. Legyen $x_{i_1} \leq \dots \leq x_{i_m}$. Legyen y_{i_j} , ($j = 1, 2, \dots, m$) számsorozat. Ha $\{y\}$ számsorozat monoton nem csökkenő, akkor $x_i^* = y_i$, $1 \leq i \leq m$, rendezett a sorozat.

Ha nem, akkor tegyük fel, hogy

$$y_{i_1} \leq \dots \leq y_{i_q}, \quad y_{i_q} > y_{i_{q+1}}, \dots, y_{i_{q+t}},$$

és vagy $q + t = m$, vagy

$$y_{i_q} \leq y_{i_{q+t+1}}.$$

Helyettesítsük az $y_{i_{q+1}}, \dots, y_{i_{q+t}}$ számokat az átlagukkal. Ez természetesen kisebb is lehet, mint $y_{i_{q-1}}$.

Amennyiben a számsor végére érünk, úgy kezdjük újra, mindezt addig folytatva, amíg egy monoton nem csökkenő sorozatot nem nyerünk.

Amikor megállunk, az így nyert sorozatunk lesz x_i^* .

E monoton regresszió felhasználásával a Shepard–Kruskal-algoritmus a következő két lépés változtatása:

1. Azon $\{d_{r,s}^*\}$ számsorozat ($r = 1, 2, \dots, n-1$, $s = r+1, \dots, n$) meghatározása, amely a kezdeti vagy az előző lépésben meghatározott ponthalmaz euklideszi távolságaiból álló $\{\widehat{d}_{r,s}\}$ számsorozatnak az adott D távolságmátrix elemeiből álló $\{d_{r,s}\}$ számsorozatra vonatkozó legkisebb négyzetes monoton regresszióját adja.
2. A $\{d_{r,s}^*\}$ távolságmátrixhoz metrikus (klasszikus) MDS segítségével meghatározzuk az adott ponthalmazt.

Világos, hogy ezen monoton regresszió alapuló eljárás is euklideszi megoldásokon alapul. Munkánk során ezen a ponton szerettünk volna módosítást végezni, és az egész eljárást más megfogalmazásban tárgyalva, más megközelítési módot alkalmazva újraértelmezni.

Ehhez át kellett tekintenünk azt, hogy az alkalmazásunkban milyen osztályozási módszerek, illetve e fenti eljárás milyen típusú általánosításai alkalmazhatók.

3.2. Az MDS néhány alkalmazási területe

Az MDS-eljárás során történeti áttekintést adni nehézkes lenne, hiszen – mint azt majd vázoljuk is – fejlesztéseket nem, inkább újabb és újabb alkalmazási területeket lehet e módszerhez kapcsolni.

A szakirodalom teljes áttekintése lényegében lehetetlen feladat. A teljesség igénye nélkül inkább azt szeretnénk bemutatni e részfejezetben, hogy az eljárást hány különböző tudományterületen lehet sikeresen alkalmazni.

• Közgazdasági

Közgazdaságtani értelemben minden olyan területet ide sorolhatunk, melyben gazdasági – pénzügyi – optimalizálásról, vagy területi jellegzetességekről, gazdasági értelemben vett vállalati viselkedésről, magatartásról értekeztek e módszer segítségével. Teljes piaci viselkedést (tehát vállalatok, vagy akár országok tipizálását), akár egy-egy cég különféle telephelyeinek összehasonlítását is vizsgálat tárgyává lehet tenni.

Az MDS-elemzés közgazdasági területeken zajló kutatásokban (bizonyos esetekben szociológiai jellegű tanulmányok is ide sorolhatók) klasszikus értelemben általában alkalmazható, hiszen a változók folytonos volta (pl. GDP, jövedelmek) megkönnyíti a módszer alapjául szolgáló távolságok definiálását, értelmezését.

Blake és társai kutatási jelentésében [4] az MDS-eljárás klasszikus változatát vásárlói magatartási formák összehasonlítására, elemzésére használják.

Syrquin tanulmányában [47] egy olyan gazdasági kutatást mutat be, ahol az MDS robusztus, monoton regresszió segítségével végrehajtott változatát alkalmazza. Ezen robusztus módszert általában a pszichometriai vizsgálatokban szokás bevetni, ahol a skálaváltozók ordinális változók (ritkébbak a folytonos változók a pszichometriai kutatásokban, mint pl. gazdasági elemzésekben). E tanulmány a robusztus változat gazdasági alkalmazásokban való létjogosultságát mutatja be.

Machado és társai tanulmányukban [25] gazdasági periódusokat és válságokat elemeztek Fourier-transzformációval kombinált MDS segítségével. A módszerüket számos (pl. *S&P500*, *Nasdaq*) tőzsdei adaton tesztelik és értelmezik.

• Geográfiai

Természetesnek tűnik, hogy az MDS-elemzést geográfiai kérdések megválaszolására használjuk, ahol értelemszerűen adott távolságok (földrajzi helyek közötti távolságok) találhatóak. Azonban érdemes tudnunk azt is, hogy itt már nyilvánvalóan előjönnek azok a problémák, amik az általánosításunkat is inspirálták: nem igaz az, hogy minden távolság euklideszi lenne, még egy ilyen területen sem. Gondoljunk arra, hogy egyik városból a másikba eljutni nem mindig, minden körülmények között ugyanannyi idő (sőt, az oda-vissza út sem feltétlenül ugyanannyi idő, vagy akár távolság).

Smallman-Raynor és Cliff tanulmányukban [45] egy orvosi-geográfiai vizsgálatot mutatnak be a kolera földrajzi terjedésére, a XX. század eleji Fülöp-szigeteken tapasztalt járvány elemzésére alkalmazva az MDS-eljárást.

Szintén egészségügyi, biológiai vetülete van a genetikai struktúrák, jellegzetességek geográfiai vizsgálatának, melyre Lessa tanulmánya [23] lehet példa.

Mindkét esetben klasszikus MDS-eljárást alkalmaztak az adatokra – de ez nem annullálja a bekezdés elején felvetett problémát.

- **Orvosi**

Az előző fejezetben két olyan geográfiai problémát ismertettünk, mely felvetette az orvosi alkalmazások lehetőségét. Orvosi alkalmazások esetén elsősorban olyan problémákra kell gondolni, melyekben akár diagnosztikai osztályozásokat lehet elvégezni – azonos vagy hasonló tüneteket adó betegségek keresése, illetve egymástól nagyon eltérő tünetek keresése.

A már említetteken felül diagnosztikai vizsgálatokkal (tehát adott esetben már valóban osztályozási eljárásaként való felhasználással) is találkozhatunk. Erre lehet példa egyik oldalról Rounds és Zevon tanulmánya [40], melyben az MDS-t különböző ráktípusok jellemzésére, csoportosítására használják.

Haruna és Ishii munkájukban [13] szintén diagnosztikára használják az MDS-eljárást – ráadásul annak klaszterezéssel való összehasonlításával – általános diagnosztikai eljárásaként feltüntetve azt.

Mindkét esetben metrikus, tehát a hagyományos MDS-eljárás alkalmazását látjuk, robusztus eljárásokat nem feltétlenül indokolt ezeken a területeken felhasználni.

- **Menedzsment**

Alapvetően gazdasági megfontolásokat lehetne e fejezethez felsorolni, de tágabb értelemben többfajta, előző területekkel (geográfiával pl. logisztikai megfontolások miatt) összekapcsolható, illetve a következőkhöz köthető terület a menedzsment területe.

Robinsontól és Benett-től a munkahelyi viselkedéseket tipizáló, osztályozó tanulmányt említhetünk [39]. Ebben az esetben nyilván gazdasági alkalmazásnak tekinthető, amikor munkahelyi viselkedéseket próbálunk feltárni, hiszen a megfelelő munkahelyi légkör jobb termelést tesz lehetővé – illetve fordítva, a nem megfelelő körülmények rontják a termelékenységet.

Másik oldalról viszont munkahely-típusok is feltárhatók ennek segítségével, gondoljunk csak arra, hogy bizonyos munkahelyeken a monotonitás-tűrés, míg máshol pl. a kreativitás lesz az elsődleges szempont. Ennek mérése, tipizálása hasznos információkkal szolgálhat a munkaerő-piaci szereplők számára.

- **Pszichológiai**

Az előzőekhez kapcsolhatók – tehát a munkahelyi menedzsmenthez – a pszichológiai alkalmazások. Fontos azonban kiemelni, hogy a pszichometriában e módszer alkalmazásáról nem csak cikkek, hanem teljes kézikönyvek is készülnek vagy készültek (pl. [43]).

Geográfiai és pszichológiai együttes alkalmazása van Jaworska és Chupetlovska-Anastasova tanulmányának [17], ahol az Amerikai Egyesült Államok városait vizsgálják MDS segítségével különböző pszichológiai változók mentén.

Mentális problémák vizsgálatára és osztályozására alkalmazzák Griffiths és Kalish az MDS klasszikus változatát [12].

- **Ismeretterjesztés, segédanyagok**

A kézikönyvek mellett több, egyes programokhoz használt, illetve egy-egy programra konkrét példákban bemutatott leírások is fellelhetők a témában, amelyek alkalmazási segédanyagként szolgálnak. Ilyenre példa Giguére dolgozata [10], melyben az SPSS[©] programcsomagban megtalálható összes MDS-változat használatát mutatja be egy pszichometriai probléma feldolgozásán keresztül.

Jól látható tehát, hogy az MDS esetében számos alkalmazási területet tudunk felvonultatni, azonban azt is észre kell venni, hogy a lényegében szűrőpróbaszerűen választott alkalmazások jelentős hányada – lényegében mind – a klasszikus megoldást alkalmazta. Egy esetben a monoton regressziót használó robusztus változattal dolgoztak, azonban általánosításra példákat nem találtunk.

Számunkra az elsődleges célkitűzés az volt, hogy az általunk a következőkben vázolt probléma esetében az MDS általánosításait is alkalmazni tudjuk, hiszen nagyon sok esetben ez lehetne/lenne az adekvát megoldás.

3.3. A többdimenziós skálázás más megközelítése

A fejezet számítási eredményei és felállított modelljei cikkünkben [31] és kutatási riportunkban [32] publikálásra kerültek.

Mint az az alkalmazásoknál is említésre került, egyes helyzetekben az euklideszi távolságok nem megfelelőek számunkra. Ebből következően felvetődik a probléma esetleges más megközelítése.

Ilyen szempontból egyik oldalról érdekes a feladat definiálásának újragondolása, másik oldalról pedig az újragondolt feladat másfajta eszköztárral való megoldása, annak viselkedése, vizsgálata.

Arra teszünk kísérletet, hogy e másfajta megközelítések buktatóit és eredményeit ismertessük.

Az MDS-eljárásban tehát a következő problémára keressük a választ: Adott n darab p dimenziós pont, melyek különbözőségi mátrixa legyen S . Az S mátrixról feltesszük, hogy fődiagonálisában 0-k állnak és szimmetrikus. (A módszert távolságmátrix megadásával definiáltuk korábban, azonban mi szeretnénk ennél általánosabban megfogalmazni az eljárásunkat.)

1. Feladat. *A kezdő feladatunk legyen a következő: keresendő n darab $q \leq 3$ dimenziós pont úgy, hogy az n pont közötti, valamely előre meghatározott távolság szerint vett D távolságmátrix a legkevésbé különbözzék az eredeti S különbözőségi mátrixtól (optimális esetben $S = D$).*

Ezen a ponton – mint azt már a klasszikus esetben is kifejtettük – a reprezentáló ponthalmaz dimenziója általában alacsonyabb, mint az eredeti. Továbbá fontos kiemelni azt is, hogy a feladat kitzűzésekor csak az eredeti S mátrixot ismerjük, azaz nem írjuk elő, hogy az eredeti p dimenzió ismert legyen.

15. Megjegyzés. *Általánosságban $q \leq p$ elvárással is megfogalmazhatjuk a feladatot, és ilyenformán a dimenziószám redukciója is része lehet a feladatnak. Sőt, az általánosítás fenntartása érdekében kell kihangsúlyoznunk, hogy D nem feltétlenül az euklideszi távolság, illetve éppen D távolságmátrix az, melynek szélesebb lehetőségeiért dolgozunk.*

Szándékosan az optimális megfogalmazást használtuk, hiszen a statisztikai eljárásokban való megközelítések esetében általában az a cél, hogy a konstruktumok a legkevésbé különbözzenek, illetve a leginkább hasonlítsanak egymáshoz.

A mi esetünkben más lesz a cél: keresendő $Y_i \in T^q$ pontok, ($i = 1, \dots, n$) úgy, hogy $d_{ij} = d(Y_i, Y_j)$ valamilyen távolság és a feladat a fenti, tehát e $D = (d_{ij})$ távolságmátrixot szeretnénk a lehető legközelebb látni S különbözőségi mátrixhoz.

16. Megjegyzés. *Az előzményekkel összhangban most normában fogalmazunk: tegyük fel, hogy $n = 3$. Ebben az esetben definiálható például az alábbi d : minden pont legyen $Y_i = (x_i, y_i, z_i)$, hagyományos koordinátázás mellett, és*

$$d(Y_i, Y_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + 2(z_i - z_j)^2},$$

amivel azt próbáljuk érzékeltetni, hogy vízszintes irányban a mozgásunk „könnyű”, míg függőleges irányban nagyobb munkába, több energiába kerül elmozdulnunk.

Olyan reprezentánst keresünk tehát 3 dimenzióban a távolságmátrix alapján, melyben ez a rendező elv működik.

Maga a probléma valós, hiszen ha adott egy raktárunk, azon belül általában hasonló a tárgyak mozgatása (oldalirányba viszonylag könnyű, emelni és biztonságosan leengedni már nehezebb).

3.3.1. Optimalizálási feladatként való felírás

Keresendő $Y_i = Y(x_{ki})$ pontok és hozzájuk tartozó ϵ_{ij} , $k = 1, \dots, q$, $i, j = 1, \dots, n$ minden pontpárhoz tartozó „tolerancia” változók.

17. Megjegyzés. *A fenti (x, y, z) koordinátázás helyett az x_1, x_2, \dots, x_q , $q \leq p$ jelölést használhatjuk majd azért, hogy a reprezentáló ponthalmaz dimenzióját is (akár) paraméterként kezelhessük.*

Megoldandó az alábbi feladat:

2. Feladat. Keresendő változóink:

$$x_{ki}, \epsilon_{ij},$$

ahol $i, j = 1, \dots, n, i \neq j, k = 1, \dots, q$, és az x -ek a megfelelő pontok koordinátái, ϵ a megfelelő tolerancia-változók.

A kielégítendő feltételek:

$$\begin{aligned} \epsilon_{ij} &\geq 0, \\ |d_{ij} - s_{ij}| - \epsilon_{ij} &\leq 0, \end{aligned}$$

ahol d_{ij} a fenti x_{ki} koordinátákból származnak, az s értékek előre adottak, míg az ϵ értékeket szeretnénk majd minimalizálni. Feltehető, hogy

$$x_{ki} \geq 0.$$

Ez utóbbi feltétellel a pozitív ortánsban keressük a reprezentáló pontokat, semmi egyéb megkövetéssel nem jár.

Az első feltétel nem igazán igényelhet magyarázatot. A második és harmadik feltétel azt jelenti, hogy a reprezentáló pontok távolsága kellően meg kell hogy közelítse az eredeti pontok távolságát.

Az optimalizálandó célfüggvény:

$$\min \sum_{i=1}^M \sum_{j=i}^M \epsilon_{ij}.$$

A fenti feladat segítségével elérhető, hogy a reprezentáló pontseregünk ϵ tolerancia-határ közelében legyen – távolságok alapján – az eredeti, p dimenziós ponthalmazhoz.

Az eredeti, euklideszi távolságok esetén alkalmazott MDS-eljárásban egy projekció segítségével megoldható a feladat – mint azt már korábban [30] ismertettük – azonban az általános eljárásokban is azt a kérdést feszegetjük elsősorban ezen eljárás kapcsán, hogy mennyire jó az alacsonyabb dimenziós reprezentáns.

A célunk az is, hogy az ϵ változók analizálásával többletinformációkhoz jussunk (pl. mely pontoknál a legnagyobb az összes eltérés, milyen az eloszlása e tolerancia-változóknak, stb.).

8. Tétel. *Tegyük fel, hogy p dimenzió fix. Ha $\forall \epsilon_{ij} = 0$ és a reprezentánsokhoz tartozó távolság (D) euklideszi, akkor a reprezentánsok az eredeti struktúra (S) valamely egybevágósági transzformációk egymásutánjából nyert képe (amennyiben S is euklideszi).*

A bizonyításhoz szükségünk lesz az alábbi lemmára:

9. Lemma. *Vegyünk legalább p darab p dimenziós gömböt (ne legyen közöttük koncentrikus pár). Ekkor e legalább p darab gömbnek 0, 1, 2 vagy végtelen sok közös pontja van.*

Bizonyítás: Konstruktív bizonyítást adunk a lemmára. A végtelen sok közös pont esetén azt lehet mondani, hogy egy alacsonyabb dimenziós gömb lesz a metszet. Ez az eset nyilvánvaló, ha pl. három dimenzióban elképzeljük: két gömb metszete egy körvonal, mely rajta van egy harmadik gömb felületén.

A későbbiekben erre az esetre – a tétel alkalmazásakor – nem lesz szükségünk, így most csak a véges esetekre koncentrálnak.

Ebben az esetben vegyük a következő algoritmust:

1. Írjuk fel a legalább p gömb egyenletét egyenletrendszerként.
2. Az első egyenletet vonjuk ki az összes többi egyenletből.
3. A maradék egyenletek most már lineárisak – így ezeknek vagy van megoldása (ebben az esetben csak ellenőrizni kell, hogy az rajta van-e a gömbön), vagy egyáltalán nincsen megoldása (ilyenkor nyilván nincsen közös metszet).
4. A végtelen sok megoldás esetén, ha a rendszerünk alulhatározott, két eset van.
5. Amennyiben 1 szabad változó van, úgy azt visszaírva az első egyenletbe egy másodfokú egyenletet nyerünk, melynek nyilván 0, 1 vagy 2 megoldása adódik.
6. Amennyiben legalább 2 szabad változónk van, úgy az első egyenletbe visszaírva megkapjuk, hogy milyen alacsonyabb dimenziós, másodfokú görbén helyezkednek el a megoldások.

■

Most rátérhetünk a tétel bizonyítására.

Bizonyítás: A problémára konstruktív bizonyítást szeretnénk adni.

Vegyük az adott távolságmátrixunkat, és tegyük fel, hogy $P_1 \equiv 0$, azaz az első pontunk legyen az origó.

Az első főminor alapján (2×2 -es részmátrix), az abban lévő távolság alapján egy origó középpontú körvonal bármely pontja megfelelne P_2 pontnak. Vezessünk be egy rendező elvet: válasszuk mindig a soron következő, Descartes-féle koordináta-rendszer megfelelő tengelye irányába vett pozitív ortánst addig, amíg el nem érjük a megadott p dimenziót. Azaz ebben a lépésben a körvonalról azt a pontot válasszuk, melynek első koordinátája éppen a megadott $D(1, 2) = D(2, 1)$ érték.

A P_3 érték meghatározásakor vegyük észre, hogy két kör metszete lesz a megoldás, melynél 3 esetet kell elkülöníteni:

1. 0 megoldás van. Ebben az esetben az egész algoritmussal megállhatunk, hiszen ilyen helyzetben rögtön látjuk is, hogy bár az MDS-problémát megoldottuk, és találtunk egy mátrixot, mely pontosan megegyezik az eredeti távolságmátrixszal, nincsen olyan ponthalmaz az euklideszi térben, melynek reprezentánsaiként előállíthatók lennének (a távolságmátrixban szereplő pontok közül 3 nem teljesíti a háromszög-egyenlőtlenséget).

2. 1 megoldás van. Ebben az esetben „kollinearitás” áll fenn, azaz nem növelünk dimenziót (eddig egy szakaszunk van, nem kell a 2 dimenziós eset, tehát nem kell háromszöget meghatározni). Maradunk az első Descartes-koordináta mentén, melynek pozíciója fix, hiszen csak egy megoldás van.
3. 2 megoldás van. Ebben az esetben a következő Descartes-koordinátára koncentrálna azt a megoldást választjuk, ahol a következő Descartes-koordináta pozitív.

A Descartes-koordináták pozitívításának választása egészen addig történik, amíg a szükséges dimenziók száma el nem éri az p -t (látható, hogy nem minden lépésben történik növelés).

E fenti rendező elvvel elérjük, hogy az algoritmus „általában” a pozitív ortáns felé mozdítja a megoldásokat.

Nyilván nem a teljes pozitív ortáns fogja kitölteni (gondoljunk egy tompaszögű háromszögre a két dimenziós térben), de az teljesül, hogy az utolsó lépésben a $p-1$ dimenziós hipersík nemnegatív oldalán lesz az alakzatunk.

Amennyiben $n \leq p$, azaz kevesebb pontunk van, mint amekkora a reprezentáláshoz használt dimenzió – vagy már az összes pontot felhasználtuk és nem értük el p -t, akkor a ponthalmaz egy alacsonyabb dimenziós térben is ábrázolható.

Tegyük fel tehát, hogy felvettünk annyi pontot, hogy a szükséges dimenziószámmal elértük p -t. Világos, hogy ekkor alkalmas $p+1$ pont meghatároz egy szimplexet,

$$P_1, \dots, P_{n+1} \triangleright S_0,$$

melynek egyértelműségéhez elegendő a szimplex oldalhosszainak ismerete.

Ebben az esetben (kihasználva a Descartes-féle koordinátarendszernél alkalmazott nemnegatív ortánsba való elhelyezést) még azt is tudjuk, hogy abban a térrészben helyezkedik el.

Ezek után már csak azt kell tennünk, hogy a következő algoritmust kövessük:

1. Vegyünk egy új pontot, és válasszuk ki a szimplex p pontját. Ehhez legyen I indexhalmaz, melybe a már megtalált pontok indexei kerülnek, azaz

$$\begin{aligned} I &= \{1, 2, \dots, K\}, \quad K = p + 1, \dots, n - 1, \\ J &= \{K + 1, \dots, n\}, \\ I \cup J &= \{1, 2, \dots, n\}, \end{aligned}$$

és így:

$$\begin{aligned} P_j, \quad j \in J, \\ P_{i_1}, \dots, P_{i_p}, \quad \{i_1, \dots, i_p\} \subset I. \end{aligned}$$

2. Ekkor az új $p + 1$ pont

$$P_j, P_{i_1}, \dots, P_{i_p} \triangleright S_j$$

meghatároz egy új szimplexet, mely természetesen egybevágósági transzformációk erejéig meghatározott, azonban a már meglévő szimplexszel vagy szimplexeinkkel nem tudjuk, milyen viszonyban van.

3. A már meglévő, de a mostani szimplexben nem meghatározott pontoktól vett távolsága az újonnan választott pontnak egyértelműen meghatározza az új szimplex elhelyezkedését.

Azaz: nem tudjuk, hogy S_j szimplex hol helyezkedik el, egészen pontosan azt nem tudjuk, hogy P_j pontnak és azon P_i pontoknak, ahol

$$P_i \notin S_j, i \in I,$$

mi az egymáshoz való viszonya. Azonban tudjuk $P_i, \forall i \in I$ és P_j távolságokat, melyek együttesen már pontosan definiálják a két alakzat egymáshoz való viszonyát.

18. Megjegyzés. Szemléletesen ez a következőt jelenti: az újonnan felvett pont a p csúcs által meghatározott hipersíkkal a következő viszonyban lehet:

- (a) 0 megoldás van, a megadott távolságok alapján nem teljesül a háromszög-egyenlőtlenség, leáll az algoritmus.
- (b) 1 megoldás van, a kiválasztott pontunk esetében a háromszög-egyenlőtlenségben egyenlőség van, egyértelmű a megoldás, nem kell választanunk.
- (c) 2 megoldás van, a hipersík 1-1 oldalán. Ebben az esetben a kimaradó pont(ok)tól vett távolsága egyértelműen meghatározza, hogy melyik oldalra kerül.

Az utolsó pontban akkor nem tudnánk dönteni, ha az elhelyezkedéshez szükséges $p + 1$ -edik pontunk beleesik a hipersíkba (ezt azonban kizártuk akkor, amikor olyan $p + 1$ pontot választottunk, melyhez kellett a teljes p dimenzió).

4. Addig folytatjuk, amíg minden pont el nem fogy.

■

19. Megjegyzés. Az algoritmusban vázolt struktúra a pontok sorrendjétől és a tükrözésektől eltekintve egyértelmű, hiszen ha minden lépésben csak 1 megoldás van, akkor ott nyilván egyértelműség áll fenn, 0 megoldás esetén megadja az algoritmus azt a $p + 1$ pontot, mely nem teljesíti a bármely metrikától vagy normától elvárt háromszög-egyenlőtlenséget, két megoldás esetén pedig az adott, p pont által meghatározott hipersíkra való vetítésben különbözik a két megoldás, így ezen vetítések kompozíciója adja a szükséges transzformációt (ez a vetítés persze az 1 megoldásos esetben is beépíthető, csak akkor helyben hagyja a rendszert, hiszen önmagára vetít).

10. Következmény. *Fontosnak tartjuk kiemelni, hogy a bizonyításban vázolt algoritmus automatikus ellenőrzést is tartalmaz – nevezetesen, hogy egyáltalán létezik-e olyan pontsereg, melynek távolságmátrixa az, ami.*

Az ellenőrzés az algoritmusban a háromszög-egyenlőtlenség segítségével történik, mely általánosságban elvárt bármely távolsággal szemben, így nem feltétlenül csak euklideszi térben alkalmazható – bár ekkor az egyértelműség kérdését még vizsgálni kell.

3.3.2. Tesztelési eredmények

Tapasztalatként megfogalmazható, hogy a MATLAB solverének segítségével a pontok alacsony száma esetén az NLP-feladat megoldható, illetve általában sikeresen működik strukturált pont-halmaz esetén. Itt a MATLAB[©] beépített rutinját használjuk az általánosan felírt probléma megoldására.

20. Megjegyzés. *Újításként szerepel az algoritmusban, hogy a hagyományos MDS-hez képest mi nem az euklideszi távolságot használjuk az ábrázolásakor, hanem az ábrázoláshoz használt norma paraméterként változtatható.*

Az első – igen kismintás – tapasztalatok a következő eredményeket hozták:

1. Amennyiben a hagyományos, euklideszi norma szerinti egységnyezet távolságait adjuk meg, úgy a program – véletlen x_0 kezdőpontból indítva a feltételes nemlineáris optimalizálást végző algoritmust – az esetek döntő többségében négyzetet talál (természetesen nem a $(0, 0)$, $(1, 0)$, $(0, 1)$, $(1, 1)$ pontokkal).
2. Magyarországi városokat nyugatról keletre sorrendben felsorolva a távolság-mátrixban, az ország térképe szerint megfelelő pozíciókba helyezi el a reprezentáló pontokat – igaz ugyan, hogy kis számú városra alkalmaztuk egyelőre tesztelési képpen az algoritmust.
3. Nagyobb pontmennyiség esetén (akár 10-20 pontra), hiszen úgy az NLP-feltételek száma már 100 feletti, az algoritmus egyelőre nem talál optimumot, így fejlesztésre szorul az eredeti, induló eljárás.
4. l_1 és l_∞ normák esetén is megtalálja a négyzetet, azonban nyilvánvalóan nem egységnyezetet ábrázolva találja meg a pontokat (a struktúráját látja).

3.3.3. Első tesztelési/kísérleti feladatok

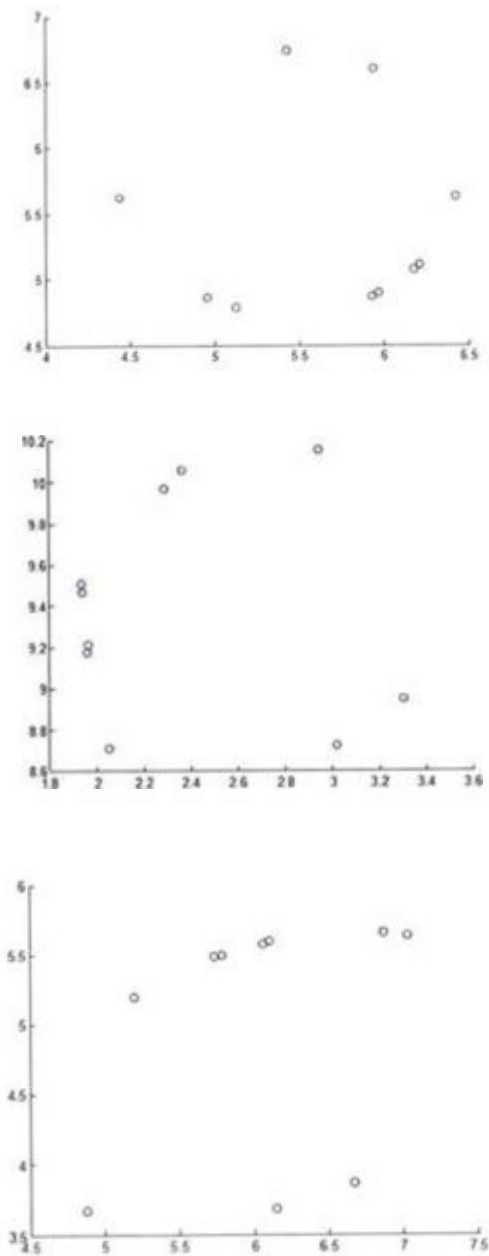
Az algoritmust úgy teszteltük először, hogy a 2 dimenziós egységkör határára generáltunk egyenletes eloszlás szerint pontokat, majd ezek távolságait megadva vizsgáltuk, hogy vajon a körvonalra rendezz-e azokat.

Vizsgálataink eredményeit először külön össze is foglaljuk.

11. Következmény. *Két fontos tapasztalatunk adódott:*

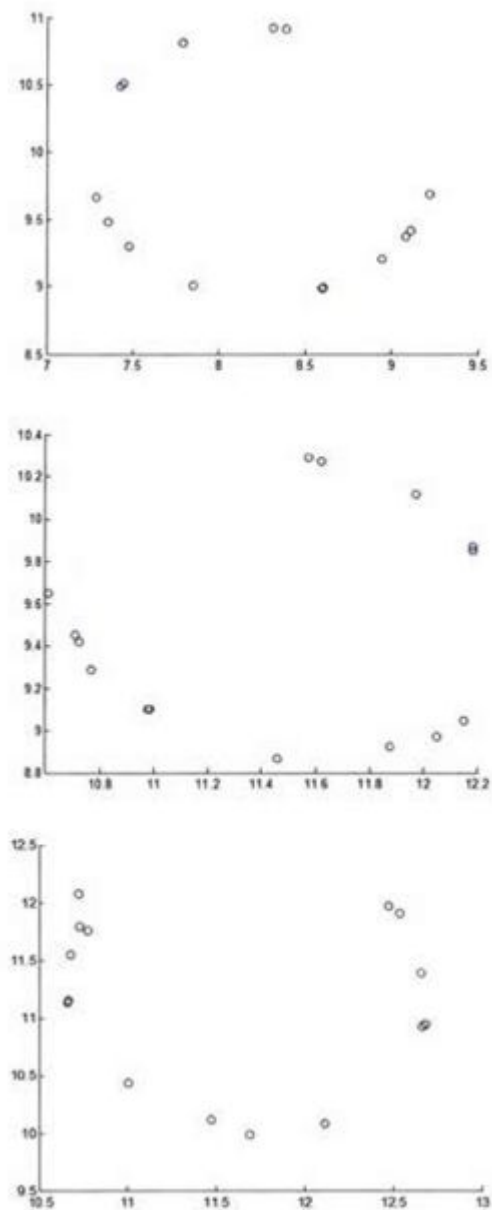
- 1. Mind az euklideszi, mind az l_1 , mind pedig az l_∞ normát alkalmazva lényegében megtalálta a struktúrát a program, azaz az esetek döntő többségében rátalált a globális optimumra, és egy körvonalra rendezte a pontokat. Olykor előfordult, hogy alternatív optimumban ragadt az algoritmus. Ennek kivédésére azt tapasztaltuk, hogy a programot véletlen kezdőpontból indítva, az algoritmust többször futtatva egymás után ez a probléma kivédhető.*
- 2. Az algoritmusunk futásideje a pontok számának növelésével exponenciálisan nőtt mind a három vizsgált norma esetén, továbbá nagyjából azonos futásidőket mértünk mind a három normára.*

E vizsgálatok eredményeit mutató ábrák, illetve a futásidőket tartalmazó táblázat a későbbiekben láthatók. A tesztekhez 10, 15 és 20 pontot generáltunk, melyeket rendre euklideszi, l_1 és l_∞ normában ábrázoltunk. Már 10 pont esetén is kivehető a struktúra mindhárom távolság esetén.



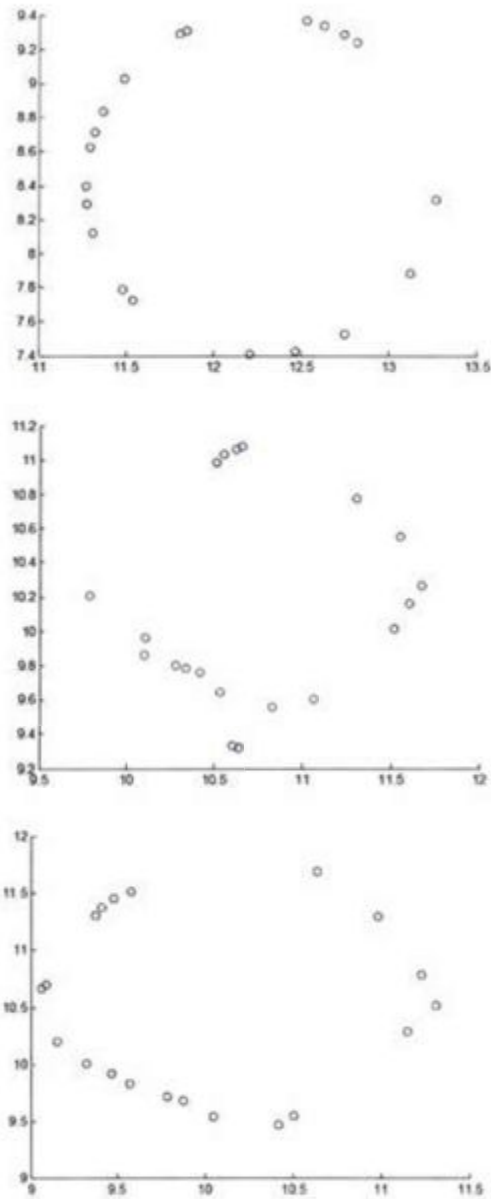
1. ábra. MDS-teszteredmények, körvonalon véletlenszerű 10 pontra – MATLAB[®]

15 és 20 pont esetén egyértelműen felfedezhető az egységkör mindhárom normában – és számunkra a strukturáltság az elsődleges szempont. Más közelítésben: a fenti ábráink alapján nem állíthatjuk, hogy a reprezentáló ponthalmazunk véletlenszerű lenne.



2. ábra. MDS-teszteredmények, körvonalon véletlenszerű 15 pontra – MATLAB©





3. ábra. MDS-teszteredmények, körvonalon véletlenszerű 20 pontra – MATLAB[®]

A későbbiekben is fontos lesz, hogy látunk-e strukturáltságot a pontokban, vagy lényegében véletlenszerűen elhelyezkedő ponthalmazt alkot az MDS által alkotott kép? Az alkalmazásokban ennek nagyobb a szerepe, mint annak, hogy valóban „köröszerű” struktúrát kaptunk-e ezen algoritmus végén.

A futási idők az alábbi táblázatban foglalhatók össze:

Pontok száma / Norma	Euklideszi	l_1	l_∞
10	8,52	11,64	20,74
11	14,71	32,04	18,52
12	10,05	39,67	39,22
13	20,59	40,78	60,48
14	40,80	81,70	85,93
15	76,89	120,20	115,78
16	228,68	154,02	167,22
17	337,62	313,20	308,72
18	300,52	380,56	404,96
19	1093	805,35	660,09
20	1599	859,13	969,23

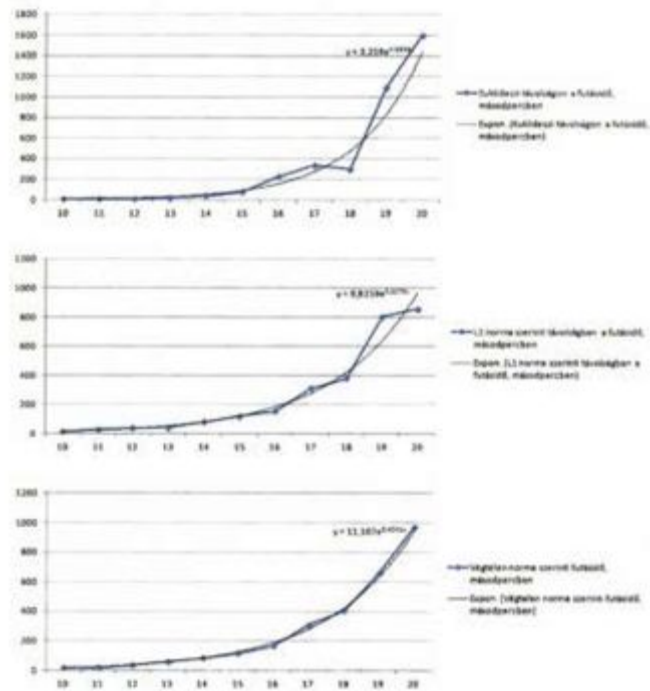
6. táblázat. Futásidők a különböző normákban, másodpercben

Mint az látható, nagyobb esetszám esetén az általános felírás nagyszámú feltétele miatt már lassan dolgozik a rendszer. Az is igaz, hogy általánosságban az MDS-feladatot nem pontok százaira szokás felírni, de már itt látszik, hogy 20 pont esetén – bár az ábrázolás már tetszetős – sok gépidőbe is kerül.

Ráadásul, bár kevés mérési eredményről van szó, nem tévedhetünk nagyot, ha azt állítjuk, hogy a fenti eredmények a pontok számának növekedésével exponenciálisan növekedő idő alatt számíthatók (beleértve azt is, hogy egy idő után a pontok nagyobb száma miatt, a szükséges számítások egyéb igényei miatt már nem lesznek elvégezhetők).

A futásidő exponenciális mivolta a pontok számának növelésére még az euklideszi és az l_1 norma alapján is elég jó illeszkedést mutat – azonban az alább látható l_∞ esetén a leginkább szembetűnő. Az ábrák továbbra is rendre a normák fenti felsorolását követik.

21. Megjegyzés. *Fontos kiemelni, hogy itt nem egy statisztikából jól ismert, regressziós egyenlet meghatározása történik, hiszen minden pontszám-nagysághoz egyetlen futásidőt rendeltünk, azaz az illesztéseink nem statisztikai alapokon készült regressziós egyenleteket takarnak. Az ábrákkal csak érzékeltetni szeretnénk volna azt a tényt, mely az optimalizálási feladatok sajátjaként is felfogható: a változók és/vagy a feltételek számának növekedésével a rendszereket megoldó algoritmusok futásideje exponenciálisan növekszik.*



4. ábra. Exponenciális illesztés a pontok száma alapján az algoritmus futásiidejére – EXCEL[®]

Az illesztett exponenciális egyenletek pedig rendre a fenti felsorolásban:

$$sec = 3,259e^{0,5538(x-9)}$$

$$sec = 9,826e^{0,4173(x-9)}$$

$$sec = 11,107e^{0,4041(x-9)}$$

3.3.4. Második tesztelési/kísérleti feladatok

Második kísérletünk nem a körvonal, hanem a négyzetrács volt, ahol a 2, a 3, majd a 4 élhosszúságú négyzet rácspontjait alkalmaztuk. Ez azt jelenti, hogy a vizsgálandó pontok száma 9, 16, majd 25 volt. Az eredményeket most is euklideszi, l_1 és l_∞ normában vizsgáltuk, azonban a futásidőket most csak feltüntetjük, függvény-approximációt nem fogunk végrehajtani.

A tapasztalatok itt is hasonlóak, mint a körnél. Értelemszerűen meredeken emelkedő futásidő, a három normában (rendre euklideszi, l_1 és l_∞ az általános felírású algoritmust használva) a rácspontokat megtalálja – a két utóbbi normában nyilván torzított formában.

A futásidőket és a készített ábrákat az alábbiakban ismertetem:

Pontok száma / Norma	Euklideszi	l_1	l_∞
9	0,0520	0,0027	0,0047
16	0,0041	0,3835	0,0659
25	0,0380	0,0873	0,3536

7. táblázat. Futásidők a különböző normákban, másodpercben

22. Megjegyzés. *Megjegyezzük, hogy ellentétben a véletlenszerű ponthalmaz esetében, a strukturált megoldásban a futásidők (a pontszámok emelkedésével sem, legalábbis 25 pontig) nem emelkedtek olyan látványosan. A körteszt véletlenszerű ponthalmazára esetén 20 pontra már jelentős számítási időt vett igénybe az eljárás, míg a négyzetrács esetén az emelkedés nem volt jelentős mértékű.*

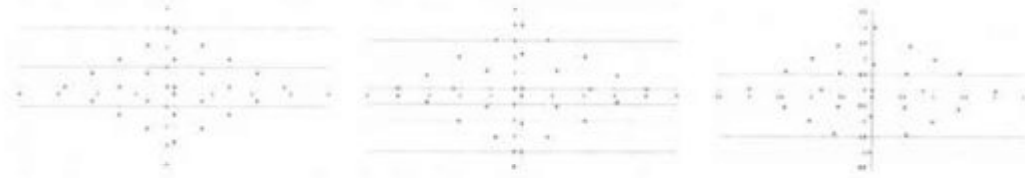
Az alábbi képeken láthatók az eredmények – első oszlopban az euklideszi távolság alapján vett megoldások, majd az l_1 és l_∞ , illetve blokkonként 9, 16, majd 25 ponttal.



5. ábra. MDS-teszteredmények 9 rácspontra – EXCEL[©]



6. ábra. MDS-teszteredmények 16 rácspontra – EXCEL[©]



7. ábra. MDS-teszteredmények 25 rácspontra – EXCEL[©]

3.4. Futásidő-javításra tett kísérlet – specifikálás

A feladatunkban annak nemlineáris mivolta azt eredményezi, hogy az általános megoldó számára nagy lesz a feladat – magasabb változószám esetén kezelhetetlenné téve azt.

Így az irány, melyet magunknak kijelöltünk: két specifikus norma esetén alkalmazunk olyan felírást, mellyel a feladat feltételei linearizálhatók, így elérve, hogy a program könnyebben találjon megoldást – gyorsítva annak működését.

3.4.1. Az l_1 és l_∞ modellek specifikálása, linearizálása

A fenti tapasztalatok azt mutatták, hogy már az euklideszi normával számolt adatok esetén is fennáll: nagyon lassan és olykor pontatlanul dolgozik a rendszerünk, a probléma nagysága és annak általános megfogalmazása miatt. Azonban az euklideszi norma esetén van egy másik fajta megközelítésünk:

23. Megjegyzés. Emlékeztetnénk arra, hogy euklideszi norma esetén a feladat általánosságban megoldott, ott egy alkalmas mátrix legnagyobb sajátértékeinek megkeresésén múlik a feladat, egzakt megoldással szolgálva – négyzetes veszteségfüggvényt minimalizálva. Így arra az esetre a feladat más eszköztárral könnyen oldható.

Hogy esélyt teremtsünk a feladat másik két normában való nagyobb pontszámok esetén is elfogadható idejű megoldására, más megfontolást kell szem előtt tartanunk. Így l_1 és l_∞ normákban elvégezzük a feladat linearizálását. Ekkor lineáris programozási feladatként lehetőségünk lesz arra, hogy ne egy teljesen általános solverrel dolgozzunk, hanem pl. a simplex algoritmussal – lehetőséget teremtve így a nagyobb méretű feladatok általában polinomiális futásidejének elérésére (természetesen nem megfeledkezve arról, hogy a simplex algoritmus nem feltétlenül polinomiális időben szolgáltatja a megoldásait).

Az l_1 norma esete

A lineáris programozási feladat a következő.

1. A változók legyenek – korábbi jelöléseinkkel összhangban:

$$x_{ik},$$

$$h_{ij},$$

$$\epsilon_{ijk}^+,$$

$$\epsilon_{ijk}^-.$$

ahol $i, j \in [1, 2, \dots, M]$ az M pont indexei, míg $k = 1, 2, \dots, N$, azaz a pontok koordinátái, míg h_{ij} skalárookra az l_1 norma kiváltásához lesz definiálva.

A következő definíciókkal éljünk:

25. Definíció. $P \in \mathbb{Z}^{dim^N \times N}$ olyan $-1, 1$ értékű mátrix legyen, melynek van N , azaz eredeti dimenziószámnyi oszlopa és dim^N sora, ahol dim azt a dimenziót jelöli, amely dimenzióban az ábrázolás (közelítés) történik.

P sorai legyenek olymódon megadva, hogy azokban a ± 1 minden lehetséges permutációja legyen felsorolva.

2. Példa. Amennyiben $dim = 3$, úgy

$$\begin{pmatrix} 1 & 1 & 1 \\ -1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \\ 1 & -1 & -1 \\ -1 & 1 & -1 \\ -1 & -1 & 1 \\ -1 & -1 & -1 \end{pmatrix}$$

26. Definíció. Csak a mostani felírás erejéig jelölje z_{ij} az $x_i - x_j$ pontpárok különbségét. Legyen továbbá $e = (1, 1, \dots, 1)$, és legyen $h_{ij} \in \mathbb{R}$ minden pontpárhoz szabadon választott skalár.

2. A feltételek a következők. Az l_1 norma kiváltásához a következő feltételeket kell kielégítenünk:

$$P * z \leq h * e$$

minden szóhajóhető i, j esetén, majd ennek megoldásához kell minimalizálnunk az $\sum_{i,j} h_{ij}$ függvényt. Ekkor ugyanis igazolható az alábbi segédétel:

12. Lemma. *A h_{ij} értékek felülről korlátozzák az adott pontpárok eltérésvektora koordinátáinak összegét, tehát ahol minden, adott pontpárhoz tartozó feltételre teljesül az összes egyenlőtlenség (megengedett megoldásunk van), ott a minimum éppen a két pont l_1 normája.*

Az MDS-feladat megoldásához szükséges, hogy

$$\begin{aligned} h_{ij} &\leq s_{ij} + \epsilon_{ij}^+, \\ h_{ij} &\geq s_{ij} - \epsilon_{ij}^-, \end{aligned}$$

ahol minden, $i \neq j$ indexre teljesülnie kell a fenti két feltételnek.

3. A célfüggvény, melyet optimalizálnunk kell:

$$\min \sum_{i=1}^M \sum_{j=1}^M h_{ij} + \epsilon_{ij}^+ + \epsilon_{ij}^-.$$

Az l_∞ norma esete

A lineáris programozási feladat a következő.

1. A változók legyenek – korábbi jelöléseinkkel összhangban:

$$\begin{aligned} &x_{ik}, \\ &\epsilon_{ijk}^+, \\ &\epsilon_{ijk}^-, \\ &\epsilon_{ij}, \end{aligned}$$

ahol $i, j \in [1, 2, \dots, M]$ az M pont indexei, míg $k = 1, 2, \dots, N$, azaz a pontok koordinátái. Minden pontpárhoz tartozik egy „felfelé” és egy „lefelé” eltérés – erre az eredeti és a megoldás abszolút értékének kivédése miatt van szükség, illetve a maximum kiválasztása érdekében keressük az ϵ_{ij} maximalizáló változókat.

2. A feltételek a következők.

$$\begin{aligned}x_{ik} - x_{jk} &\leq s_{ij} + \epsilon_{ijk}^+, \quad \forall i, j, k, \quad i \neq j, \\x_{jk} - x_{ik} &\leq s_{ij} + \epsilon_{ijk}^-, \quad \forall i, j, k, \quad i \neq j,\end{aligned}$$

továbbá $\forall i, j$, ahol $i \neq j$ indexre teljesülnie kell, hogy

$$\begin{aligned}\epsilon_{ij1}^+ &\leq \epsilon_{ij} \\ \epsilon_{ij1}^- &\leq \epsilon_{ij} \\ &\vdots \\ \epsilon_{ijN}^+ &\leq \epsilon_{ij} \\ \epsilon_{ijN}^- &\leq \epsilon_{ij}.\end{aligned}$$

3. E feltételek mellett az alábbi célt fogalmazhatjuk meg:

$$\min \sum_{i=1}^M \sum_{j=1}^M \epsilon_{ij}.$$

Ezzel a minimalizálással elérhető, hogy a maximum norma szerinti egyenlőségre fogunk törekedni.

24. Megjegyzés. *Ahogy azt már említettük: sok objektum esetén az eredeti felírások nemlineáris volta miatt egyre nagyobb futásidővel kellett kalkulálnunk. A linearizálással a még megoldható feladatok számát szerettük volna növelni.*

3.5. Az MDS alkalmazása: ellenőrzés, visszakövetkeztetés

Az MDS egyik lehetséges alkalmazása egyfajta ellenőrzés (ez adja azt az ötletet, hogy esetlegesen a véletlenszerűség eldöntésére is felhasználjuk).

Az ellenőrzési szerepének bemutatására ismertetjük tanulmányunk [31] idevágó részeit, melyben egyik alapvető célkitűzésünk egy, a jelenkori publikációkban egyre gyakrabban megjelenő, számunkra nyugtalanító jelenségre való figyelemfelhívás volt.

3.5.1. A cikk filozófiai célkitűzése

A kvadratikus hozzárendelési feladat (Quadratic Assignment Problem – továbbiakban QAP) egy jól ismert és igen nehéz feladat. A fent említett cikkben azt vizsgáltuk, hogy teljesül-e az a kritérium, mely szerint ezek a feladatok tényleges, való életből vett problémákon alapulnak, vagy pusztán a megoldóhoz generált, adott esetben minden valós hátteret nélkülöző feladatokon tesztelnek.

Sőt, előfordulhat olyan eset is, hogy egy-egy távolság-mátrixnak mondott mátrixban a háromszög-egyenlőtlenséget sem tudjuk teljesíteni, így már eleve az is megkérdőjeleződhet, hogy van-e létjogosultsága a megoldónak, ha egy olyan adaton teszteljük, mely adat nem is létezik?

A QAP-hez létezik egy QAPLIB könyvtár, melyet 1996 áprilisában hoztak létre és 2002-ig folyamatosan fejlesztettek különböző problémák hozzávételével. A könyvtárban a feladatokat az alábbi struktúra szerint kell keresnünk: irányadó a probléma kitűzőjének neve, a probléma mérete, és ha adott feladatnagysághoz a szerzőnek több példája is tartozik, akkor egy-egy betűvel még külön kódolt maga a feladat is.

Ezen kívül – és ez adta az általánosítás ötletét az MDS-problémához – a feladat kitűzője megadja azt is, hogy a probléma vázolásához milyen távolság-definíciót alkalmaz. Ugyanis azokat az eseteket szerettük volna vizsgálni, amikor egy feladatot képtelenek vagyunk a megadott dimenzióban és a megadott távolsággal visszafejteni – tehát valójában nem egy jól megfogalmazott feladatot látunk, hanem egy „generált példát”.

3.5.2. Az l_1 norma speciális esete

A cikkben szerepel az l_1 távolságra felírt, általánosított MDS-modellünk, 2 dimenziós felírás esetén (az előző fejezetben ezt általánosabban fogalmaztuk meg). E speciális esetben az MDS-feladat az alábbi módon fogalmazható meg.

Legyen n darab pontunk. Legyenek adottak $d_{j,k}$ pontok közötti távolságok, ahol $1 \leq j, k \leq n$.

Ábrázoljuk őket az $[0, h] \times [0, h]$ négyzetben, ahol $h > 0$ az általánosság korlátozása nélkül feltehető. Legyen (x_j, y_j) az ismeretlen j -edik pontunk x és y koordinátái. A h értéket megválaszthatjuk úgy, hogy mondjuk a tapasztalt legnagyobb távolságot választjuk h -nak, hogy minden pontot garantáltan ábrázolni tudjunk.

Azaz

$$h = \max d_{j,k}.$$

Általánosságban ha (a, b) és (c, d) két pontunk, akkor az $l_1((a, b), (c, d)) = |a - c| + |b - d|$ összefüggés az alábbiakban írható át:

$$l_1((a, b), (c, d)) = \max \{a - c + b - d; a - c + d - b; c - a + b - d; c - a + d - b\}.$$

Így az alábbi módon írható le az adott, l_1 távolságot számító feltételrendszer ($1 \leq j < k \leq n$):

$$\begin{aligned}x_j - x_k + y_j - y_k &\leq d_{jk}, \\x_j - x_k + y_k - y_j &\leq d_{jk}, \\x_k - x_j + y_j - y_k &\leq d_{jk}, \\x_k - x_j + y_k - y_j &\leq d_{jk},\end{aligned}$$

továbbá, hogy a reprezentáló pontjaink valóban a kijelölt négyzetbe kerüljenek:

$$0 \leq x_j, y_j \leq h, \quad j = 1, \dots, n.$$

Az első 4 egyenlőtlenség maximuma adja a megfelelő l_1 reprezentáló távolságot.

Ezt ki is tudjuk választani, amennyiben bevezetünk 4 darab bináris változót (továbbra is $1 \leq j < k \leq n$ feltétel mentén maradva):

$$\begin{aligned}x_j - x_k + y_j - y_k + Mu_{jk1} &\leq d_{jk}, \\x_j - x_k + y_k - y_j + Mu_{jk2} &\leq d_{jk}, \\x_k - x_j + y_j - y_k + Mu_{jk3} &\leq d_{jk}, \\x_k - x_j + y_k - y_j + Mu_{jk4} &\leq d_{jk},\end{aligned}$$

ahol u_{jkt} , $t = 1, 2, 3, 4 \in \{0, 1\}$, amennyiben $1 \leq j < k \leq n$.

Itt M megválasztása fontos lehet, de a reprezentáló kocka élhosszának elég nagy skalárszorosa jól működhet (esetünkben $M = 4h$ választás történt).

Ezek után a célfüggvény:

$$\min \sum_{j=1}^{n-1} \sum_{k=j+1}^n (u_{jk1} + u_{jk2} + u_{jk3} + u_{jk4}).$$

Vegyük észre, hogy az azonosan 0 megoldás mindig megengedett megoldása a rendszernek. Az optimális esetben a célfüggvény értéke $3n$ lesz, hiszen egyetlen esetben tudjuk elérni a d_{jk} eredeti távolságokat, a többi esetben az l_1 norma definíciója miatt az adott érték alatt maradunk. A [31] cikkben az MDS-t hasonlósági mátrixokra fogalmazzuk meg, négyzetes veszteségfüggvénnyel.

Fontos kiemelni, hogy ha az MDS egy automatizált programcsomag részeként kerül futtatásra, akkor a programcsomag a minimalizáláson túl a reprezentáló dimenziót is minimalizálhatja. Ezen azt kell értenünk, hogy ha 2 vagy 3 dimenziós ábrázolást kérünk, de a 3 dimenzió nem hordoz érdemi (szignifikáns) többletinformációt a 2 dimenzióhoz képest, akkor érdemesebb a könnyebben átlátható 2 dimenziós reprezentációt alkalmazni.

3.5.3. Visszakövetkeztetésben tett tapasztalati eredményeink

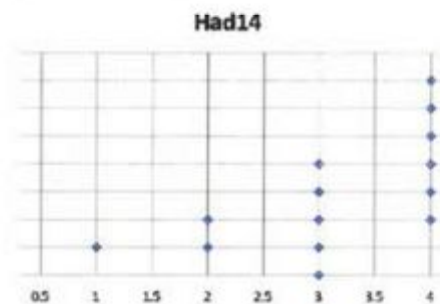
Esetünkben az volt a lényeges szempont, hogy az QAPLIB könyvtárban ismertetett feladatokhoz valóban kapcsolódnak-e (kapcsolódhatnak-e) valós problémák, vagy pusztán egy generátummal állunk szemben?

Ezt a kérdést úgy oldottuk fel, hogy a megadott távolságok és pontszámok, illetve dimenziók alapján a fenti esetben (l_1 távolság esetén a fenti optimalizálási feladatot, míg l_2 esetén a hagyományos feladatot oldva) egy megfelelő dimenziós MDS-megoldást, illetve annak képét vizsgáltuk.

Azt tapasztaltuk, hogy a különböző feladatok másként viselkednek, de egy jól levonható következtetés adódott:

13. Következmény. *Amennyiben egy leírás mind rekonstruálható, mind nem rekonstruálható adatokat is tartalmaz, úgy a módszer csak általános megoldónak tekinthető – és az adott feladathoz nem tekinthetjük specifikus megoldónak.*

A Had14 feladatot 1980-ban oldották meg Burkard és Derigs. Ezen probléma esetén a megadott l_1 távolsággal a reprezentáló pontok távolságmátrixa és az eredeti megegyezett.



8. ábra. A Had14 feladat megoldása l_1 normában – EXCEL[®]

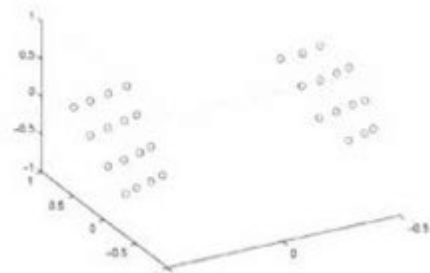
Felmerülhet a kérdés, hogy l_1 és l_2 távolságok közötti ábrázolások esetén mekkora különbségek adódhatnak. Nyilvánvalóan eltérések lesznek a pontok és ábrázolások között nem csak koordinátában, de adott esetben oly módon is, hogy egybevágósági transzformációk erejéig sem ekvivalensek az ábrák egymással.

Azonban ebben az esetben is megmarad a strukturáltság mint közös motívum – tehát a pontsereg reprezentálhatósága ugyan gyengül, de elmondható, hogy valós – vagyis nem véletlenszerűen konstruált vagy generált – feladatok esetén a reprezentációink strukturáltsága szembeötlő lesz.

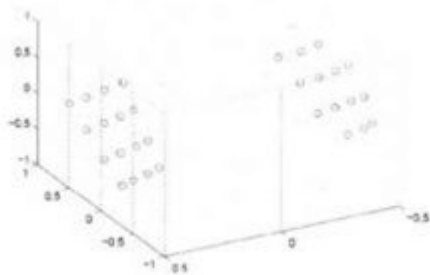
Erre példa a Kra30a feladat 2 és 3 dimenzióban vett, l_1 és l_2 normában egyaránt vázolt képe. Ez esetben két épület 3 emeletének képe kellene hogy kirajzolódjon a leírások szerint. Először az l_1 normában vett, 3 dimenziós képet vizsgáljuk:

Jól látható a 2 épület, benne a 3 szint. Hasonlóan megfigyelhető a struktúra az l_2 távolság esetén is:

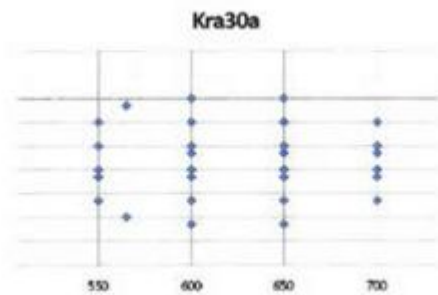
Amennyiben nem az eredeti, hanem 2 dimenzióban ábrázoljuk a pontokat, úgy továbbra is egy szimmetrikus, strukturált esetet fogunk tapasztalni, azonban az illesztése nyilván rosszabb lesz, mint a 3 dimenziós eseté:



9. ábra. A Kra30a feladat megoldása 3 dimenzióban, l_1 normában – MATLAB[©]

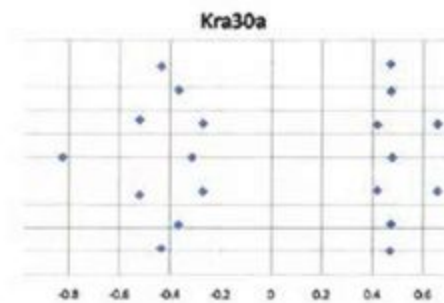


10. ábra. A Kra30a feladat megoldása 3 dimenzióban, l_2 normában – MATLAB[©]



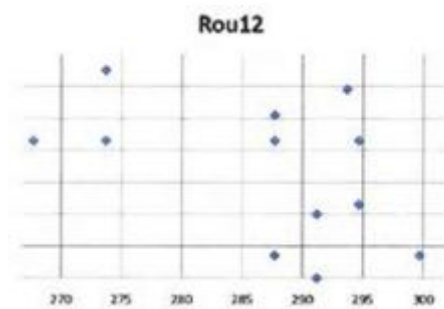
11. ábra. A Kra30a feladat megoldása 2 dimenzióban, l_1 normában – EXCEL[©]

Az l_1 modell esetében jól megfigyelhető egyfajta szimmetria, illetve szabályosság az ábrán, hasonlóan az alábbi, l_2 esethez.



12. ábra. A Kra30a feladat megoldása 2 dimenzióban, l_2 normában – EXCEL[©]

Ezekkel ellentétben a Rou12 probléma esete közelebb áll egy véletlenszerű pontsereg esetéhez, ráadásul az illeszkedés ez esetben nem törént meg l_1 normában. Az l_1 távolságban vett legjobb ábrázolás (hangsúlyoznánk, hogy a rekonstrukció nem volt sikeres) az alábbi volt:



13. ábra. A Rou12 feladat megoldása l_1 normában – EXCEL[©]

3.6. Az általánosított MDS-eljárás tesztelési eredményei

Az MDS-eljárást a következő módszerrel még általánosabban fogalmazhatjuk meg:

- A feltételeket, melyek a reprezentáns pontokra vonatkoznak, valamely l_p távolságban fogalmazzuk meg. Fentebb látható néhány példa, ezek közül az általános az l_2 távolság használata, továbbá linearizáltuk az előzőekben az l_1 és l_∞ eseteket.
- A reprezentáns pontok jóságát a fenti feltételek teljesülésén kívül még a célfüggvényünkön elért minél kisebb érték is mutatni fogja (ha éppen az eredeti pontokat sikerült elérni, akkor a célfüggvény értéke zérus lesz). E célfüggvényt szintén szabadon megválaszthatjuk.

A hagyományos MDS-eljárás az (l_2, l_2) reprezentáció, ahol az zárójelben lévő távolságok közül az első a feltételekre vonatkozik, míg a második a célfüggvény-távolságot mutatja.

Egy kutatási riportban már megjelent dolgozatunkban [32] többfajta felállást is teszteltünk – általánosságban persze, maradva az eddigieknél, az l_1 , l_2 és l_∞ távolságok segítségével dolgoztunk. A teszteket a már fentebb ismertetett esetekben hajtottuk végre: az egységkörön generált véletlen pontrendszeren (a pontok száma 10 és 20 között mozgott), továbbá a 4, 9, 16 és 25 pontot tartalmazó rácsszerkezeten, melyek mind euklideszi térből származó minták voltak. A feltételeket l_1 és l_∞ módszerekkel fogalmaztuk meg, míg a célfüggvény l_1 , l_2 és l_∞ távolságok alapján volt definiálva.

Bemutatjuk az (l_1, l_∞) modell eredményeit a fenti 15 tesztpéldán, továbbá az (l_1, l_1) , illetve a hagyományos (l_2, l_2) esetet úgy, hogy ezen utolsó esetben a reprezentáns pontok közötti távolságokat mind l_1 , mind l_2 , mind l_∞ távolságokat használva definiáltuk.

A számítások elvégzéséhez XPRESS[®] programot használtunk.

3.6.1. Az elvégzett tesztek és eredményeik

Tesztfeladat sorszáma	Pontok száma	Eredeti konfiguráció
1	4	2 × 2-es rács
2	9	3 × 3-as rács
3	10	körvonal
4	11	körvonal
5	12	körvonal
6	13	körvonal
7	14	körvonal
8	15	körvonal
9	16	4 × 4-es rács
10	16	körvonal
11	17	körvonal
12	18	körvonal
13	19	körvonal
14	20	körvonal
15	25	5 × 5-ös rács

8. táblázat. Tesztfeladatok listája az általánosított MDS-eljáráshoz

1. Az (l_1, l_∞) rekonstrukciós modell

Tesztfeladat	Pontszám	Célfüggvény értéke	Optimum
1	4	0	Optimális megoldás
2	9	0,1527	Optimális megoldás
3	10	0,1553	Optimális megoldás
4	11	0,1376	Optimális megoldás
5	12	0,2824	Optimum nem igazolt
6	13	0,3755	Optimum nem igazolt
7	14	0,5534	Optimum nem igazolt
8	15	0,2489	Optimum nem igazolt
9	16	0,8980	Optimum nem igazolt
10	16	0,3477	Optimum nem igazolt
11	17	0,5697	Optimum nem igazolt
12	18	0,5570	Optimum nem igazolt
13	19	0,9543	Optimum nem igazolt
14	20	0,5296	Optimum nem igazolt
15	25	2,7190	Optimum nem igazolt

9. táblázat. Az (l_1, l_∞) rekonstrukciós modell eredményei

2. Az (l_1, l_1) rekonstrukciós modell

Tesztfeladat	Pontszám	Célfüggvény értéke	Optimum
1	4	0	Optimális megoldás
2	9	2,3430	Optimum nem igazolt
3	10	2,3660	Optimum nem igazolt
4	11	2,7473	Optimum nem igazolt
5	12	5,9751	Optimum nem igazolt
6	13	5,8956	Optimum nem igazolt
7	14	11,6358	Optimum nem igazolt
8	15	6,9336	Optimum nem igazolt
9	16	16,5577	Optimum nem igazolt
10	16	10,1937	Optimum nem igazolt
11	17	11,9211	Optimum nem igazolt
12	18	25,6415	Optimum nem igazolt
13	19	12,7347	Optimum nem igazolt
14	20	26,3420	Optimum nem igazolt
15	25	59,8229	Optimum nem igazolt

10. táblázat. Az (l_1, l_1) rekonstrukciós modell eredményei

3. Az (l_2, l_2) rekonstrukciós modell

Ezen modell esetén úgy tekintjük, hogy bár tudjuk az eredeti pontok távolságát, azt nem ismerjük, hogy milyen távolság-definíció alapján számították azt. A modellünkben három távolságot teszteltünk az eredeti pontok távolságának feltételezésekor, ezek rendre:

- *(I)*: az l_1 távolság,
- *(II)*: az l_2 távolság,
- *(III)*: az l_∞ távolság.

A táblázatban az utolsó 3 oszlopban a célfüggvény értéke szerepel a fenti távolságok esetén.

Tesztfeladat	Pontszám	(I)	(II)	(III)
1	4	0,0032	0,0037	0,0029
2	9	0,0029	0,0041	0,0028
3	10	0,0527	0,0042	0,0547
4	11	0,0617	0,0041	0,1239
5	12	0,0528	0,0044	0,0286
6	13	0,0585	0,0041	0,0317
7	14	0,0439	0,0038	0,024
8	15	0,0402	0,0032	0,0422
9	16	0,2204	0,0034	0,0653
10	16	0,0736	0,0037	0,0713
11	17	0,0372	0,0043	0,0242
12	18	0,0413	0,0027	0,047
13	19	0,0388	0,0040	0,0485
14	20	0,0498	0,0027	0,0278
15	25	0,0677	0,0031	0,3569

11. táblázat. Az (l_2, l_2) rekonstrukciós modell eredményei

3.6.2. Az általánosított MDS-eljárás elvégzett számítások tapasztalatai

Az MDS-eljárás során, hagyományos esetekben euklideszi távolságokat alkalmazunk. Két helyen is távolságokat kell definiálnunk: a fenti megfogalmazások alapján részint a reprezentáns pontokra vonatkozó feltételekben, részint pedig az optimalizálandó célfüggvényben.

E két ponton általánosítható az eljárás: l_1 és l_∞ normák esetére linearizálni tudtuk a módszert, és 15 tesztfeladaton meg is vizsgáltuk azt. A tapasztalatok alapján elmondhatjuk, hogy az l_2 távolsággal ellentétben ezen esetekben a solver nem talált optimális megoldást – azonban az eredeti struktúra felismerhető volt.

3.7. Az MDS-eljárás egy másik alkalmazása – munkaügyi kirendeltségek osztályozása

A többdimenziós skálázás egy másik lehetséges – és gyakoribb – alkalmazása egyfajta osztályozás elkészítése. Ezt – az utolsó fejezetben ismertetett munkaügyi alkalmazás előkészítéseként – arra fogjuk felhasználni, hogy a fővárosi munkaügyi kirendeltségeket „tipizálni” próbáljuk.

Ennek kapcsán az úgynevezett „Megegyezőes Eredménycélokkal Való Vezetési Rendszer” – továbbiakban MEV – mutatószámaiból készített kirendeltségi értékeléseket fogjuk alapul venni.

A kirendeltségekről származó adatok nem titkosak, de mindenképpen olyan mutatószámoknak tekintendők, melyek nyers elemei nem feltétlenül publikusak, így minden kirendeltség esetén azt a megoldást választottuk, hogy a mutatók standardizált értékei alapján döntöttünk.

A gyakorlati megvalósítás során az MDS-elemzést átlátható mennyiségű ponton szokás alkalmazni, azaz ritka az az eset, hogy több ezer pontot próbálunk 2 vagy 3 dimenziós térben ábrázolni. Így jogos az a feltételezés, hogy a pontok számát korlátozzuk, és csak alacsonyabb számosságú pontthalmaz esetén próbálunk meg ábrázolásokat készíteni, illetve ilyen esetekben nézzük az algoritmusok futásidejét, teljesítményét – és sikerességét.

25. Megjegyzés. *A statisztikában nem szokatlan az, hogy vizuális tesztekkel döntünk: gondoljunk csak pl. a P-P vagy Q-Q ábrákra, melyek az eloszlásvizsgálatok bevett elemei. Így azt sem zárhatjuk ki, hogy a véletlenszerűség egyik tesztje egyfajta vetületi, vizuális teszt, melyre az MDS-eljárás alkalmas – nyilván a pontok elfogadható számának esetében.*

Fontos kiemelni, hogy az eljárást számtalan mutató kiválasztásával el lehet végezni (jelenleg a kirendeltségeket 150-170 mutató segítségével értékelik).

A következő mutatókat használtuk demonstrációs céllal:

- (1) : Nyilvántartott álláskeresők éves, érintett létszáma
- (2) : Támogatással foglalkoztatottakká váltak száma
- (3) : Képzésbe bekapcsolódók száma összesen
- (4) : Közvetítéssel, nem támogatott foglalkoztatásba kerültek száma
- (5) : Kiáramlás (a nyilvántartásból kivezetett álláskeresők száma)

Minden esetben standardizált eredményekkel dolgoztunk, elérve ezzel azt is, hogy a szempontok között nem tettünk különbséget fontossági sorrendben (hiszen így azonos skálán volt mérve mindegyik változó).

Kirendeltség neve	(1)	(2)	(3)	(4)	(5)
Katona	1,13	-0,49	-0,2	-0,19	0,72
Baross	-1,33	-1,07	0,13	1,33	-0,64
Bajáki	-0,17	-0,54	-0,05	-1,61	-0,13
Vívó	1,74	1,89	0,03	-0,7	-0,66
Bartók	-0,27	1,48	1,5	0,72	-0,38
Heltai	1,28	-0,36	-0,8	1,7	-1,53
János	-0,71	0,02	-2,15	0,56	1,95
Kassai	0,08	-0,71	0,12	-0,2	0,38
Darus	-0,24	-0,56	0,34	-0,51	-0,79
Haller	-0,26	1,05	1,47	-1,03	1,25
PÁDIKI	-1,24	-0,71	-0,39	-0,08	-0,16

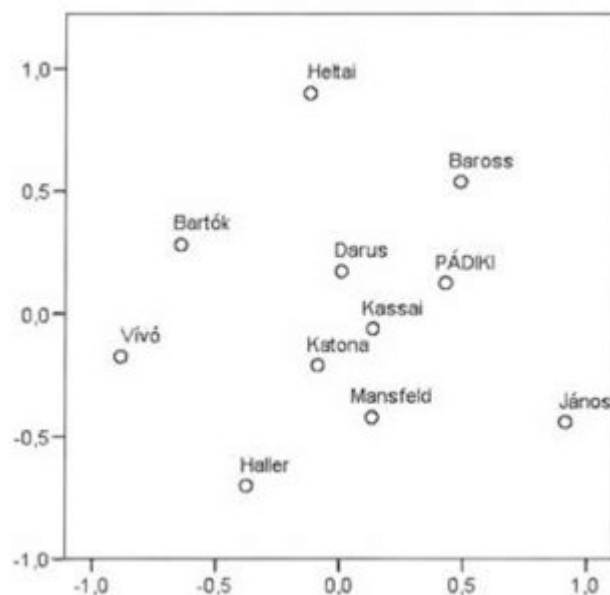
12. táblázat. A mérlegelési szempontok értékeinek standardizált táblázata, kirendeltségenként

3. Példa. Az eredmények értelmezésére egy példa: a Vívó utcai kirendeltség esetén a (2)-es érték: a támogatással elhelyezkedett álláskereső számában az előzetesen tervezetnél átlagosan majdnem két szórással többet teljesítettek.

Egyik oldalról ez jelentheti azt is, hogy átlagosan sokkal jobban dolgoztak – vagy azt is, hogy igen-igen alulbecsült tervezést készítettek, hiszen az általában teljesített tervekhez képest ők lényegesen magasabb értékeket produkáltak.

A PÁDIKI a Pályakezdő és Diplomás Kirendeltség, mely 2012 januárjában megszűnt. A többi kirendeltségből a Bajáki utcai 2012. január 1-jétől Mansfeld Péter utcai kirendeltség lett (nem költözött, hanem az utcát nevezték át).

Az MDS az alábbi elhelyezést találta optimálisnak. (A klasszikus MDS jóságát az s-stress mutató méri, melynek általánosságban 0,05 alatti értéke jelzi az illesztés elfogadható mivoltát. Esetünkben az s-stress mutató 0,03.)



14. ábra. Az MDS alkalmazása munkaügyi adatokra – SPSS[©]

Megfigyelhetjük, hogy egy középső, masszív réteg mellett néhány kirendeltség eltéréseket mutat. A Heltai Jenő téri kirendeltség úgy fest, hogy nagyon különbözik a Haller utcai (hajléktalan) és a János utcai kirendeltségtől, illetve a Vívó utcai és a Baross utcai kirendeltség között is jelentős eltérés mutatkozik e fenti, MEV számok alapján.

26. Megjegyzés. *Fontos kiemelni, hogy jelen esetben nem geometriai alakzatokat keresünk elsősorban: azaz nem a középső ellipszoid, a bal felső kvadráns egyenese és e két mintától teljesen eltérő, egyetlen kilógó, jobb alsó kvadránsban elhelyezkedő pont a lényeges. Egészen pontosan nem mintázatokat keresünk, hanem a pontok közötti távolságokat vizsgáljuk.*

Másik megjegyzésünk: az MDS esetén nem feltétlenül tudjuk a koordinátákat értelmezni – hiszen nem is feltétlenül van közvetlen jelentésük, ahogy itt is nehéz lenne közvetlen jelentést találni a vízszintes és függőleges tengelyeinknek.

- Az első esetben a különbséget jól láthatóan az első és az utolsó mutatóban való eltérés okozza. A Heltai Jenő téri kirendeltségen a nyilvántartott álláskeresők száma átlag feletti, míg a másik két esetben átlag alatti, a kiáramlás azonban a János utca és a Haller utca esetén magas, átlag feletti, míg a Heltai Jenő tér esetében alacsony.

Szintén eltérést okoz a negyedik mutató is, ahol a Haller utca átlagon aluli, a János utca az átlag felett helyezkedik el, azonban a Heltai Jenő tér messze átlag feletti értéket mutat: ez a mutató a közvetítéssel, nem támogatott álláshelyre kerülő álláskeresők számát jelenti.

- A Vívó utcai és a Baross utcai kirendeltség esetében jól megfigyelhető, hogy két érték mentén szinte egyeznek. Ezek a mutatók a képzésbe bevont álláskeresők száma (aránya), ahol lényegében átlagos teljesítményt nyújt mindkét kirendeltség, míg a kiáramlásban átlag alatti mutatókat hoznak.

Amiben lényeges eltérést mutatnak – ellentétes előjellel – az a közvetítéssel, nem támogatott álláshelyre került álláskeresők száma (a Baross utcában átlag feletti magas, míg a Vívó utcában igen alacsony). Továbbá – éppen fordítva – a Baross utcában a nyilvántartott álláskeresők számában és a támogatással elhelyezkedettek számában átlag alatti, míg a Vívó utcában átlag feletti értékek figyelhetők meg.

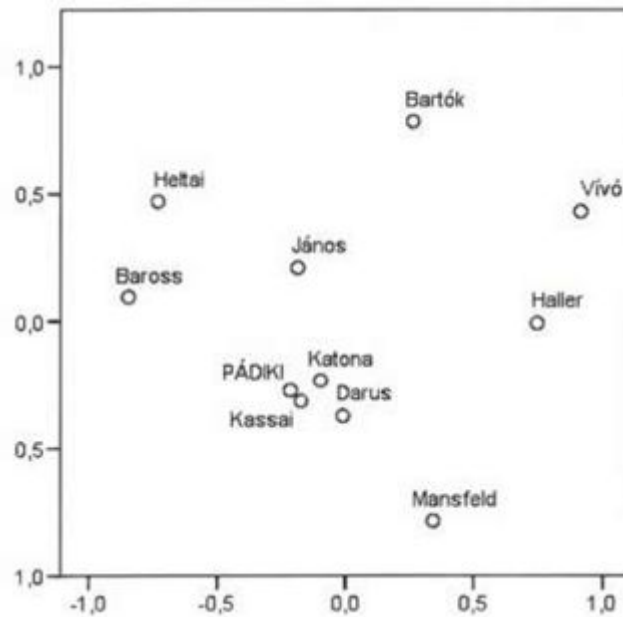
Fontos kiemelni, hogy a mutatók jelentése az, hogy az adott évben a tervezetthez képest milyen elmozdulások láthatók. Azaz: egy negatív eredmény nem feltétlenül jelenti, hogy rosszul működik egy kirendeltség. A tervhez képesti eltéréseket okozhatják pl. hirtelen bejelentett csoportos létszámleépítések, de akár egy nagyobb cég hirtelen munkaerő-felvétele is, mely pl. a képzésektől elmozdítja támogatott munkahelyek irányába a mutatókat.

Minden ilyen eltérés esetében gondosan vizsgálni kell az eltérés okát – és a következő tervezés esetén ezeket be kell tudni építeni.

3.7.1. Csak két koordináta használata – könnyebb interpretáció

Felhasználhatóság szempontjából lényegesen könnyebb – és általában bevett gyakorlat – az ilyen jellegű osztályozások során, hogy csak két koordinátát választanak ki, melyek segítségével könnyebben interpretálhatóvá válnak az eredmények.

Vegyük az előző 5 szempontot, azok közül is mondjuk a (2) és (4) szempontokat. Az első tehát – ez lesz az ábrázolás X tengelye, a támogatással elhelyezett álláskeresők számának tervhez képesti alakulása, míg az Y tengelyen azt mérjük, hogy a támogatás nélküli, kiközvetítéssel tevékenység miként alakult a tervezetthez képest.



15. ábra. Az MDS alkalmazása két koordinátára – SPSS©

Figyeljük meg, hogy a 4 ortánsnak most könnyebben magyarázható jelentést tudunk adni – így maga az eljárás is erősebben használható.

- A bal alsó ortánsban olyan kirendeltségek helyezkednek el, amelyek mindkét szempontot alultervezték (az átlaghoz képest). Ilyen pl. a Bajáki utcai kirendeltség.
- A bal felső ortánsban azok vannak, akik a támogatásokat felül-, míg a közvetítéseket alultervezték. Ilyen pl. a Baross utcai kirendeltség.
- A jobb alsó ortánsban tartózkodó kirendeltségek pont fordítva: a támogatásokat alul-, míg a közvetítéseket felültervezték. Ilyen pl. a Haller utcai kirendeltség.
- A jobb felső ortánsban pl. a Heltai Jenő téri kirendeltség tartózkodik: az ő esetükben az mondható el, hogy mindkét tervet alul-tervezték (igaz, a támogatásokat majdnem jól tervezte a Heltai Jenő téri, és e kirendeltségen a közvetítések sem lettek túlságosan alultervezve).

Az ábrából az is leolvasható, hogy a középen található 4 kirendeltség (ideérthetjük a Heltait is, vele együtt 5) nem tervezte e két szempontot nézve érdemben rosszul, hiszen általában az átlagos tervet hozták – azaz e kirendeltségek összességében valóban centrumot képeznek, hiszen az átlagot nem úgy teljesítik, hogy körülötte hatalmas szórásokkal egyes kirendeltségek borzalmasan alul-, míg mások borzalmasan felülteljesítenek, hanem van egy masszívnak festő mag, akik centralizálják az eredményeket.

3.8. Konklúzió, célkitűzés

Előfordulnak olyan esetek, amikor specifikus problémák megoldó rutinjait általános megoldóként szeretnénk használni (hibásan). Egyre gyakoribb, hogy általános megoldókhoz specifikus problémákat készítünk, hogy ezáltal azok alkalmazása létjogosultságot nyerjen. Olykor ezt olyan példákon tesszük, melyek nem a való életből származnak, sőt: eljárásunk tesztelése során olyan eseteket is feltártunk, amikor egy adott feladatnak a rekonstruálása, való életbeli interpretálása lehetetlen, megkérdőjelezve annak valódi életből vett előnyét – és megkérdőjelezve így az egész eljárás létjogosultságát.

Tapasztalataink szerint ez egy létező gond, és az MDS segítségével sikerült bemutatnunk [31] cikkünkben olyan helyzetet, amikor a feladat kitűzése, meglátásunk szerint nem származhatott valós helyzetből.

Az MDS-t többféle módon is általánosítani tudtuk, aminek a segítségével olyan helyzetekre is alkalmazhatóvá tettük az eljárást, amikor az euklideszi távolságot nem tudjuk a pontok között garantálni.

Ezen túlmenően sikeresen alkalmaztuk az eljárást egy olyan helyzetben, amikor vezetői döntéseket kellett megalapoznunk különböző mutatók segítségével. Az olykor döntést nehezítően sok mutatót alacsonyabb dimenzióban reprezentálva jelentős segítséget tudunk nyújtani egy megalapozottabb döntés meghozatalában.

4. Hierarchikus pénzosztási feladat – Alkalmazás

4.1. Kérdésfelvetés

A bevezetőben is említetteknek megfelelően most vázoljuk azt a közigazgatási feladatot, melyben az első két fejezetben ismertetett eljárásokat alkalmaztuk. A most bemutatásra kerülő projekt kapcsán tudunk számos fejlesztésbe belekezdeni – többek között az itt bemutatásra kerülő alkalmazás megírásába is. A feladat leírását és az eredményeket a Munkaügyi Szemle Stúdium rovatában 2012 szeptemberében megjelent cikkünk [50] mutatja be.

A közép-magyarországi régióban, a Regionális Munkaügyi Központ felügyeletével 2011 közepén elindult egy Európai Unió által támogatott projekt [54]. Ennek tervezésekor hasznosnak bizonyult, hogy a felmerülő, előre nem kalkulálható mennyiségek miatt szimulációk segítségével többletinformációkhoz juthattunk a tervezési folyamatban. A projektben a később majd ismertetésre kerülő álláskeresők célcsoportjaira fókuszáltak, számukra próbáltak alternatívákat, segédeszközöket kínálni. Mivel már a projekt elején kezdetét vette a közigazgatás átszervezése, ezért a számításokat nem az egész régióra, hanem csak a fővárosi illetékességű kirendeltségekre fogjuk bemutatni (a projekt a régió két központjában folytatódik tovább megosztott költségvetéssel).

Maguk a problémák egyes projektek kapcsán nyilvánvalóan többrétűek, melyek egymással is szoros összefüggésben lehetnek. Ilyen többek között a felhasználható pénz kirendeltségek közötti elosztása – vagy más megközelítésben a pénzből finanszírozandó álláskeresők különböző eszközökre történő irányítása annak figyelembe vételével, hogy az eddigi projektekben a különböző eszközöknek milyen sikerességi rátái voltak. Ezek segítségével megpróbálhatjuk előrejelezni, hogy az adott térségben (figyelembe véve, hogy a térségek munkaerőpiaci helyzete ritkán változik heti/havi gyakorisággal) várhatóan milyen költséggel jár egy-egy eszköz alkalmazása.

Általában az EU-támogatott projektek esetén megtérülési ráták betartásával lehet a pénzeket büntetés nélkül felhasználni. Amennyiben a megtérülési rátákat nem tudja egy projekt teljesíteni, úgy a kiutalt pénz visszafizetésére kötelezhető a projektet koordináló hivatal, intézmény vagy szervezet.

Esetünkben bizonyos – a későbbiekben ismertetésre kerülő – célcsoportok és előre meghatározott eszközök kerülnek szétosztásra. A célcsoportok valamilyen szempontból munkaerőpiaci részvételükben korlátozottak, hátrányos helyzetűek. Az aktív eszközök általános jellemzője, hogy valamilyen mérhető teljesítmény áll mögöttük, így egyéni elbírálás szerint adható az álláskeresőknek (a járadék típusú ellátások passzívak olyan értelemben, hogy a jogi feltételek teljesülése esetén automatikusan jár az azt igénylő álláskeresőnek). A célcsoportok bevonásának, arányának meghatározása nem tartozott a matematikai modell keretei közé, hiszen ez nem kimeneti mutatója a modellnek – azonban korlátozó feltételként felhasználásra kerülnek ezek az ismereteink.

A számunkra fontos kérdés az, hogy az egyes kirendeltségekről – az eddigiekben tapasztalt megtérülési ráták alapján, illetve a kirendeltségenkénti célcsoportok feltételezhető nagyságából, mértékéből kiindulva – az egyes aktív eszközökre milyen létszámú álláskereső toborzása javasolt.

A sikerességi ráták nem célok, hanem korlátozó feltételek lettek, míg az elérendő cél az olcsó működtetés volt. Fontos kiemelni, hogy – mint azt majd látni fogjuk – a sikerességi ráták alapvetően alsó korlátok (tehát minimálisan elérendő létszám-előírások szerepelnek a szerződésekben), így érthető módon az azonosan 0 megoldás nem szolgáltatathat minimális költségű megengedett, tehát optimális megoldást.

4.2. Specifikumok, a közigazgatási környezet

A feladat megoldásához definiálnunk kell, hogy mely kérdések azok, melyekre válaszokat várunk, illetve melyek azok, amiket befolyásolni tudunk egyáltalán. Fontos meghatározni, hogy ezen változókra vagy változók kombinációira milyen korlátozó feltételek vonatkoznak, illetve a változók milyen célfüggvényét szeretnénk optimalizálni a feladat megoldása során.

A közigazgatási környezet

A kérdéses változók halmaza az, hogy kirendeltségenként az adott eszközökre milyen számú álláskereső kerüljön.

Ezen változók különböző kombinációira vonatkoznak a korlátozó feltételek, hiszen részint egyes kirendeltségeken is összefügg egymással a különböző eszközök mennyisége (nem korlátlanok az erőforrások), részint pedig az erőforrások egymással is összefüggnek: pl. a képzőhelyek száma is korlátozott, így összességében sem lehet végtelen sok embert képzésre irányítani.

Továbbá a korábbi tapasztalatokból kiindulva az egyes kirendeltségek – az ottani lakossági összetétel és egyéb befolyásoló tényezők hatására – más és más megtérülési rátával bírnak, tehát ezt is figyelembe kell majd vennünk a tervezéskor. Ez bővebben annyit jelent, hogy olyan kirendeltségekre, ahol az eddigi sikeresen befejezett képzések aránya igen alacsony volt, lehetőség szerint ne tervezzük képzések indítását nagy létszámban – hiszen a várható megtérülési ráta alacsony lesz.

Helyszínek, eszközök és résztvevők

A fenti változókat pontosítani fogjuk. Felsorolásra kerülnek a vizsgált kirendeltségek, illetve azok az aktív eszközök, melyekre a pénzosztás (álláskeresők toborzása) a projekt keretein belül megvalósul.

• Kirendeltségek

A fővárosban e fejezet feldolgozásakor 10 munkaügyi kirendeltség működik. A kirendeltségek között a fővárosban egy tematikus kirendeltség is helyet kap, általában azonban a kirendeltségek helyi, területi felosztás alapján szerveződnek.

A kirendeltség illetékessége vagy kerületre, vagy szerepre vonatkozik. Ezeket tüntetjük fel majd egy későbbi táblázatban.

Ebben látható, hogy a főváros esetén a hajléktalanokkal külön kirendeltség foglalkozik. Minden esetben az adott kirendeltség „címe” a kirendeltség megnevezése. Országos szinten pl. a Nemzeti Foglalkoztatási Szolgálat honlapján ([55]) lehet tájékozódni az adott lakóhelyek szerinti illetékes kirendeltségről.

Kirendeltség kódja	Kirendeltség megnevezése	Kirendeltség illetékessége
0102	Katona	VI., XIII.
0106	Baross	V., VII., VIII.
0109	Mansfeld	XXI.
0110	Vívó	X., XVI., XVII.
0112	Ráday	IX., XI., XII., XXII.
0113	Heltai	I., II., III.
0114	János	XX., XXIII.
0115	Kassai	IV., XIV., XV.
0116	Darus	XVIII., XIX.
0118	Haller	Hajléktalan

13. táblázat. A fővárosi munkaügyi kirendeltségek kódjai, megnevezései és illetékességei

• Aktív eszközök

A munkaügyi központok az álláskeresők ellátásában többfajta eszközt vehetnek igénybe. Ezek egyik fele passzívnak tekinthető olyan értelemben, hogy a központoknak (vagy kirendeltségeknek) vagy nincsen elbírálási joguk, vagy annak igénybe vételéhez, folyósításához az álláskeresőnek nincsen „érdemi” tennivalója. Ilyen pl. az álláskeresési járadék, mely alanyi jogon jár – amennyiben a jogi feltételek (pl. szükséges időtartamú munkaviszony) fennállnak. (Bár jogi szabályozásban ennek folyósítását kötelezettségekkel is bővítették, mégsem tekinthető aktív eszköznek, mint az majd látszódní fog.)

Vannak olyan eszközök, melyek igénylése során a központoknak mérlegelési jogkörük van, és az álláskeresőnek munkát (vagy tanulást) kell a jogosultság kihasználásához végeznie (vállalkozóvá válási támogatás, bértámogatás, képzések). Ezen utóbbi eszközöket nevezzük aktív eszközöknek.

Az aktív eszközök korlátozott mértékben állnak a központok rendelkezésére, illetve időközönként kiegészülnek a keretek EU-s forrásokkal. Ezekhez pályázatok útján juthatnak a központok, külön célcsoportok és eszközök megjelölésével – megtérülési ráták vállalásával. Ilyenkor az eszközökre adott pénz szintén korlátozott, azonban a teljesülési ráták elérése esetén vissza nem térítendő.

A bemutatásra kerülő projekt-feladatban több aktív eszközt is elérhetővé tettek az álláskeresők számára, ezek az alábbiak:

1. *Képzés*

Felnőttképzési, a projektben jellemzően OKJ-s végzettséget adó képzési formák (kiemelt szakképesítések), ahol a képzési díj jelentős hányadát átvállalja a projekt – azonban önhibából fakadó kimaradás esetén a támogatási összeg visszatérítésére kötelezhető a résztvevő.

A célcsoportnak megfelelően az elavult végzettséggel rendelkezők jellemzően ilyen képzéseken, átképzéseken vesznek részt.

2. *Vállalkozóvá válási támogatás*

Azon álláskeresők esetében, akik a munkaerőpiacra önerőből, valamilyen vállalkozás beindításával kívánnak visszatérni, az álláskeresési járadék továbbfolyósítása válik lehetségessé fél éven keresztül, továbbá a vállalkozáshoz szükséges egyéb kiadások egy részét is átvállalja a munkaügyi szervezet (irodabérlés, képzési költségek stb.).

3. *Bértámogatás*

Munkáltatók számára alkalmazott támogatási forma, ahol abban az esetben, ha regisztrált és törvényben meghatározottan veszélyeztetett (hátrányos helyzetű) álláskeresőt alkalmaznak, meghatározott időtartamig a bér, illetve a járulékok egy részét a munkaügyi szervezet állja – cserébe az álláskeresőt például ugyanannyi ideig továbbfoglalkoztatás illeti meg (szerződés kérdése is lehet), azaz a munkáltató nem bocsáthatja el, máskülönben a támogatás visszafizetésére kötelezendő.

4. *Bérvisszatérítés*

A bértámogatáshoz igen hasonló támogatási forma, azonban ez csak munkaerő-piaci program keretén belül – viszont nem csak a törvényben meghatározott veszélyeztetett csoportoknak – adható. Támogatási feltételeit rendelet szabályozza.

A vizsgált projekt egyik jelentős eleme, hogy a bevonandók 60%-ának mindenképpen képzésben kell részt vennie, megadott sikeres befejezési rátával, ráadásul úgy, hogy a képzés/támogatás lezárása utáni 180. napon munkaviszonnnyal kell rendelkeznie. Ezeket a korlátozó feltételek során még ki fogjuk fejteni.

• **Célcsoportok**

Bár közvetlenül nem fogunk velük számolni, didaktikailag mégis fontos feltüntetni a résztvevőket (a velük kapcsolatos információk alapvetően inkább a korlátozó feltételekben fognak megjelenni, közvetlenül nem érinti őket egyetlen számítási eredményünk sem).

1. *Szakképzettelenek:* a szakképzettelen álláskeresők azok, akiknek egyáltalán nincsen szakképesítésük, illetve azok is, akiknek bár van szakképesítésük, de az elavult vagy hasznavehetetlen a jelen munkaerő-piaci helyzetben.

2. *Pályakezdők és/vagy 25 év alattiak:* a pályakezdő csak 25 éves kor alatti álláskeresőt jelenthet, azonban nem minden 25 év alatti minősül pályakezdőnek, hiszen annak feltétele még, hogy 2 évnél kevesebb munkaviszonnyal rendelkezzen. 25 éves korig lényegesen hosszabb munkaviszonyt is el lehet érni, így e két kategóriát összevonva kezeli a projekt.
3. *50 év feletti:* az egyik leginkább veszélyeztetett álláskereső-kategória, hiszen koruknál fogva nehezebben találnak munkát – már általában nehezebben is mobilizálhatóak. Esetükben fennállhatnak olyan tényezők is, melyek a mostani munkaerő-piaci helyzetben rájuk nézve fokozottan hátrányosak (informatikai háttértudás hiánya, idegen nyelvek ismeretének hiánya), azonban ezek jól irányzott képzésekkel viszonylag rövid idő alatt pótolhatók.
4. *GYES-ről vagy egyéb ellátásról visszatérők:* nagyon gyakran fennáll az ilyen helyzetekben az előzőek valamelyike is: fiatal, netán a szakképesítése szorulhat korszerűsítésre. Azonban fontos azt is megjegyezni, hogy mind a projektben, mind az egész szervezetben, bár kiemelt helyen szereplő célcsoport, létszámuk igen-igen csekély a többi célcsoport méretéhez képest.
5. *12 hónapnál régebbi, tartós álláskeresők:* esetükben nagyon gyakran fennáll a szak-képzetlenség vagy az 50 év felettiakkal vett metszet: visszatérésük a munka világába talán a legnehezebb az összes célcsoport közül – és általában ők lesznek a leginkább rászorulva valamilyen segítségre.
6. *Bérpótló juttatásban vagy Rendelkezésre állási rámozgatásban – együttesen: Foglalkoztatást helyettesítő támogatásban – részesülők:* már rendelkeznek valamilyen támogatással – jellemzően valamely előző projekt keretéből. Náluk általában a folytonosság fenntartása a cél.

A projekt célcsoportjai tehát nem diszkrét halmazok, közöttük nyugodtan lehetnek átfedések: miért is ne lehetne szakképzetlen pályakezdő valaki, aki éppen GYES-ről próbál meg elhelyezkedni?

4.3. A matematikai modell alakja

A fentiek alapján meghatározandó tehát, hogy adott kirendeltségen, adott eszközre a célcsoportokból összességében hány főt kalkuláljunk. A fenti ismérvek kapcsolatát elsősorban a korlátozó feltételekben tudjuk tetten érni. Ezeket a korlátozó feltételeket részint a létszámokra, részint pedig az elérendő teljesülési rátákra fogjuk megszabni.

4.3.1. Korlátok

1. A célcsoportok automatikus felső korlátot jelentenek, azonban ezek becsléses mivolta miatt $\pm 10\%$ ráhagyással dolgozunk minden kirendeltség elérendő létszámának esetén.
2. Alsó korlátként alkalmazandó a támogatásban foglalt irányelvek teljesítéséhez szükséges elérendő mutatók értéke, amik a bevonandó létszámokról, elérendő sikeres eszközfelhasználásról szólnak.

4.3.2. Az optimalizálandó célfüggvény

Minden aktív eszköznek megvan a maga ára: egy képzés esetén ki kell fizetni a részvételi díjat, egy vállalkozóvá válási támogatásnál magát a támogatást, munkáltatóknál az átvállalt bért, minden egyéb esetben pl. a kapcsolódó szolgáltatások árát, stb.

Feladatunk, hogy a korlátozó feltételek mellett a lehető legolcsóbb módon tudjuk a projektet üzemeltetni oly módon, hogy az előírt sikerességi rátákat teljesítsük.

A projektben természetesen felmerülnek fix költségek (a projekten dolgozók bére, rezsiköltségek), ezeket azonban előre lehet tervezni, tehát ezek összege elkülönített, és ezeken változtatni nem tudunk – így kalkulálni sem kell velük, a teljes összegből levonhatók.

4.3.3. A feladat megfogalmazása

Feladatunk tehát az, hogy minden kirendeltség minden aktív eszközére megmondjuk – a célcsoport és a megtérülési ráta korlátozó feltételeinek figyelembe vételével – a résztvevők várható vagy elérendő számát oly módon, hogy a lehető legolcsóbb működtetést tudjuk biztosítani.

4.3.4. Formális felírás

A létszámkorlátokat, illetve összegeket a teljes projekt keretében vizsgáljuk, hiszen összességében ezek a paraméterek nyilván arányosítás után módosíthatók.

A változók

$$X_{i,j} \in \mathbb{Z}, \quad i = 1, \dots, 10, \quad j = 1, \dots, 4,$$

azaz a 10 kirendeltség (csak a fővárosiak) mindegyikén megadásra kerül a 4 aktív eszközre küldendő álláskeresők létszáma. A változók egész mivolta következik persze abból, hogy létszámadatokkal dolgozunk.

A korlátozó feltételek

$$X_{i,j} \geq 0, \quad i = 1, \dots, 10, \quad j = 1, \dots, 4.$$

A nemnegativitási feltétel fontos, hiszen létszámadatokkal dolgozunk.

$$\sum_{i=1}^{10} X_{i,1} \geq R_1.$$

E feltétel a projekt egyik monitoring adata: nevezetesen, hogy adott számú résztvevő átképzése kötelező. Jól látható tehát, hogy a 4 aktív eszköz közül csak az egyikre, a képzésre vonatkozik monitoring adat – a többi eszközben való részesedés a képzés után fennmaradó keretből szabadon csoportosítható. (R_1 e projektben 1560.)

$$\sum_{i=1}^{10} \sum_{j=1}^4 X_{i,j} \leq K,$$

azaz K számú résztvevőnél több összességében nem segíthető e projekt keretein belül. (K e projektben 2600.)

$$\sum_{j=1}^4 X_{i,j} \geq 0,9P_i, \quad i = 1, \dots, 10$$
$$\sum_{j=1}^4 X_{i,j} \leq 1,1P_i, \quad i = 1, \dots, 10$$

ahol P_i az adott kirendeltséghez tartozó, előre megadott célcsoportokba sorolható összes álláskereső becsült száma. A teljes létszámon belül a projekt nem rendelkezik arról, hogy mely célcsoportból mekkora bevonás az elvárt, így nem érdemes a célcsoportokat külön-külön kezelni a modellben.

$$\sum_{i=1}^{10} d_{i,1} X_{i,1} \geq \left\lceil \frac{5}{6} R_1 \right\rceil,$$

a képzések során elvárjuk, hogy a képzésen résztvevők $\frac{5}{6}$ -a sikeresen teljesítse a képzést. A $d_{i,1}$ értékek az előző projektekben tapasztalt, kirendeltségenkénti sikeres képzési teljesítések aránya, továbbá a képzésre bevonandók számát fixnek tekintjük – így a sikeresen végzők száma is előre meghatározott.

Bár alapvetően 1560 fő képzésre bevonása van előírva, ha mégis többen kerülnének be, úgy a sikerességi ráta arányosítva van – tehát nem abszolút, bizonyos projekteknelvánt létszám van megfogalmazva arra vonatkozóan, hogy hány sikeres átképzés esetén sikeres a projekt. (Ez 1560-es képzési bevonás esetén 1300 fős sikeres befejezést jelent.)

$$\sum_{i=1}^{10} \sum_{j=1}^4 a_{i,j} X_{i,j} \geq 0,25K,$$

ahol $a_{i,j}$ az adott kirendeltségen adott eszközzel ellátott álláskereső esetén a 180. napon állásban lévők arányát jelzi. Azaz: a projektbe bevontak 25%-a esetén előírás, hogy ilyen arányban az ellátás befejezése után 180 nappal alkalmazásban legyen a projektben résztvevő. (Ez a $K = 2600$ -as létszám esetén 650 főt jelent.)

$$\sum_{i=1}^{10} \sum_{j=1}^4 b_{i,j} X_{i,j} \geq 0,85K,$$

ahol $b_{i,j}$ az adott kirendeltségen az adott eszközzel ellátott álláskereső esetén a sikeres eszközhasználat becsült arányát jelenti. (Sikertelen eszközhasználat esetében arra lehet gondolni, hogy például abbahagyja a képzést idő előtt, tönkremegy a vállalkozása, stb.) Az elvárt sikeresség a teljes projekt résztvevőinek 85%-a. (Ez a projekt 2600-es bevonása alapján 2210 főt jelent.)

$$\sum_{i=1}^{10} \sum_{j=1}^4 c_{i,j} X_{i,j} \leq M,$$

ahol $c_{i,j}$ a kirendeltségenként lebontott, eszközökre fordított összeget jelenti.

Ez jellemzően eszközönként egyenlő (azaz jellemzően nem fizetnek több vállalkozóvá válási támogatást bizonyos kirendeltségeken), azonban eltérések lehetnek, mert pl. könnyen megeshet, hogy bizonyos képzéseket pályakezdőknek olcsóbb megszervezni, mint valamely más veszélyeztetett csoportnak.

27. Megjegyzés. Fontos lehet kiemelni a $b_{i,j}$ és $d_{i,j}$ értékek közötti eltérést. A projektben több eszköz is helyet kap, melyek mindegyikére előírhatnak sikerességi rátákat, mutatókat. A d együtthatók csak a képzésre, míg a b együtthatók minden projektelemlre vonatkoznak (specifikusan a képzéseket is magukba foglalva).

M a teljes projekt fix költségekkel csökkentett mértékét jelenti, azaz nyilván nem léphetjük túl a projekt költségvetését (ezt a korlátozó feltételt attól függetlenül el kell helyeznünk, hogy utána a célfüggvényünk is ezt a mennyiséget fogja minimalizálni).

28. Megjegyzés. *Az első feltétel esetén is felmerülhet akár az alábbi átfogalmazás: amennyiben arra koncentrálunk, hogy adott esetben az arányok kézben tartása, illetve arányok megtartása a cél, úgy:*

$$\sum_{i=1}^{10} \left(d_{i,1} - \frac{5}{6} \right) X_{i,1} \geq 0,$$

felírás is megfelelő. Azonban esetünkben előre kötött a képzésbe bevonandó létszám, így az első felírás is megfelelő.

A célfüggvény

$$\min \left\{ \sum_{i=1}^{10} \sum_{j=1}^4 c_{i,j} X_{i,j} \right\},$$

azaz, bár felülről korlátozott, hogy mekkora összeggel tudunk dolgozni, szeretnénk mégis minél olcsóbb üzemeltetést elérni a feladatunkban. Fontos lehet megjegyezni, hogy előfordulhatnak olyan esetek, amikor alsó korlátokat fogalmaznak meg bizonyos pénzköltségekkel kapcsolatban. Ilyen lehet az, amikor egyes célcsoportok támogatására előírják, hogy adott összeget mindenképpen a támogatásra kell fordítani, stb. Ilyen esetben ez nem képezi az optimalizálás részét: ez az összeg – annak kötelezően elköltendő volta miatt – kikerül a rendszerből, hiszen az eredményességtől függetlenül elköltésre kerül.

Összefoglalva

Az előzőekben vázolt formális megfogalmazásokat egy modellben összeírva az alábbi alakot nyeri a megoldandó feladat:

A változók:

$$X_{i,j} \in \mathbb{Z}, \quad i = 1, \dots, 10, \quad j = 1, \dots, 4.$$

A korlátozó feltételek:

$$\begin{aligned} X_{i,j} &\geq 0, \\ \sum_{i=1}^{10} X_{i,1} &\geq R_1, \\ \sum_{i=1}^{10} \sum_{j=1}^4 X_{i,j} &\leq K, \\ \sum_{j=1}^4 X_{i,j} &\geq 0,9P_i, \quad i = 1, \dots, 10, \\ \sum_{j=1}^4 X_{i,j} &\leq 1,1P_i, \quad i = 1, \dots, 10, \\ \sum_{i=1}^{10} d_{i,1} X_{i,1} &\geq \left\lceil \frac{5}{6} R_1 \right\rceil, \\ \sum_{i=1}^{10} \sum_{j=1}^4 a_{i,j} X_{i,j} &\geq 0,25K, \\ \sum_{i=1}^{10} \sum_{j=1}^4 b_{i,j} X_{i,j} &\geq 0,85K, \\ \sum_{i=1}^{10} \sum_{j=1}^4 c_{i,j} X_{i,j} &\leq M. \end{aligned}$$

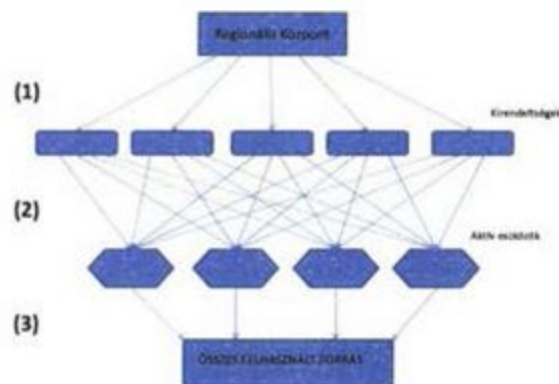
A célfüggvény:

$$\min \left\{ \sum_{i=1}^{10} \sum_{j=1}^4 c_{i,j} X_{i,j} \right\}.$$

29. Megjegyzés. Így egy lineáris programozási feladatot kaptunk, mely akár nagy méretben is könnyen megoldható, ráadásul így nem igényel külön programot – használható a felhasználó tulajdonában lévő EXCEL[©] megoldója.

4.3.5. Gazdasági interpretáció

Az előző pontban tárgyaltaknak sajátos vetületei vannak a gyakorlatban. Egy megfelelő gazdasági környezetben az alábbi módon világíthatjuk meg e speciális vetületet.



16. ábra. A vizsgált közigazgatási rendszer vázlatos szerkezete – POWERPOINT®

A megfelelni vágyás kényszerítő ereje

Képzeliük el, hogy egy szervezet (pl. az Európai Unió) biztosít egy minisztérium számára egy akkora összeget, ami messze nagyobb, mint az adott minisztérium eddigi, arra a tevékenységére fordított költségvetése. Viszont csak akkor rendelkezik ezzel az összeggel, ha egy előre meghatározott, racionális elvárást teljesíteni tud.

Ezek után a minisztérium azt mondja a regionális központnak, hogy adódott egy külső forrás, amiből az elvárt teljesítést végre kell hajtani. A regionális központ, hogy elnyerje a minisztérium megbecsülését, a kirendeltségeit már egy nagyobb tervről tájékoztatja mint minimálisan teljesítendő elvárásról. (Persze jól felfogott érdekében nem irracionálisan nagy emelést hajt végre.)

A megfelelni vágyás persze a kirendeltségekben is él a központ felé, így mozgósítja a vele kapcsolatban álló szolgáltatókat, hogy azok mindenképpen teljesítsék az elvárásaikat – szintén nagyobb mértékben, mint amit a központ elvár tőlük.

Ez az irány mutatja majd, hogy az alsó korlátok hogyan alakulhatnak ki, mi motiválhatja az alsó korlátok „egymásba ágyazódását”.

Felülről szerveződő biztosítás

A hazai viszonyokhoz kicsit közelebb álló, realiztikusabb módon illeszkedő modell: a minisztérium rendelkezik egy bizonyos összeggel, amit a biztonság kedvéért „eltitkol” a regionális központ elől. Azt állítja, hogy ennél csak kevesebbet költhet.

Ezt a központ szintén tovább csökkentve adja tovább a kirendeltségeknek, akik a velük kapcsolatban álló szolgáltatók felé szintén nem a teljes összeget jelentik – így minden szinten kialakul egy kisebb biztosíték az esetlegesen fellépő többletigények kielégítésére.

Világos, hogy ez az elkölthető pénz mértékére mutató felső korlátokat fogja befolyásolni a későbbiekben még megvilágításra kerülő módon. Esetünkben is általában ezzel a rendszerrel találkozunk – és az is érthető, hogy általában ennek inverze jelenhet meg, ha pl. elérendő létszám-feltételeket kívánunk szabni (többet mondunk, mint amennyit valóban teljesíteni kell a sikerhez).

E fenti tulajdonságokat jól leírható matematikai tulajdonságokkal is jellemezhetjük az alábbi módon.

4.4. Strukturális tulajdonságok

Amennyiben egy ilyen pénzosztási rendszert folyamfeladatként fogunk fel, ahol egy központi helyen átadják a pénzt, mely nyelők sokaságában (vagy egy központiként felfogott nyelőben) a felhasználókhoz kerül, úgy sajátságos struktúrát vehetünk észre.

Ahhoz, hogy a fenti feladatban vázolt probléma viselkedését fel tudjuk tárni, néhány definícióra, megállapodásra lesz szükségünk.

Legyen adott egy $D = (V, A)$ irányított gráf, ahol V a csúcsok halmaza, $|V| = n$, míg A az irányított élek halmaza $|A| = m$. A gráf nem tartalmaz párhuzamos éleket és hurokéleket. Tekintsük e véges, összefüggő gráfon értelmezett azon folyamfeladatot, ahol csak 1 forrás (S) létezik, és az f folyamérték adott. Vizsgáljuk a megengedett folyamok poliéderét.

Kiemeljük, hogy azon tulajdonságok, amiket általában egy folyamfeladattól elvárhatunk, teljesülnek: a forrás és a nyelő kivételével minden közbülső állapotra feltételezhető, hogy a befolyó pénzmennyiség távozik is onnan (azaz pl. a működési költségeket nem ezen projekt keretéből, illetve nem ezen problémakör keretein belül tárgyaljuk).

27. Definíció. *Extremális folyamnak* nevezek mostantól egy folyamot, ha az nem áll elő másik két folyam konvex kombinációjaként.

28. Definíció. *Legyen adott minden x_i , $i = 1, \dots, m$ élre egy l_i alsó korlát, azaz $0 \leq l_i \leq x_i$. Ekkor definiáljuk az alábbi, csúcsokon értelmezett értékeket:*

$$\underline{\beta}(y_j) := \sum_{x_i: x_i = \overrightarrow{y_j}} l_i, \quad j, z = 1, \dots, n,$$

$$\underline{\kappa}(y_j) := \sum_{x_i: x_i = \overleftarrow{y_j}} l_i, \quad j, z = 1, \dots, n,$$

azaz az alsó kapacitások összege a be- és kilépő éleken.

29. Definíció. *Analóg módon legyen adott minden x_i , $i = 1, \dots, m$ élre egy u_i felső korlát, azaz $0 \leq x_i \leq u_i$.*

Ekkor definiáljuk az alábbi, csúcsokon értelmezett értékeket:

$$\begin{aligned}\bar{\beta}(y_j) &:= \sum_{x_i: x_i = \overline{y_i y_j}} u_i, \quad j, z = 1, \dots, n, \\ \bar{\kappa}(y_j) &:= \sum_{x_i: x_i = \overline{y_j y_i}} u_i, \quad j, z = 1, \dots, n,\end{aligned}$$

azaz a felső kapacitások összege a be- és kilépő éleken.

Ekkor az alábbi állítás fogalmazható meg.

14. Állítás. (Alsó és felső korlátok esete): *Tekintsük azt a véges gráfon értelmezett folyamfeladatot, amelyre teljesül, hogy*

- (i): *1 darab S forrás pontja van,*
- (ii): *minden élén értelmezett egy nemnegatív u_i felső kapacitás,*
- (iii): *minden élén értelmezett egy nemnegatív l_i alsó kapacitás,*
- (iv): *adott az f folyam nagysága,*
- (v): *S kivételével minden csúcsra*

$$\begin{aligned}\underline{\beta}(y_j) &= \underline{\kappa}(y_j), \\ \bar{\beta}(y_j) &= \bar{\kappa}(y_j),\end{aligned}$$

továbbá

$$\begin{aligned}\underline{\beta}(y_j) &< \bar{\beta}(y_j), \\ \underline{\kappa}(y_j) &< \bar{\kappa}(y_j).\end{aligned}$$

Ekkor bármely megengedett extrémális folyam esetén az S -en kívüli minden pontba legfeljebb 1 élén fut be és lép ki az alsó, illetve a felső kapacitástól különböző nagyságú folyam.

Bizonyítás: Az állítás triviális, hiszen ha két élén is futna be, illetve lépne ki bármely pontba a folyamról érték, úgy (akár a forrásig visszavezetve) az alsó és felső korlátok kapacitásainak egyenlősége miatt egy kör alakítható ki (minden, a körben szereplő pontban lenne olyan él, mely nem alsó vagy felső kapacitáson üzemel), mely kör mentén a kapacitás-eltérések minimumával felfelé és lefelé elmozdulva két olyan folyamot hozhatunk létre, melyek lineáris kombinációja az éppen megtalált folyamunk – az tehát nem lehet extrémális. ■

A fenti formálisan adott feladat megfelel annak a struktúrának, amit a matematikai modellezésnél használtunk, így annak sajátosságai is teljesülni fognak az optimális megoldásunkban.

Ennek azonban vannak sajátságos, lényegében „politikai” interpretációi. Ugyanis az alsó vagy felső korláton telített élek azt eredményezik, hogy lesznek kirendeltségek, amelyek bizonyos eszközökön maximális terheléssel futnak, míg más kirendeltségek minimális terhelésen lesznek tartva. Így az adott kirendeltség területén élő álláskeresők determinálva lesznek, hogy az optimális gazdálkodás érdekében mely eszközök elérését fogják elsősorban preferálni/szorgalmazni a projekt megtérülési rátájáért felelős ügyintézők.

Ez diszkriminatív bánásmódot eredményezhet, mely ugyan gazdálkodási szempontból könnyen védhető (hiszen így költünk a legkevesebbet, ráadásul a hatékonyságot is szem előtt tarthatjuk, hiszen nem darabolódik szét többfajta ügyintézésre a szolgáltatás), azonban társadalmi szempontból mindenképpen megkérdőjelezhető ennek jogossága.

30. Megjegyzés. *Az alsó létszámkorlátokat a rendszerből kiiktathatjuk oly módon, hogy a kötelezően előírt alsó létszámokat és annak költségét ignoráljuk a rendszerből – így a változóinkra továbbra is a nemnegativitás lesz az alsó korlát, míg a felső korlátokat az előírt legkisebb létszámokkal csökkenthetjük.*

4.4.1. Robusztus optimalizálás

Világos, hogy a fenti modelltől a felhasználók azt várják el, hogy az előre nem látható igények esetén is stabilan működtethető legyen. Az általunk vizsgált rendszerben ugyanis fontos szempont a rendszer érzékenységének – ha úgy tetszik, más megközelítésben: flexibilitásának vizsgálata. Azonban a rendszerben lévő, előre nem látható vagy kalkulálható elemek megjelenése miatt nagyfokú bizonytalanság van. Ennek okán a fenti feladat érzékenység-vizsgálatát sztochasztikus eszközökkel kívántuk megvalósítani.

Ehhez szimulációkat szerettünk volna bevetni, de a szimulációk szolgáltatva eredményeket valamely megoldáshoz kellett mérnünk – azok jóságát valamely fix, determinisztikus állapothoz kellett viszonyítanunk.

Amennyiben figyelembe vesszük Ben-Tal, El Ghaoui és Nemirovski [3], illetve Prékopa [35] megjegyzéseit, úgy a következőket állapíthatjuk meg.

Olyan megoldást kell keresnünk, mely a rendszerben lévő véletlen mennyiségek tetszőleges realizációja mentén szintén megengedett megoldásként jelentkezik.

Ez a megkövetelés számunkra két okból is túlságosan erős: az egyik ok, hogy a felmerülő véletlenek nem feltétlenül korlátosak. Ha például normális eloszlású véletlen változókkal szeretnénk dolgozni, vagy legalábbis olyan folytonos változókkal, melyekre nem feltétlenül szeretnénk korlátos tartókat meghatározni.

Könnyen előfordulhat, hogy a robusztus megoldás bizonyos paramétereit nem teljesíti élesen. Érdemes észrevenni, hogy elképzelhetőek olyan esetek, amikor egy 25%-os megtérülési rátát a robusztus megoldás 40-50%-on teljesít, miközben az összes pénzt elkölti.

Szintén érdekes vizsgálni, hogy még ha a bizonytalanság kezelésére korlátos tartójú eloszlást választunk is (pl. egyenletes eloszlásokat), akkor is kontinuum-sok realizáció van, mely kontinuum-sok poliédert eredményez.

Ennek következtében az alábbi megfontolásokat tesszük – melyek a gyakorlati probléma szempontjából adekvátak.

4.4.2. A megoldandó feladat vázlatos szerkezete

A rendszer legegyszerűbb módon az alábbi formában adható meg:

$$\begin{aligned} Wx &\leq h, \\ 0 &\leq x, \\ \min c^T x. \end{aligned}$$

Konkrét feladatunk esetében az alábbi megszorítások is érvényben vannak a fenti formális feladat megoldásakor: az l értékek módosíthatók 0-ra, ekkor az u felső korlátok $u-l$ -re módosulnak. A létszámkorlátok a megfelelő módon csökkenthetők, illetve a felhasználandó pénzmennyiséget is ezen elmozdításoknak megfelelően csökkenthetjük (fix költségként elkönyvelve ezt a kiadást), amennyiben minimális elvárásokat fogalmaznak meg a felhasználók.

Ezen túlmenően esetünkben az alábbi megállapítások is megtehetők:

- $h \geq 0$ feltehető (hiszen itt általában létszámokra vonatkoznak a feltételek);
- $W \geq 0$ feltehető, hiszen itt általában vagy a megtérülési ráták szerepelnek, vagy létszámok, illetve azok összegzései;
- W elemei közül egyesek konstansok (pl. 0-k, ha az adott kirendeltség esetén egy-egy eszköz nem érhető el egyáltalán), míg más elemeket olyan valószínűségi változóknak gondolunk, melyek eloszlása – a fenti példánál maradva – véges tartójú;
- c célfüggvény együtthatóiról feltehető, hogy fixek, hiszen törvényileg nem tilos, hogy a támogatásokat nem arányosítva, hanem fix támogatásként kezelve adjuk ki (pl. minden képzést 1.000.000 Ft-tal támogatunk).

Ezen feltételek mellett az alábbi állítás fogalmazható meg:

15. Állítás. Legyen $W = (W_{i,j})$ mátrix olyan, hogy $\exists i$ és j , melyre $W(i,j) \sim U(l(i,j), u(i,j))$, azaz legalább egy koordinátájában véletlen valószínűségi változó, mely véletlen valószínűségi változó az $[l(i,j), u(i,j)]$ intervallumon egyenletes eloszlású.

Legyen most P mátrix az alábbi módon definiálva:

$$P(i,j) = \max(W(i,j)),$$

azaz minden olyan esetben, ahol W valamely koordinátája véletlen, ott $P(i,j) = u(i,j)$ egyenlőség álljon fenn, ahol pedig W nem véletlen valószínűségi változó, ott $P(i,j) = W(i,j)$ egyenlőség teljesüljön.

Ekkor, ha valamely x vektorra teljesül, hogy

$$\begin{aligned} Px &\leq h, \\ 0 &\leq x, \end{aligned}$$

úgy az is teljesül, hogy

$$\begin{aligned} Wx &\leq h, \\ 0 &\leq x, \end{aligned}$$

bármely $W(i, j)$ véletlen realizáció mentén.

Bizonyítás: A nemnegativitási feltételek okán az állítás triviális. ■

Ezen állítás értelmében robusztus megoldását nyerjük a feladatnak abban az esetben, ha az egész folyamat oly módon oldjuk meg, hogy a lehetséges realizációkat maximálisnak vesszük, a célfüggvényben nincsen véletlen, és a realizációk tartója nemnegatív és korlátos.

Esetünkben a rendszer olyan, hogy vannak alulról korlátozó feltételek is – melyek esetében a fenti állítás egy kisebb módosítással átvezethető, az alábbi formában.

16. Következmény. *Tegyük fel, hogy a fenti feladatban megfogalmazott mátrixok és vektorok nemnegativitási feltételei továbbra is fennállnak.*

Legyen W_1 olyan, hogy $\exists i$ és j , melyre $W_1(i, j) \sim U(l_1(i, j), u_1(i, j))$, azaz legalább egy koordinátájában véletlen valószínűségi változó, mely véletlen valószínűségi változó az $[l_1(i, j), u_1(i, j)]$ intervallumon egyenletes eloszlású.

Legyen most P_1 mátrix az alábbi módon definiálva:

$$P_1(i, j) = \max(W_1(i, j)),$$

azaz minden olyan esetben, ahol W_1 valamely koordinátája véletlen, ott $P_1(i, j) = u_1(i, j)$ egyenlőség álljon fenn, ahol pedig W nem véletlen valószínűségi változó, ott $P_1(i, j) = W_1(i, j)$ egyenlőség teljesüljön.

Legyen W_2 olyan, hogy $\exists i$ és j , melyre $W_2(i, j) \sim U(l_2(i, j), u_2(i, j))$, azaz legalább egy koordinátájában véletlen valószínűségi változó, mely véletlen valószínűségi változó az $[l_2(i, j), u_2(i, j)]$ intervallumon egyenletes eloszlású.

Legyen most P_2 mátrix az alábbi módon definiálva:

$$P_2(i, j) = \min(W_2(i, j)),$$

azaz minden olyan esetben, ahol W_2 valamely koordinátája véletlen, ott $P_2(i, j) = l_2(i, j)$ egyenlőség álljon fenn, ahol pedig W nem véletlen valószínűségi változó, ott $P_2(i, j) = W_2(i, j)$ egyenlőség teljesüljön.

Ekkor, ha valamely x vektorra teljesül, hogy

$$\begin{aligned}P_1 x &\leq h_1, \\P_2 x &\geq h_2, \\0 &\leq x,\end{aligned}$$

úgy az is teljesül, hogy

$$\begin{aligned}W_1 x &\leq h_1, \\W_2 x &\geq h_2, \\0 &\leq x,\end{aligned}$$

bármely véletlen realizáció esetén.

Bizonyítás: A nemnegativitási feltételek miatt továbbra is triviális az állítás. ■

4.4.3. A véletlenítés gazdasági interpretációja

Az elsődleges kérdés ezek után nem is feltétlenül az volt, hogy optimálisan, a lehető legolcsóbban hogyan tudjuk működtetni a rendszert (a politikai hozadékok miatt sem feltétlenül ez a legfontosabb elvárásunk).

Ennél sokkal érdekesebb, alkalmazási szempontból azonban adott esetben fogósabb kérdés volt az, hogy a megjelenő, esetünkben véletlennek tekintett jelentkezők száma milyen torzulást, milyen variabilitást eredményezett a rendszerben.

Amennyiben fixnek tekintjük a teljesítendő korlátokat, érdekes lehet vizsgálni az alábbiakat:

1. A véletlen hatására más és más kirendeltségenkénti teljesülési rátákat tételezünk fel – ezekkel külön-külön, újra és újra megoldjuk a feladatot, figyelve, hogy a célfüggvény, illetve a kirendeltségi szerepek (alsó vagy felső korláton teljesítés) mennyire változnak.
2. Szintén érdekes hatás lehet, hogy a célcsoportok nagyságát, illetve az ott megfogalmazott korlátokat némiképpen módosítjuk, tehát akár a $\pm 10\%$ -ot változtatjuk (akár ferdén is, tehát felfelé mondjuk jobban elengedve a sávot, míg lefelé szigorúbbnak tekintve az elmozdulásokat).
3. Lehetséges irány az is, hogy azon eszközök esetén, ahol mérlegelési lehetőség van arra, hogy mekkora összeggel támogatunk egy álláskeresőt (vállalkozóvá válási támogatás), a különböző mértékekkel való véletlenítés hatása sem elhanyagolható, hiszen esetleges megtakarításokkal vagy nagyobb összeggel való motiválásokkal is kiváltható lehet jobb sikerességi ráta az adott eszközön.

31. Megjegyzés. *A fenti strukturális megkötésekre vonatkozó tétel feltételeit az alsó és felső korlátok kimenő, illetve bemenő összesítéseiről – a gazdasági interpretációs elvárások okán is – teljesíteni tudjuk, tehát a fenti tétel érvényben marad az alkalmazásunkban használt feladatra.*

Fontos továbbá kiemelni, hogy a támogatásokat menet közben nem szokás változtatni: azaz ha egy projekt elején egy adott összegű támogatási mérték mellett döntöttek, akkor minden kirendeltségen, minden időpontban azt az összeget fogják megítélni.

A korábban már bemutatott ábrán 3 szintet különböztettünk meg, az ezeken való véletlenítések az alábbi módokon értelmezhetőek:

1. Az első szinten a központ és a kirendeltségek közötti véletlenítés történhet – ez azt jelenti, hogy a központ nem feltétlenül kalkulál minden egyedi esettel, ami a kirendeltségekről érkezik, hanem csak az összes pénzmozgást igyekszik figyelni.

Ez az önálló gazdálkodás esete: a központ megbízza a kirendeltségek pénzosztási technikájában, így csak a teljes igényt figyeli. Ezen a ponton való véletlenítés során feltételezhetjük, hogy a gyakorlatban csak az számít, hogy a kirendeltségek teljesítik-e a vállalt rátájukat, és hogy azon belül mit milyen módon oldanak meg, szuverén felelősségükre van bízva.

A rendszer szabadságfoka nyilván kevesebb, mint a kirendeltségek teljes száma, hiszen a teljes költségvetés fix – így nem lehet minden paraméter szabad.

2. A kirendeltségek és az aktív eszközök közötti kapcsolat: itt jelennek meg a közvetlen igények. Ezen a ponton lehet leginkább fontos a véletlenítés folyamata, ráadásul itt a szabadsági fok megegyezik az élek számával – legfeljebb lesznek olyan igények, melyeket nem tudunk kielégíteni, elfogy a pénzünk. (Vagy éppen a rátákat nem tudjuk teljesíteni.)

Ezen a ponton nyer jelentőséget az egyes élek maximalizálása vagy minimalizálása, hiszen itt látható, hogy mely kirendeltségeken mely eszközökre érdemes jelentkezni.

3. A harmadik szint azt mutatja, hogy mely eszközre összességében milyen jelentkezés történt: világos, hogy a kimeneti ráták, illetve közvetlenül (az egységes elbírálás miatt) a célfüggvény értéke is ezektől a véletlenítésektől függ.

Természetesen a célfüggvény a második szint értékein ugyanúgy értelmezhető, tehát nincsen szükség külön, még a harmadik szinten is elvégzett összegzésre.

4.4.4. A robusztus megoldás gazdasági interpretációja

A robusztus megoldásunk tehát olyan, hogy bármely esetben megengedett megoldásként kell hogy szolgáljon. Azonban ez túlságosan erős megkötéseket, illetve alkalmazási oldalról túlonúl kötött, túlonúl merev helyzeteket eredményez, mely ráadásul felhasználói oldalról „politikailag sem egészen korrekt” magatartáshoz vezet – bár tény, hogy egy minden helyzetben optimális, olcsó megoldást kínál.

A szimulációnknak és véletlenítésnek jelenleg tehát lényegében hármas célja lesz:

1. A realisabb kép érdekében megvizsgáljuk, hogy számos kísérlet és realizáció várhatóan milyen megoldásokat szolgáltat – a létszámok milyen keretek között mozoghatnak, átlagosan, egyes kirendeltségekre és egyes eszközökre hány fő elhelyezésével kell számolnunk.
2. A robusztus megoldás kapcsán nyert optimális megoldáshoz képest milyen mértékben fogunk elmozdulni a pénzmennyiséget illetően, azaz átlagosan – e megoldáshoz képest – mennyivel vagyunk veszteségesebbek.
3. A nagyszámú szimuláció okán egyfajta könnyed, empirikus ellenőrzését adhatjuk azon strukturális tételnek, hogy valóban csak egyetlen olyan kirendeltség létezhet majd az optimális megoldásokban, mely kirendeltségen mozgástér alakul ki – minden más kirendeltség vagy alsó, vagy felső korlátot üzemel majd abban az értelemben, hogy a vállalt létszámukhoz képest vagy 10% többletben, vagy 10% hiányban kell dolgozniuk.

Mindezen matematikai követelményeken túl a felhasználók felé is voltak kötelezettségek, melyeket a projekt kapcsán szerettünk volna kielégíteni:

1. Az általunk alkalmazott eljárás a felhasználónál használt programcsomagba leprogramozható legyen. Ezzel kiküszöböljük esetleges más program megvételét. Az alkalmazások során fontos annak figyelembe vétele, hogy a felhasználónak mire van anyagi forrása, és ezen az anyagi forráson belül tudjuk a lehető legjobban működő, leginkább elfogadható és számára is kielégítő megoldást szolgáltatni.
2. A felhasználó által állított szempontok minden esetben prioritást élveznek egy alkalmazásban – amennyiben azok nem készítetnek minket hibázásra vagy megalapozatlan állításokra.

Ennek okán olyan megoldást kellett választanunk, mely az EXCEL[®] programcsomagban is működőképes. Ezen a ponton nyert jelentőséget az általunk is tesztelt véletlengenerátor. Ennek használata során ugyanis egy munkatábla volt létrehozható a programban, ahová előre generált módon (hagyományos értelemben véletlenszám-táblázatként) elhelyeztünk 60.000 véletlen vektort, melyeket a programunk a szimuláció során használt.

Ennek oka az volt, hogy így arra is felkészítettük a programot, ha esetleg olyan véletlen vektorrendszerre lenne szükségünk, melyben különböző, előre meghatározott esetleges összefüggéseket szeretnénk kódolni, akkor csak ezt a táblát kell lecserélni a megfelelő véletlen vektorok táblázatára.

32. Megjegyzés. *Erre lehet példa az, hogy ha egy időszakban valamely eszköz preferáltabb, akkor ott jellemzően növekednek a létszámok, míg más eszközökön várhatóan csökkenéseket fogunk tapasztalni.*

Ráadásul a táblázat létrehozásával csak egyetlen véletlen számot kell definiálni – hogy mely sortól kezdődően indítsa el az algoritmust, utána a szimuláció már „determinisztikus”, ezzel pedig reményeink szerint stabilabb, gyorsabb alkalmazást tudunk a felhasználóknak biztosítani annál, mintha minden lépésben újabb véletlen számokat, vektorokat generáltunk volna az adott pozíciókra.

Két feladatot szeretnénk tehát szimuláció segítségével vizsgálni:

1. Miként alakul a célfüggvény értéke annak függvényében, hogy l és u közötti véletlen számokat választunk (vagy akár csak egyes koordináták mentén választunk véletlen számokat, egyeseket pedig fixnek tekintünk). Mindezen hatást a robusztus megoldással fogjuk elsősorban összevetni.
2. A következő, számunkra talán legfontosabb vizsgálatot szeretnénk elvégezni: tartsuk nyilván (N darab szimulációt készítve) az alábbi arányokat:

$$DBK_{i,+} = \frac{\#\{x_i = u_i\}}{N},$$

$$DBK_{i,0} = \frac{\#\{l_i < x_i < u_i\}}{N},$$

$$DBK_{i,-} = \frac{\#\{x_i = l_i\}}{N}.$$

Azaz megnézzük, hogy a véletlenített koordinátákon hány szerepel arányaiban az előzetesen kalkulált alsó ($DBK_{i,-}$) vagy felső korlátot ($DBK_{i,+}$), vagy netán mennyi helyen van mozgásterünk ($DBK_{i,0}$).

E második vizsgálatnak a következő a jelentősége: a közigazgatás sajátossága, hogy felülről irányított rendszer, amiben a módosításokat ki lehet adni utasításba – azonban nehezen végezzük el azokat, mert a rendszer nehezen mobilizálható. Ennek következtében, ha ráállnak egyfajta munkafolyamatra a dolgozók, nehézkes menet közben irányt változtatni.

Ha egy kirendeltségnek utasításba adjuk, hogy mostantól egy adott eszközön minimális eladás lehet – vagy épp fordítva, toborozni kell rá az embereket, majd negyed év múlva az ellenkezőjére változtatom a stratégiát, mert más kirendeltségen történt valami, akkor a rendszer nagysága miatt a változtatások – bár az utasítás megszületik – nagyon nehezen indulnak be.

Továbbá egy sajátos helyzet alakul ki felhasználói oldalról: egyik nap még mindenkinek ajánlunk mindent, sőt, a toborzás sajátossága miatt még kérjük is a már felvetteket, hogy ha tudnak valakiről, akit érdekelne az ajánlat, hívják oda. Majd következő nap az ellenkezőjét kell mondanunk az érdeklődőknek: minden elfogyott, elkésett. Ez pedig nagyobb veszteségekkel járhat hosszú távon (bár mérni nem tudjuk), mintha még hagytuk volna a rendszert működni egy kicsit a régi metódus szerint.

Ez azt jelenti, hogy az előző fejezetben ismertetett feladat esetében több olyan pont is van, ahol véletlenítést alkalmazhatunk – ezek mindegyikét szeretnénk vizsgálni, sőt: ezek különböző kombinációival is kísérleteket szeretnénk végrehajtani.

4.5. A lehetséges szimulációs pontok

Felsorolás-szinten vizsgáljuk meg, hogy – az előző fejezet jelöléseit alkalmazva – a bizonytalanság matematikai modellezésére mely pontokon hajthatunk végre véletlenítést, illetve mely pontok azok, melyek megkívánják, hogy véletlen változóként gondoljunk rájuk.

1. P_i , azaz az adott kirendeltségen fellelhető összes, adott projektbe bevonható álláskereső teljes létszáma. Világos, hogy ezen a ponton alkalmazható lenne véletlenítés, hiszen semmifajta garancia nincsen arra, hogy egy nagyobb elbocsátási hullám, vagy egy adott területen történő nagyobb beruházás ne tudná alapjaiban megváltoztatni a rendszer lokális viselkedését – azonban az feltételezhető, hogy az előző évek tendenciái, tapasztalatai 1-2 éves távlatokban aránylag stabilnak tekinthetők (nem számolunk, mert nem is tudunk jelenleg a különböző kormányzati intézkedések rövid vagy méginkább ultrarövid távú hatásaival). Mindezek alapján elfogadhatónak fest, hogy adott keretek között akár egyenletes eloszlás szerinti, vagy adott középérték melletti alacsonyabb szórású véletlen változókkal (pl. normális eloszlásúval) szimuláljuk ezen paraméter ingadozását.

Fontos kiemelni, hogy ezen a ponton nem történt vizsgálatunkban véletlenítés, hanem alsó és felső korlátként ennek teljesítését vagy teljesíthetőségét kirendeltségi hatáskörnek tekintettük (azaz, ha kell, akkor tudnak toborozni, vagy adott létszámot nem túllépve működni).

2. $d_{i,1}$ értékek, azaz a képzések sikeres teljesítésének kirendeltségi arányai. Világos, hogy bizonyos becslésünk, előzetes elvárásunk a korábbi projektek tapasztalataiból feltételezhető.

A létszámokat az előző pontban említett hatások befolyásolhatják, azonban azt feltételezhetjük, hogy a képzések sikerességét általában az álláskeresők végzettségi összetétele, illetve a képzőhelyek minősége befolyásolja leginkább, e két tényező aránylag stabil – így alacsony bizonytalansággal, de adott, előző évek tapasztalatán alapuló középértékkel dolgozhatunk itt is. Tehát egy adott középértékű és adott terjedelmű, pozitív tartójú egyenletes eloszlás elfogadható.

3. $a_{i,j}$, azaz a projektben való részvétel utáni 180. napon való állásban, alkalmazásban tartózkodás aránya, mely szintén rendelkezésre áll kirendeltségenként, eszközönként az előző évek hasonló projektjei után. Itt is feltételezhető, hogy a munkaerő-piaci általános mozgások (felvételek és elbocsátások) nem váltakoznak és változnak olyan drasztikus mértékben egyes kirendeltségekre koncentrálva, hogy indokolatlanná tegyék egy adott középérték melletti kis szórású, vagy adott, akár szűk határok közötti egyenletes eloszlású változó alkalmazását.

4. $b_{i,j}$, azaz a sikeres befejezés (a projekt bármely eszközén) aránya. Világos, hogy az előzőekben tárgyaltak közül ez a paraméter hatásában leginkább a $d_{i,1}$ paraméterhez van legközelebb, azonban itt látni kell, hogy az összes eszközre is vonatkozik egy teljesítési kvóta – de strukturális szinten, szigorúan a modellt vizsgálva ennek véletlenre vonatkozó megállapításai a $d_{i,1}$ paraméter megállapításaival analóg módon kezelhetők.

5. $c_{i,j}$, M , azaz a kirendeltségi eszközönkénti ár, illetve a teljes előírt költség. Ezeknek inkább politikai indíttatása van: míg a $c_{i,j}$ értékek világos, hogy a piacon elérhető képzési árakat, vagy a minimálbér mértékét jelenthetik (bizonyos támogatások a minimálbérhez kötöttek), melyek külső szemlélő mivoltunk okán „véletlenszerűnek” látszanak a tervezés pillanatában. Az M érték véletlenszerűsége már kérdésesebb, de itt arra kell elsősorban gondolnunk, hogy ha az igényelt fix összegnek egy adott hányadát nem tudjuk elkölteni, befektetni, akkor a projektnek a jövőben mutatott kockázata megnő: nem fogják a következő projekt meghirdetésekor akkora összeggel támogatni az intézményt, amekkorát igényként benyújt, mert az előzőekben vett tapasztalatok alapján úgysem használ el akkora tőkét – ebből következően a saját jövőbeni mozgásterünket szűkítjük. Illetve a fix költségek olyan mértékben lehetnek egy hosszabb távú projekt esetén véletlenszerűek, hogy pl. a bérköltségek nem fixek (valaki kilép pl. a munkából, és nem sikerül azonnal pótolni, így az a bérköltség felszabadul).

Jól látható tehát, hogy összességében számos olyan pont van, mely befolyásolja a központ által irányelveként megfogalmazott, kirendeltségnek javasolt létszámokat – adott eszközönkénti bevonás mellett.

Azt a megoldást választottuk, hogy ezek mindegyikét bevonjuk a véletlenítés folyamatába, hiszen még talán az egyes eszközök ára lehetne fixként kezelhető (bizonyosan azok is, mert pl. van olyan, mely a minimálbérhez kötött), de világos, hogy pl. képzések esetén nem mondható meg előre, hogy mennyibe fog kerülni (hiszen a bevont alany abba is hagyhatja), illetve más és más képző intézményekkel kötnek szerződést, ami nyilván befolyásolja az árakat, ahogy a képzés célja is differenciálja azokat.

33. Megjegyzés. *Fontosnak érezzük külön kiemelni az alábbiakat: a fenti helyzetekre és változókra nem rendelkezünk érdemi előzetes mérésekkel, információkkal. A fenti változók eloszlását nem ismerjük, sőt: azzal a problémával is számolnunk kell, hogy akár menet közben is változhat a fenti változók eloszlása, melyről mi csak utólagosan értesülünk (hiszen már megtörtént a realizáció, amiről az adatokat egy későbbi munkafolyamat során rögzítik csak), így előzetesen esélytelenek vagyunk kalkulálni.*

Az érzékenység-vizsgálatra mindenképpen szükségünk van egy ekkora rendszer esetén, így a legkényelmesebbnek az látszik, hogy egy elfogadható határon belül tartva a változókat, azok mindegyikét számtalan esetre szimulációkkal feltérképezzük (azt az esetet, hogy esetlegesen minden létező scenáriót feltárjunk, szintén elvethetjük, hiszen gigászi számú esetet kéne tárgyalnunk pusztán amatt, hogy a változók egy jelentékeny hányada – ahol pénzüsszegek szerepelnek – folytonosnak tekinthető).

4.6. A különböző megoldások összehasonlítása

Az eddig elmondottak alapján először megoldjuk a robusztus feladatot, majd ennek eredményéhez hasonlítjuk a szimuláció segítségével nyert megoldások átlagát.

4.6.1. A robusztus megoldás

Mint már említettük, a robusztus megoldással szeretnénk volna összehasonlítani a szimulációnk eredményeit. Ennek következtében tehát a realizációkat generáló véletlen mennyiségeket azok felső korlátján rögzítettük, definiálva így a realizációk metszetében található poliéderünket, a célfüggvény paramétereit pedig fixnek tekintettük (ahogy majd a szimulációban is tesszük).

Az alábbi eszközöket vizsgáljuk tehát:

- Képzés (K)
- Vállalkozóvá válási támogatás (VVT)
- Bértámogatás (BT)
- Bérköltség (BK)

Aktív eszköz	(K)	(VVT)	(BT)	(BK)
Ár	1.000.000	468.000	500.000	1.000.000
Sugár	0	0	0	0

14. táblázat. Az aktív eszközök ára, sugara a robusztus feladatban

Az aktív eszköz árán az előző évek tapasztalatai alapján megállapított átlagos árat értjük, míg a sugarán minden véletlen szám esetén az alkalmazott valószínűségi változó terjedelmének (a maximum és a minimum közötti különbségnek) a felét fogjuk érteni.

Politikai korlátokat nem állítottunk fel, illetve minden kirendeltségen betartottuk, hogy az előzetesen vállalt létszámhoz képest $\pm 10\%$ -ot eltérhetünk (ezek megjelennek mint korlátozó feltételek).

Kirendeltség kódja	Vállalt létszám
102	300
106	500
109	150
110	300
112	200
113	250
114	200
115	380
116	250
118	150

15. táblázat. Vállalt létszámok kirendeltségenként

Kirendeltség	K	VVT	BT	BK	SZEREP
102	40	216	0	14	ALSÓ
106	537	0	0	13	FELSŐ
109	71	80	0	14	FELSŐ
110	257	0	0	13	ALSÓ
112	37	128	0	14	ALSÓ
113	243	19	0	12	FELSŐ
114	166	0	0	14	ALSÓ
115	328	0	0	14	ALSÓ
116	32	187	0	14	MOZGÁSTÉR
118	121	0	0	14	ALSÓ

16. táblázat. Irányítandó létszámok és keretek a robusztus feladat megoldásában

A célfüggvény értéke 2250 millió Ft volt, tehát a teljes projekt-költségvetés 90%-át felhasználtuk.

Jól megfigyelhető, hogy majd minden kirendeltségen alsó vagy felső korláton vannak a létszámok. Vannak kirendeltségek, ahol a megadott paraméterek egyenlősége miatt (megtérülési ráták) több eszközzel is élhetnek, de a létszámok akkor is telítettek vagy le vannak szorítva. Bértámogatásra és bérköltség-támogatásra igen csekély létszámot oszt a robusztus megoldás – holott ezek igen népszerű eszközök.

4.6.2. Kevert robusztus megoldás

A szimulációkkal tehát a szerepcserékre, szerepváltásokra vagyunk kíváncsiak (ezek átlaga, konvex kombinációja nem feltétlenül rendelkezik a tulajdonsággal, hogy mindenképpen optimális megoldás legyen). A robusztus megoldással azonban előállítható egy olyan megoldás, mely optimális minden realizáció mentén. Ezt az alábbi eljárással érjük el.

A feltételek között rögzítjük a célfüggvény értékét is, majd különböző, újabb célfüggvényekkel újabb és újabb megoldásokat generálunk, azaz a poliéderünk újabb és újabb csúcsait keressük meg. Így olyan megoldásokat találhatunk, melyek mind azonos célfüggvény-értéken szerepelnek – és ezek konvex kombinációja szintén robusztus optimuma lesz az eredeti rendszernek.

E technika segítségével – megtartva a solver által szolgáltatott eredeti optimális megoldás mértékét – az alábbi megoldást nyerjük:

Kirendeltség	K	VVT	BT	BK	SZEREP
102	256	14	8	30	MOZGÁSTÉR
106	313	52	28	85	ALSÓ
109	53	48	5	36	MOZGÁSTÉR
110	80	145	8	51	ALSÓ
112	113	25	6	45	MOZGÁSTÉR
113	151	10	31	50	MOZGÁSTÉR
114	141	14	6	26	ALSÓ
115	282	20	11	68	MOZGÁSTÉR
116	73	120	7	36	ALSÓ
118	123	5	6	5	ALSÓ

17. táblázat. Irányítandó létszámok és szerepek a kevert robusztus megoldásban

Megfigyelhető, hogy ebben az esetben nem keletkeztek túlterhelt kirendeltségek (minden kirendeltség az alsó korlátja felé mozdult, vagy mozgástérrel rendelkezett). Az eszközök közül – a projekt sajátja ez – a képzéseket favorizálják. A bértámogatás ezzel ellentétben kifejezetten kis mértékben szerepel a megoldások között.

4.6.3. A feladat megvalósításának bemutatása

A program jelenleg a Budapest Főváros Kormányhivatala Munkaügyi Központjának tulajdonát képezi, belső használatra megírva és az adott problémára, projektre aktualizálva. A szoftverhasználati jogszabályok miatt a jelenleg tulajdonban lévő EXCEL[®] programot használtuk a program megírására, azon belül a szimulációkat és a teljes vezérlést makrók segítségével oldottuk meg.

A makrosítás mellett egy vezérlő panelt alkalmaztunk, melyben a felhasználók be tudják állítani, hogy mely paramétereken szeretnének véletlenítést kérni, illetve ott milyen tartományban tekintsünk egyenletes eloszlású véletlen változókat.

Az eredményeket számszerűsítve szolgáltatjuk a felhasználónak, tájékoztatást kap a kirendeltségenkénti eszközhasználat optimális létszámadatairól, összkirendeltségi létszámokról, a célfüggvény várható mozgásáról, alakulásáról és a kirendeltségek várható pozíciójáról (alsó vagy felső korláton tartott vagy mozgástérrel bíró kirendeltség).

Vezérlési beállítások

Az alkalmazásban a vezérlő lapon tudunk minden beállítást eszközölni. A fenti már említett összes ponton alkalmazhatunk véletlenítést (kiválasztva természetesen azokat a pontokat, ahol kérjük azt). Beállíthatjuk a szimulációk számát, továbbá átállíthatjuk a szimulálandó tényezők eloszlásának középvértékét és terjedelmét (az eloszlás mindenképpen egyenletes).



17. ábra. Az alkalmazás nyitófelülete – EXCEL[©]

Az első ábrán azt láthatjuk, hogy 1 és 0 kódokkal tudjuk vezérelni, hogy szeretnénk-e véletlenítést az adott paraméteren vagy sem. Továbbá az első lehetőségünk a 4 aktív eszköz esetén megadható ár, melynél – amennyiben 1-est választunk, úgy adott középvérték melletti, adott sugárban generál egyenletes pszeudo-véletlen számokat a rendszer (illetve összességében véletlen vektorokat). Ha 0-s értéket használunk, úgy a középvértéket használja a program.

Az eszközök között a vállalkozóvá válási támogatás (miután az a minimálbérhez kötött) alapbeállításként nem véletlen – hiszen ez a paraméter nem változik túl sűrűn.

SIKERESSÉGI RÁTA															
102				106				109				110			
K	VVT	BT	BK	K	VVT	BT	BK	K	VVT	BT	BK	K	VVT	BT	BK
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

RÁTA KÖZÉP															
102				106				109				110			
K	VVT	BT	BK	K	VVT	BT	BK	K	VVT	BT	BK	K	VVT	BT	BK
0,85	0	0	0	0,92	0	0	0	0,88	0	0	0	0,9	0	0	0
0,85	0,8	0,85	0,85	0,92	0,85	0,92	0,92	0,88	0,82	0,88	0,88	0,9	0,8	0,9	0,9
0,4	0,4	0,5	0,7	0,41	0,41	0,5	0,7	0,4	0,4	0,5	0,7	0,55	0,55	0,5	0,7

RÁTA SUGÁR															
102				106				109				110			
K	VVT	BT	BK	K	VVT	BT	BK	K	VVT	BT	BK	K	VVT	BT	BK
0,01	0	0	0	0,01	0	0	0	0,01	0	0	0	0,01	0	0	0
0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01
0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01

18. ábra. Az alkalmazási felület véletlenítési lehetőségének beállítása – EXCEL[©]

A megtérülési ráták beállítása esetén egy $p \in (0, 1)$ érték adandó meg (középvérték) és egy sugár (r), melyet úgy kell beállítanunk, hogy $p \pm r \in (0, 1)$ teljesüljön.

34. Megjegyzés. Ez utóbbi, $p \pm r \in (0, 1)$ feltételre azért van szükség, hogy a véletlenítés hatására se alakulhasson ki pl. 101%-os teljesítmény valamely kirendeltségen egy $p = 0,95$ és $r = 0,6$ -es érték megadásával.

Amennyiben az adott ponton nem kérünk véletlenítést, úgy a program automatikusan a középértéken fogja az optimalizálást végrehajtani.

Így tehát az alábbi eljárást kell követnünk:

1. Először azt kell eldönteni, hogy az adott pozíción szeretnénk-e véletlenítést vagy sem (alapbeállításként mindenhol kérünk, bár az első sorban csak a képzéseknél van értelme kérni – a többit azonban úgyszólván 0-val pótolja a program, tehát nem okoz gondot az 1-es kódok használata).
2. Másodsor azt kell megadnunk, hogy milyen középértékek mellett értékelünk, és azok milyen ingadozását engedélyezzük eszközönként, kirendeltségenként. Itt bármely pontra bármely érték megadható – nyilván a fenti $p \pm r \in (0, 1)$ megkötés betartásával.

POLITIKAI ALSÓ KORLÁTOK				LÉTSZÁMKÖZEPEK	
102	Képzés		0	102	0,00
	VVT		0	106	0,00
	Bértám		0	109	0,00
	Békköltség		0	110	0,00
106	Képzés		0	112	0,00
	VVT		0	113	0,00
	Bértám		0	114	0,00
	Békköltség		0	115	0,00
109	Képzés		0	116	0,00
	VVT		0	118	0,00
	Bértám		0		
	Békköltség		0		

19. ábra. Létszámközépek és politikai korlátok – EXCEL[©]

A harmadik pont igazából részint politikai, részint gazdálkodási ismereteinken alapul, hiszen előírhatnak olyan helyzeteket, amikor valamely eszközre mindenképpen el kell érniük egy adott kirendeltség keretei között egy minimális értéket (pl. a diplomások adott hányadát képzésre szeretnénk küldeni).

Továbbá előzetesen tájékoztunk arról, hogy a projekt célcsoportjai milyen létszámban képviseltetik magukat egy-egy kirendeltség illetékességi körzetében: nyilván csak egy részük vonható be, tehát nem érdemes egy adott ráta vagy létszám fölé tervezettni egyik kirendeltséget sem.

A solver beállításai

Az EXCEL[©] program beépített megoldójához nem nyúltunk hozzá, egészen pontosan a makró azt nem befolyásolja. A fent megadott lineáris rendszert optimalizálja a program (az eredmények ismertetésénél fogjuk specifikálni az aktuális korlátozó feltételeket). Ezen túlmenően semmi más beállítást nem alkalmaztunk. A különböző beállítások tehát mind a fenti leírásnak megfelelő módon történtek (az egyetlen kivételt a nemnegativitási feltétel adta, hiszen a kötelező politikai korlátok bevételeivel eleve adott volt ezen feltétel).



20. ábra. A megoldó beállításai, feltételek – EXCEL[©]

Látható, hogy milyen paraméterezés mellett futtattuk a megoldót, illetve a feltételeket milyen módon adtuk meg a program számára. A soron következő ábrán a futásidőre, illetve egyéb számítási beállításokra vonatkozó feltételeket mutatjuk be.



21. ábra. A megoldó beállításai, a futtatás jellemzői – EXCEL[©]

A fenti beállítások mellett végeztük el a szimulációkat, minden szimuláció alatt állandó paraméterezéssel és beállításokkal.

Az eredmények listázása

Ezen a ponton a reprodukálhatóság már nem játszik akkora szerepet, hiszen a kiíratás már egyéni igények kielégítéséről szól, a különböző adatok összeszervezése, kiértékelése már mind a megrendelőn múlik. Esetünkben az alábbi megoldást választottuk:

113	114	115	116	118	
62,5	74,8	79,4	83,1	94,5	Alsó korláton
18,6	13,3	1,4	8	3,2	Felső korláton
18,9	11,9	19,2	8,9	2,3	Közötte

106			109			110				
VVT	BT	BK	K	VVT	BT	BK	K	VVT	BT	BK
0,0	46,7	37,0	98,3	0,1	23,7	14,8	220,9	0,0	32,6	21,7
462,8			136,9			275,2				
500			150			300				
234,3			89,7			151,6				
KÖLTSEG			KÖLTSEG			KÖLTSEG				
438 064,9			124 668,0			258 398,0				

22. ábra. Az eredmények listázása kirendeltségenként, eszközönként – EXCEL[©]

Az első táblázatokban a kirendeltségi adatok láthatók: felül az adott pozícióban tartózkodás aránya az összes futásban (azaz az adott kirendeltség hányszor volt a lehetséges terheltségének maximumán vagy minimumán – vagy egyik pozíciót sem fogta, mozgástere volt).

Továbbá minden kirendeltség esetén megmutatja a program, hogy az egyes eszközökre átlagosan hány főt vezényeltünk, illetve alatta azt is láthatjuk, hogy előzetesen mi mennyit terveztünk.

Futásszám	1000
Célfüggvény paraméterek	
Átlag	2 250 433,502
Szórás	4 913,109
Relatív szórás	0,218
Ferdeség	14,849
Csúcsosság	244,114
Normalitás teszt Ferdeség alapján	191,985623
Normalitás teszt Csúcsosság alapján	1579,672907

SES	0,077344
SEK	0,154534

23. ábra. Az eredmények listázása: a célfüggvény jellemzői – EXCEL[©]

A célfüggvényünkről is kapunk egy gyors leíró statisztikai listát, ahol a ferdeséget és csúcsosságot azért kérjük, mert ebben az esetben nagyon nem szeretnénk, ha normális eloszlású mutatót kapnánk: alapvetően csúcsos és nagyon ferde eloszlásra számítunk – melyet a két paraméter standard hibájának meghatározásával könnyedén tesztelni is tudunk, hogy eltérnek-e a normális eloszlás 0-0 párosától.

4.6.4. Szimulációink eredményei

Az alkalmazást az alábbi paraméterekkel rendelkező számítógépen teszteltük: 2 GB RAM, 2×2 , 2,27 GHz-es Intel processzoros Toshiba laptop. 1000 generátumot készítettünk.

Az alábbi beállítások mellett végeztük a szimulációkat: az elkölthető pénzmennyiség maximuma 2,5 mrd Ft volt, melynek segítségével legalább 2600 főt kellett elhelyezni a programban. Az árak és a kirendeltségi, előre megadott, célcsoportokból származtatott létszámok a robusztus megoldáshoz képest nem változtak.

Politikai korlátokat nem állítottunk fel, illetve minden kirendeltségen betartottuk, hogy az előzetesen vállalt létszámhoz képest $\pm 10\%$ -ot eltérhetünk. A megtérülési ráták megadása esetén egy $p \in (0, 1)$ érték adandó meg (középérték) és egy sugár (r), melyet úgy kellett tehát beállítanunk, hogy $p \pm r \in (0, 1)$ teljesüljön.

A kérdés elsősorban a megbízhatóság, az érzékenység tesztelése volt: a kirendeltségek milyen arányban, miként „váltak szerepet”. Az előzetesen bekért létszámvállalások tekintetében mely kirendeltség hová kerül általában a $\pm 10\%$ -os sávban (alsó vagy felső korlátra, vagy inkább köztes állapotban maradnak, és így szabad mozgástérrel rendelkeznek).

A szimulációkat – megkönnyítendő a szervezést – úgy oldottuk meg, hogy a lehetséges szimulációs pontok méretének megfelelő, 50.000 soros véletlen-vektor táblázatot illesztettünk egy új munkalapként a megoldót tartalmazó EXCEL[©] dokumentumba. E táblázat elkészítéséhez az első fejezetben tesztelt algoritmust használtuk, ugyanis az EXCEL[©] korábbi verziójának véletlen számokat generáló függvénye hibás – javítását a kiadott szervízcsomagok nem oldják meg, hanem külön javítást kell hozzá keresni a gyártó internetes elérhetőségein.

Az összesítés ezek után aszerint történt, hogy a futások arányában hányszor szerepelt egy kirendeltség a felső és hányszor az alsó korláton (illetve hányszor közöttük).

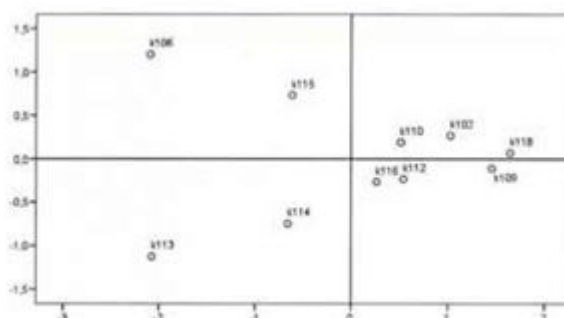
Kirendeltség kódja	Alsó korláton (%)	Felső korláton (%)	Közötte (%)
102	90,3	2,6	7,1
106	68,9	0,1	31
109	92,6	4,9	2,5
110	86,2	4	9,8
112	85,3	7,3	7,4
113	63	18,5	18,5
114	74,8	13,3	11,9
115	79	1,5	19,5
116	83,1	8	8,9
118	94,5	3,2	2,3

18. táblázat. A kirendeltségi szerepek alakulása a szimulációs eljárásban

35. Megjegyzés. A táblázatban tehát az látható, hogy 1000 szimulációt futtatva az esetek hány százalékában kerültek az egyes kirendeltségek alsó vagy felső korlátra, illetve hányszor volt mozgástérük.

Megfigyelhető, hogy – több kirendeltséggel együtt – például a Katona József utcai 102-es kirendeltség általában az alsó korlátra került, tehát esetükben általában a projektben való minimális részvételt várjuk el. A többi kirendeltség – mely nem alsó korláton van – jellemzően felső korláton található. A legnagyobb mozgástere, úgy fest, hogy a 106-os, Baross utcai, a 115-ös, Kassai utcai, illetve a 113-as, Heltai Jenő téri kirendeltségeknek van, hiszen itt a legmagasabb a két határ közötti ingázás aránya – bár a János utcai 114-es kirendeltség esetén is 10% feletti lett a mozgástérbeli szerep részaránya.

A kirendeltségeket egymással is összehasonlíthatjuk, amire a klasszikus MDS jó választásnak tűnik. Az ábrázoláshoz az SPSS[©] programcsomag megfelelő rutinját alkalmaztuk. Az arányok, illetve létszámok okán nem volt jelenleg indokolt más távolság vagy metrika alapján ábrázolást választanunk, így részint a mozgástérre, részint majd a létszámokra is 1-1 ábrázolás segítségével vizsgáljuk meg, hogy mely kirendeltségek tűnnek hasonlóaknak – és melyek azok, ahol a terheltség vagy létszám alapján nagyfokú eltérések mutatkoznak.



24. ábra. A kirendeltségi szerepek ábrázolása az MDS-eljárás segítségével – SPSS[©]

Vizsgáljuk meg a kirendeltségi szerepeket: a 106-os, Baross utcai és a 113-as, Heltai Jenő téri kirendeltség különbözik leginkább a többi kirendeltségtől, majd utána a 115-ös, Kassai utcai, illetve a 114-es, János utcai kirendeltség következik. Az első kettő esetén (106-os és 113-as) megállapítható, hogy a különbséget a többiektől a nagyon alacsony alsó szinten való tartózkodás jelenti, így ebből azt is levonhatjuk következtetésként, hogy a vízszintes dimenziót elsősorban az alsó korláton való tartózkodás mozgatja (az alacsony abszcissa alacsony alsó korláton való szerepeltetést jelent).

A függőleges dimenzió interpretálása nem tűnik ennyire egyértelműnek, így inkább a kirendeltségek egymáshoz való viszonya látszik lehetséges járható útnak az ábra értelmezésében: a 106-os kirendeltség esetében a többiektől való jelentős eltérést a mozgástérrel való rendelkezés nagy aránya indokolja, míg a 113-as esetében az, hogy a mozgástér és a felső korláton tartózkodás aránya egyaránt magas – és ez jelentősen eltér a többi kirendeltségétől. A 115-ös és 114-es kirendeltségek kevesebb mozgástérrel vagy leterheltséggel bírnak, mint az előbb említett két kirendeltség, azonban még így is több alternatív szerepkörük adódik, mint a többi 6 kirendeltségnek, melyek szerepe az esetek több mint 80%-ában az alsó korlátot jelenti.

A szerepek tehát általában stabilnak mondhatók a legtöbb kirendeltség esetén – az említett 4 kirendeltségnél viszont fennállhat annak veszélye, hogy nagyfokú bizonytalanság, illetve nagyobb létszámmozgások, ingadozások esetén egyik szerepből a másikba kell kényszerülniük a gazdaságosabb működés érdekében.

36. Megjegyzés. Az MDS-eljárás esetén nem feltétlenül szükséges a dimenziók, koordináták értelmezése, hiszen tudjuk, hogy forgatás és tükrözés invariáns megoldás. Akadnak persze olyan esetek, amikor a kialakult irányoknak jelentést, interpretálást tudunk adni – mint az első esetben az abcisszáknak.

Irányítandó létszámok (kerekítve)

Megvizsgáltuk azt is, hogy kirendeltségenként átlagosan hány főt küldenénk valamely eszközre. Az eredmények azt mutatják, hogy a teljes létszámkeretet nem használjuk ki. Ez azt jelenti, hogy összességében (átlagban) néhány fővel a vállalt 2600 fő alá fogunk kerülni.

Az összes elköltött pénzünk így 2250,4 millió Ft volt, azaz alig valamivel különbözik a robusztus megoldás eredményétől (amikor a legolcsóbb támogatás is 0,5 millió Ft, akkor 0,4 millió Ft-os differenciát nem tekinthetünk érdemi különbségnek).

37. Megjegyzés. Ez azt jelentette, hogy a szimulációk során megnéztük, hogy mely kirendeltségen mely eszközre hány fős létszámot tervezett be a program szimulációként, majd utána ezen értékeket átlagoltuk. Világos, hogy így nem kaphatunk olyan megoldást, mely bármely szimuláció esetén megengedett megoldás lenne, azonban nem is ez volt a célunk – hiszen azt a robusztus megoldás szolgáltatta.

Kirendeltség	K (fő)	VVT (fő)	BT (fő)	BK (fő)
102	160,3	2,5	85,9	25
106	379,1	0	46,7	37
109	98,3	0,1	23,7	14,8
110	220,9	0	32,6	21,7
112	75,9	1	90,9	16,7
113	183,6	0	35,3	20,4
114	113,7	0	50,3	23,4
115	240,6	0	59,4	49,4
116	181,9	2,7	29,6	16,7
118	100,1	0	23,4	12,8

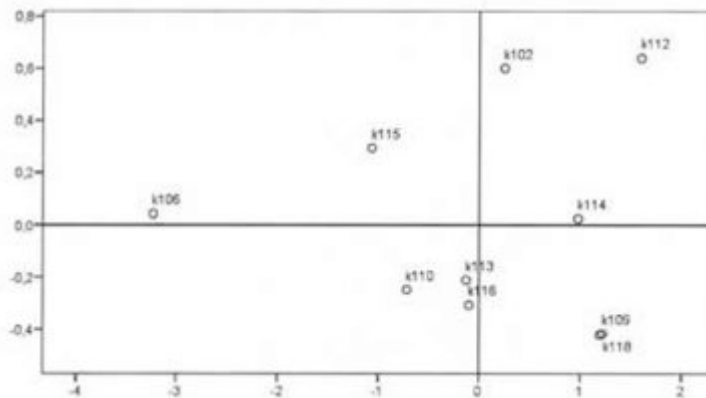
19. táblázat. A szimulációs létszámeredmények átlagai, kerekítve

Az irányadó létszámok alapján némiképpen más kép rajzolódik ki, mint a kevert robusztus megoldás esetén.

Az ábrán megfigyelhető, hogy a 106-os kirendeltség itt is jelentős eltéréseket mutat a többi kirendeltséghez képest. Természetesen a többi kirendeltség között is maradnak még értelmezendő eltérések.

Értelmezésben leginkább a képzésekre irányítandó létszám tűnik először a legegyszerűbbnek: a vállalható létszámok a képzéseken a 106-os, Baross utcai kirendeltségen kiemelkedően magasak a többiekéhez képest.

A függőleges irányban a legnagyobb szerepet a bértámogatás, illetve bérköltség valamilyen kombinációja tölti be az adatok megfigyelése alapján: a legmagasabban a 112-es, Ráday utcai kirendeltség áll, majd utána a 102-es, Katona József utcai, a 106-os, Baross utcai és a 114-es, János



25. ábra. A kirendelési létszámeredmények szerinti osztályozás MDS-eljárás segítségével – SPSS©

utcai kirendeltségek következnek. Ezek azok a kirendeltségek, ahol e két eszköz összességében jelentős mennyiségű támogatást igényelhet, illetve oszthat szét.

Az érdekesség azonban inkább abban van (vezetői szemmel nézve), hogy az ábrán jól látható módon a 109-es, Mansfeld Péter utcai (csepeli) kirendeltség és a 118-as, Haller utcai (hajléktalan) kirendeltség lényegében azonos szerepkörben van – és a létszámokat tekintve is igen-igen közel kerülnek egymáshoz, szinte nincsen különbség a két kirendeltség között.

Hasonló mondható el a 113-as, Heltai Jenő téri és a 116-os, Darus utcai kirendeltségekről is.

Azonban e második ábrázolásból nem a szerepek átmozgatása tetszik ki, hanem az ügyintézők leterheltsége lehet kérdéses. Egy-egy eszköz működtetése, annak adminisztrációja jelentős időt vehet igénybe – és ez aránytalanságokat szülhet. Jól látható, hogy a 109-es és 118-as kirendeltségek között ügyintézői létszámokban jelentős eltéréseket tapasztaltunk, miközben az eszközökre való toborzás közel azonos mennyiségű feladatot ad, így a leterheltségekben igen nagyfokú eltérések adódhatnak. Így vezetői szempontból e második adatsor arra adhat útmutatást, hogy ezen eszközök ügyintézői létszámát milyen módon lehetne kialakítani egyes kirendeltségeken.

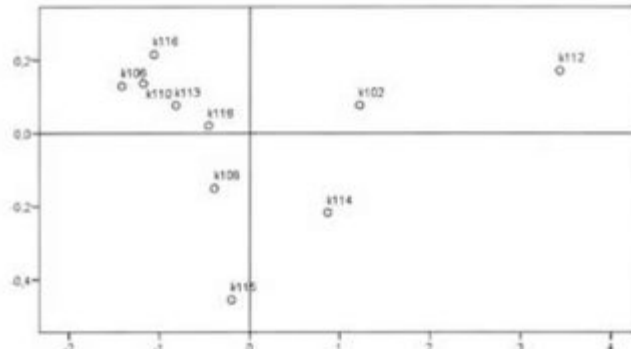
Az irányítandó létszámok arányai (kerekítve)

Érdekes lehet megfigyelni azt is, hogy mely kirendeltségek mely aktív eszközöket használják jellemzően. Ennek kapcsán megnéztük, hogy az adott kirendeltségi létszámok arányai miként alakultak – majd megpróbáltuk tipizálni az így kapott kirendeltségi szerepeket.

Kirendeltség	K (%)	VVT (%)	BT (%)	BK (%)
102	59	1	31	9
106	82	0	10	8
109	72	0	17	11
110	80	0	12	8
112	41	0	50	9
113	77	0	15	8
114	61	0	27	12
115	69	0	17	14
116	79	1	13	8
118	73	0	17	10

20. táblázat. A szimulációs eredmények létszámarányai eszközönként, kirendeltségenként

A szerepeket itt is a klasszikus MDS megoldása alapján próbáljuk értelmezni, interpretálni.



26. ábra. A szimulációs létszámarányok ábrázolása MDS-eljárás segítségével – SPSS[©]

Az ábrázolás alapján elmondható, hogy alapvetően a képzést és a bértámogatást látjuk vízválasztónak: a pozitív ortánsban ábrázolt, leginkább kiugró 112-es, Ráday utcai kirendeltség esetében a bértámogatás mindenkinél lényegesen magasabb aránya mutat másfajta viselkedést, míg pl. a bal felső negyedben lévő 5 kirendeltség nagyon hasonlóan, elsősorban képzésekkel (és a másik két, a bértámogatás és a bérköltség eszközben arányaiban egymáshoz közeli mértékkel) való munkavégzése lesz jellemző.

Az arányok segítségével egyfajta tipizálást, profilt tudunk a kirendeltségeknek adni – úgy is értelmezhetjük, hogy a kirendeltségek esetében elsősorban a sikerességi ráták voltak meghatározók: mely kirendeltségeken mely eszközök voltak leginkább megvalósíthatók (a többi eszközhöz képest). Ebből azt is leszűrhetjük, hogy a kirendeltségek zöménél elsősorban a képzések menedzselése volt a leginkább sikeres, itt tudták elsősorban teljesíteni az elvárt rátákat.

4.6.5. A robusztus megoldással való összehasonlítás

A célfüggvény értéke tehát, ha e fenti létszámmegoldást választjuk, 2250,4 millió Ft.

Ki kell emelnünk azt is, hogy ezzel is bőven alatta maradtunk a projektben felhasználható 2,5 mrd Ft-os keretnek. Hasonlóan a robusztus megoldás esetén tapasztaltakkal, itt is csak három eszközt használunk fel érdemben – a vállalkozóvá válási támogatás itt sem volt kedvelt eszköz. Ellenben a bértámogatás és bérköltség támogatási formák lényegesen többször kerültek elő megoldásként – ha úgy tetszik, a robusztus megoldással ellentétben nem az önálló megoldások irányába mozdítja el az álláskeresőket ez a modell, hanem a munkáltatók támogatása felé.

Fontos lehet kiemelni azt is, hogy e megoldás sajátossága az is, hogy ettől teljesen függetlenül alakult úgy a fővárosban, hogy a vállalkozóvá válási támogatás háttérbe szorult az egyéb aktív eszközök mellett – lényegében nem nagyon lehet már ezt az aktív eszközt megszerezni.

4.7. Felhasználói tapasztalatok, összegzés

Több lehetséges megoldást is kínálunk a felhasználók számára: egy robusztus megoldásból kevert optimális megoldást, illetve megmutatjuk számukra azt is, hogy ha a különböző realizációkat figyelembe vesszük, akkor milyen szerepváltások lehetségesek, mely kirendeltségek tekinthetők kockázatosnak.

A projekt teljes költségvetésének mindösszesen 90%-át használjuk fel, tehát bőven maradnak tartalékaink esetleges hirtelen jövő igények kielégítésére vagy korrekciókra.

A felhasználóknak mindezt egy könnyen használható, általánosan fellelhető alkalmazásban is át tudjuk adni:

- Kis módosítással (maximum 1-2 munkanapnyi munkával) át lehet írni hasonló projektekre az alkalmazást.
- A Központ által birtokolt szoftverrel dolgozunk (Microsoft[©] Excel[©] volt alkalmazható az informatikai felmérés után).
- Több ponton is módosítható és állítható a paraméterezés. Például a pénzkeretek változása esetén, figyelembe véve az eddigi eredményeket, kevés változtatással újra futtatható az alkalmazás. Világos, hogy ilyenkor a struktúra nem, a korlátozó feltételek azonban változnak.

Ezen szempontokat figyelembe véve a megírt alkalmazás elfogadható, hiszen a benne használt eljárás minden közigazgatásban megtalálható rendszerben fellelhető, az átparaméterezés egyszerű, a struktúra – a helyi jellegzetességek beépítése után – nem változik, és újabb projektek esetén vagy azonos projekt későbbi lépésében strukturális változtatás nélkül bevethető.

Érdekességként emelnénk ki, hogy a robusztus eljáráshoz képest a szimulációs megoldás más-fajta csoportosítást, másfajta eszközhasználatot kínált fel, lényegében hasonló mennyiségű pénzt költve, általában elfogadható szinten tartva a kijelölt sarokszámokat. A robusztus megoldásból kevert, azok konvex kombinációjaként előállított megoldással pedig kiküszöböljük azt, hogy túlterhelt vagy alulterhelt kirendeltségek/eszközök legyenek jellemzően a rendszerben.

5. Összefoglalás

A dolgozat során több, egymástól jól szeparálható területtel foglalkoztunk, melyek egy alkalmazás során egységként, egymást segítve kerültek felhasználásra egy nagyobb probléma megoldásakor.

A munkafolyamat során egy pszeudo-véletlen, egész számokból álló vektorokat generáló algoritmust javítottunk és teszteltünk. Előre meghatározott – általunk kitalált vagy vektorokra adaptált – tesztekkel összehasonlítottuk egy bevizsgált, ismert algoritmussal (a MATLAB[©] véletlen egész vektorokat generáló algoritmusával).

Az eredeti probléma megoldásának második lépéseként az adatok, eredmények interpretációjában fontos szerepet játszó multidimenzionális skálázás vizsgálata során akadtunk olyan kérdésekre, melyek megválaszolásával új eredményeket tudtunk e módszer már eddig is széles spektrumához fűzni. Ezt az eredendően statisztikai módszert optimalizálási feladatként felírva lényegesen általánosabb értelemben tudtuk vizsgálni, mint a statisztikai általánosításokban tapasztalható eljárások.

A harmadik fázisban az eredeti probléma matematikai viselkedésére, strukturális sajátosságaira voltunk kíváncsiak. A felírásból kiderült, hogy klasszikus folyamfeladatként való felírásakor a feladatnak sajátosságos, későbbi felhasználói döntéseket is komolyan befolyásoló tulajdonságai vannak. E strukturális sajátosság miatt szerettük volna a feladatban meglévő bizonytalanságot érzékenységvizsgálatnak alávetni, melyet sztochasztikus eszközökkel szerettünk volna kezelni és megoldani. Ezt sikerült, ha nem is mindenre kiterjedően, de a felhasználók számára mindenképpen megnyugtató szinten kezelnünk.

Így a feladatot megoldottuk, a közben használt módszerekhez kapcsolódóan a szakirodalomban eddig nem szereplő új megközelítéseket dolgoztunk ki, ezekhez kapcsolódó új vizsgálatokat végeztünk el. A feladat megoldását pedig a feladatot kitűző felhasználók azóta is alkalmazzák.

Tételesen az alábbi eredmények születtek:

1. Tesztfejlesztések generátumokhoz
2. Pszeudo-véletlen egész számsorozatok generáló algoritmus fejlesztése
3. A multidimenzionális skálázás (MDS) általánosítása
4. Az MDS egy új alkalmazási területen való bemutatása
5. Egy hierarchikusan működtetett projekt strukturális vizsgálata és a tapasztalt jelenségek általánosítása
6. Robusztus optimalizálásban a fejlesztett generátor alkalmazása
7. Az eredmények interpretációjában az MDS alkalmazása

A fenti pontokban elért eredményeket részletezzük egy-egy alfejezetben, melyekben csak az eredményekre támaszkodunk, hosszabb részletezéseket már nem teszünk, hiszen a dolgozat korábbi részein az eredmények teljes kibontása, részletezése megtörtént.

5.1. Pszeudo-véletlen generátorban elért eredmények

Pszeudo-véletlen számokat vagy vektorokat generáló algoritmusokat számtalan helyen használnak, alkalmaznak. Ezek minőségbiztosítása fontos szempont egy-egy algoritmus alkalmazhatóságához.

A dolgozat első fejezete a minőségbiztosítási munkafolyamat leírása egy konkrét algoritmus fejlesztése esetén. A dolgozatban ismertetett eredmények az alábbi pontokban szedhetők össze:

- Egy pszeudo-véletlen egész vektorokat generáló algoritmus vizsgálatához adaptáltunk pszeudo-véletlen sorozatok tesztelésére fejlesztett eljárásokat.
- Az előre meghatározott teszteken az általunk vizsgált és fejlesztett algoritmus nem teljesített gyengébben, mint egy már ismert és elfogadott – a MATLAB[©] – algoritmus.
- Véletlen számsorozatra alkalmazott algoritmusokat adaptálva és néhány új tesztet is vizsgálva olyan véletlen egészeket generáló algoritmust alkottunk, melynek ciklushossza szabadon növelhető. Ezzel elértük, hogy a pszeudo-véletlen sorozatunk tetszőlegesen ritkán szolgáltatson önismétlést (befolyásolhatjuk az algoritmusunk ciklushosszát). Az eredményeket publikáltuk, az elemzés részleteit lásd cikkünkben [49].

A fenti eredményeket az alábbiak alapján részletezzük, illetve részleteztük a dolgozatban:

5.1.1. Tesztadaptáció

A tesztek, melyeket többek között Knuth könyvében [18] vagy Marsaglia munkáiban [27] is megtalálhatunk, pszeudo-véletlen sorozatok véletlenszerűségének tesztelésére alkalmazzák, így esetünkben változtatások nélkül nem feltétlenül alkalmazhatóak. A tesztek egy része a generátum eloszlását, míg egy másik, szintén jelentős része a generátumban esetlegesen fellelhető összefüggések tesztelését hivatott elvégezni.

5.1.2. Teszteredmények

A teszteredményeket egy cikkben foglaltuk össze, mely eredmények címszavakban az alábbiak:

1. Több dimenziót vizsgálva nem találtunk érdemi eltéréseket az algoritmus viselkedésében alacsonyabb vagy magasabb dimenzióban.
2. A ritkítás paraméterről (melyet az összefüggőség csökkentésére, illetve annak vizsgálatára alkalmaztunk), azt tapasztaltuk, hogy nem érdemes 8-nál nagyobbra választani. Alacsonyabb ritkítási mérték esetén (pl. minden második vektor kiírása esetén) még érzékelhető, mérhető összefüggések mutathatók ki. Minden 8. vektort véve ez a belső összefüggése a generátumoknak lényegében elhanyagolható, nem kimutatható. Nagyobb ritkítással dolgozva az algoritmus egy másik paramétere szenved csorbát: lassabb működéssel fizetünk, miközben a minőség már érdemben nem növekedik.

3. Tapasztalataink szerint az egyenletesség tesztelésére – ezen algoritmus esetén – a Kolmogorov-teszt hatékonyabbnak bizonyult, mint a χ^2 -statisztika. Az összefüggőség tesztelésére legerősebbnek a Knuth könyvében [18] ismertetett sorozat-teszt mutatkozott.

Mindkét teszt esetén mind a MATLAB[©], mind az általunk vizsgált algoritmus 5% körüli eredményeket ért el.

4. A tesztek előre határoztuk meg, tehát az algoritmuson végrehajtott módosításokat, vizsgálatokat az előre meghatározott teszteken való megfelelés inspirálta, indokolta. A módosítások folyamata, indokai az említett dolgozatban [49] megtalálhatók.

5.1.3. Publikáció

Az eredeti algoritmus több helyen is megtalálható – lásd Vizvári dolgozatát [51] vagy Ramirez könyvét [37]. Vizsgálatunk és módosításaink eredményeit cikkünk [49] tartalmazza.

5.2. Az MDS általánosításában, vizsgálatában elért eredmények

Az MDS-t számos tudományterületen alkalmazzák, számos formában történt már meg az általánosítása. A dolgozatban ismertetésre került, [31] cikkben leírt területről nem találtunk alkalmazást. Továbbá nem találtunk olyan irányból megközelített általánosítást sem, amit a dolgozatunkban vázoltunk.

A dolgozatban ismertetett eredményeink főbb pontokban az alábbiak:

- A multidimenzionális skálázást euklideszi távolságtól eltérő távolságok esetére adaptáltuk az eddig általánosan használtaktól eltérő módszerekkel és megoldásokkal. A szakirodalomban vázolt Kruskal-féle robusztus vonaltól teljesen eltérő módon próbáltuk megoldani az eljárás általánosítását.
- Vizsgáltuk az eljárásunkat különböző struktúrákon és eltérő távolság-definíciókkal. Korábbi feladatok megoldásának kritikai eszközeként szolgált dolgozatunkban [31], illetve a fontossága miatt végig szem előtt tartott eredeti probléma megoldásában interpretációs eszközként felhasználtuk az eredeti MDS-eljárást. Hasonlóan interpretációs eszközök bemutatása, tesztelése volt azon dolgozat [46] célja is, melyben részben az MDS ötletén elindulva klaszterezési eljárásokon hajtottunk végre módosításokat.

5.2.1. Az általánosítás és új területen való alkalmazás eredményei

Az általánosítást l_1 és l_∞ távolságok esetére írtuk fel és vizsgáltuk. Vizsgálatainkat több, előre fixált helyzetben végeztük, ahol a pontok eredeti elhelyezkedését ismertük, így a módszerek helyreállító képességét könnyebben vizsgálhattuk.

E vizsgálatra azon dolgozat [31] fő témája miatt volt elsődlegesen szükségünk, amelyben különböző objektumok távolságaival dolgoztunk. E munkák során a felmerülő anomáliák kiszűrésére törekedtünk, illetve próbáltuk azok hátterét megérteni.

Tapasztalataink szerint ugyanis előfordulnak olyan helyzetek, amikor egy-egy módszerhez készítenek alkalmazást – azonban olyan adatokkal történik az alkalmazás megvalósítása, melyek a valóságban nem fordulhatnak elő. Az MDS alkalmazásával olyan területekre tudunk rámutatni az említett cikkben, melyek esetén – a megadott távolságok alapján – nem tudunk olyan objektumokat mutatni, melyek a megadott távolságokkal bírnának.

Az új területen való alkalmazáskor merültek fel az eltérő távolságokkal történő számítások, hiszen előfordulhat, hogy az objektumok által megadott távolságok nem euklideszi távolságok. Ilyen esetben torzabb képet kaphatunk az objektumok elhelyezkedéséről. Mindez indokoltá tette, hogy a probléma megoldása érdekében a módszert úgy próbáljuk meg általánosítani, hogy a távolságok paraméterként működjenek benne – még azon az áron is, hogy természetesen így lényegesen bonyolultabb és lassabban megoldható feladathoz jutottunk.

5.2.2. Interpretációs eredmények

Az MDS interpretációs tulajdonságait szintén számos helyről ismerjük. A dolgozatban több ponton is használjuk a módszert erre a célra. Elsődlegesen akkor, amikor előre megadott távolságok alapján szeretnénk a vélelmezhető objektumokat meghatározni, mint tettük azt [31] dolgozatban.

A másik lehetőséget az alkalmazások fejlesztésekor értük el: [46] cikkben nem az MDS, hanem egy MDS által is inspirált klasszifikációs, vizualizációs módszert ismertetünk egy pszichológiai alkalmazásban.

A harmadik terület az alkalmazásunkban található, ahol a célunk az volt, hogy a különböző kirendeltségek egymáshoz való viszonyát, egymáshoz való hasonlatosságát tudjuk érzékeltetni. Több változó alapján történő összehasonlításban – ilyen értelemben az MDS teljesen klasszikus felhasználásához térve vissza – ennek nagy haszna volt a felhasználók számára, nagyban megkönnyítve az eredmények értelmezését.

5.2.3. Publikációk

Az MDS klasszikus, illetve klasszikus értelemben vett általánosításának szakirodalma széleskörű. Így arra nem vállalkoztunk, hogy teljesen áttekintsük – azonban próbáltunk olyan szakirodalmakra rámutatni, melyben az általunk megtalált problémát orvosolhatták már.

Miután ilyenre nem találtunk, így [31] cikkben és [32] kutatási riportban egy újnak tűnő területen vetettük be a módszert. Ráadásul, mivel a módszerhez szükséges eredeti távolságokról csak az tűnt bizonyosnak, hogy nem euklidesziek, így az általánosításunk is adekvát volt.

Az MDS ötlete, tehát az objektumok közötti távolságok alapján történő klasszifikálás, természetesen számos eljárást inspirált. Ily módon egy pszichológiai területen újnak tűnő klasszifikációs eljárásnak is inspiráló lendületet adott az MDS általánosításában elvégzett munkánk – noha a dolgozatban ismertetett eredményeinkkel a [46] cikkben közvetlenül nem éltünk.

5.3. Alkalmazás-fejlesztés

A két előző fejlesztés részben arra irányult, hogy egy regionális pénzosztási feladatot modellezzünk, és optimális pénzosztási rendszert tudjunk vázolni a felhasználók felé.

A dolgozatban ismertettük azt a hierarchikus rendszert, mely általában sajátja egy kormányzati, pénzosztási rendszernek. Ennek a rendszernek több, strukturálisan fontos pontját sikerült matematikai eszközökkel feltárnunk – és annak közgazdaságilag is elfogadható jelentést adnunk. A rendszer matematikai sajátosságai nem megkerülhetők. A belőlük származó esetleges igazságtalanságok némi többletforrás bevonásával enyhíthetők, árnyalhatók.

Mind az árnyalásra, mind pedig a strukturális sajátosságok kivédésére elfogadható magyarázatokat és javaslatokat tettünk a felhasználó számára, így a projektet sikeresnek tekinthetjük.

Az elvégzett feladatok során az alábbi eredményeink születtek:

- Az eredeti feladatunk strukturális hátterét feltártuk, annak matematikai modelljét megvizsgáltuk, a feladatot megoldottuk – a megoldás kielégítette a megoldással szemben támasztott pénzügyi és politikai igényeket egyaránt.
- A feladat sztochasztikus modelljét felírtuk és megvizsgáltuk, robusztus megoldását találtuk az így nyert sztochasztikus rendszernek. E robusztus megoldást a szimulációs módszerekkel való összevetésben referenciaként tudtuk felhasználni, illetve belőle kevert robusztus megoldást tudtunk előállítani.

Ezen eredmények részletezését a dolgozat utolsó fejezetében tárgyaljuk, így most továbbra is csak emlékeztetünk rájuk, illetve kiemeljük a fontosabb végeredményeket (számszerűsítések nélkül).

5.3.1. A matematikai vizsgálat

A feladatot először optimalizálási feladatként írtuk fel, de annak strukturális sajátosságai nem voltak szembetűnőek, így átfogalmaztuk a feladatot, folyamfeladatként vizsgálva azt tovább.

Ekkor már tételesen igazolni tudtuk, hogy esetünkben – és minden ehhez hasonló, hierarchikusan szervezett, bizonyos (nem túl éles) feltételeket kielégítő rendszerben sajátos képe lesz a legolcsóbban üzemeltethető megoldásnak.

Igazoltuk, hogy a feltételek fennállása esetén szükségszerűen vagy teljesen leterhelt, vagy teljesen minimális működésre kényszerített állapotokban lesznek a pénzosztási rendszer szereplői – és csupán néhány kivételezett lehet abban a helyzetben, hogy érdemi döntéshozatalra legyen lehetősége (azaz: a hierarchikus rendszernek egyetlen ága lesz az, ahol nem terhelés vagy a minimalizálás lesz a feladat).

A rendszer ezen tulajdonsága determinisztikus esetben került előtérbe. Esetünkben több olyan pontja is van a struktúrának, ahol nagyfokú bizonytalanság lehetséges. Ezt a bizonytalanságot mi sztochasztikus eszközök bevonásával kívántuk kezelni (ez indokolta a véletlen, egész vektorok generálásának kérdését, amit meg kellett próbálnunk rendszertől független eszközökkel biztosítani, hiszen a felhasználó informatikai kapacitásai erősen végesek voltak – és jelenleg is azok).

A sztochasztikus eszközök bevonása újfajta strukturális kérdéseket vetett fel, melyeket szintén sikerült – ezen egyedi, speciális esetben – feltárunk. Prékopa robusztusság-definíciója [35] nyomán olyan megoldásokat próbáltunk keresni, melyek bármely bizonytalan esemény bekövetkezésekor megengedett megoldást szolgáltatnak. E megoldás a felhasználó irányában – bár szemmel láthatóan a legolcsóbb megoldást kínáltuk – elfogadhatatlan politikai áldozatokat hordozott magában (a robusztus megoldás mindenképpen éles preferenciákat adott egyes felhasználóknak és eszközöknek, amiknek hatására bizonyos eszközök teljesen lekerültek a palettáról, tiltólistás eszközökké váltak).

A robusztus megoldást így mint referenciát használtuk (ennél olcsóbb megoldását nem tudunk a rendszernek biztonságosan garantálni) az elvégzett szimulációinkhoz. Továbbá segítségével kevert robusztus megoldást tudunk előállítani – mely azonban lényegesen eltért a szimuláció által szolgáltatott megoldástól.

A szimulációk segítségével olyan megoldást alkottunk, mely várhatóan elfogadható lesz: a projekt által előírt korlátokat teljesíti, a megtérülési rátái megfelelőek (az elfogadható határon vannak), politikailag vállalható, és a robusztus megoldáshoz képest nem kerül lényegesen több pénzbe.

Itt fontos kiemelni, hogy a projekt teljes költségvetéséhez képest a robusztus megoldás lényegesen olcsóbb megoldás volt, így még a szimulációkkal elvégzett, nem optimális, de az esetek döntő többségében megengedett megoldás is bőven belül maradt a projekt teljes költségvetésén.

A robusztus optimum szükségszerűen olyan, hogy bármely, racionálisan elképzelhető, előzetes eredmények alapján feltételezhető realizáció esetén megengedett megoldások közül a legolcsóbb megoldás lesz. A segítségével előállított konvex kombinációk szintén hordozzák ezt a tulajdonságot.

Így az is világos, hogy a szimulációkkal nyert, kiátlagolt megoldásunk olyan, mely bizonyos realizációk esetén nem lesz megengedett megoldás, valamely feltétel sérülni fog (általában ez valamely szereplő megtérülési rátájának sérülését jelentette).

Fontos azonban kiemelni, hogy a teljes projekt megtérülési rátái nem kerültek veszélybe ilyenkor sem, tehát várhatóan maga a projekt sikeres lesz (ezt úgy lehet elképzelni, hogy valamely szereplő az elvárt alatt teljesít, viszont ilyenkor egy másik szereplő bőven az elvárt feletti értékeket produkálja).

5.3.2. Az alkalmazás bemutatása, publikációk

A feladat megoldása nem lett volna teljes, ha a felhasználóknak nem tudunk egy olyan eszközt biztosítani, melynek segítségével a jövőbeni hasonló feladatokat – kis módosítással – szintén fel tudják dolgozni. Bemutatjuk a felhasználó tulajdonában lévő program futás közbeni állapotait, illetve annak eredményeit is ismertetjük. A program specifikációit közzétettük, így reprodukálhatónak gondoljuk a számításainkat annak ellenére, hogy a program teljes működését, illetve forráskódját nem áll módunkban átadni.

Az alkalmazásban felhasználhatjuk az általunk tesztelt és kipróbált pseudo-véletlen vektorokat generáló rutint, továbbá az eredmények interpretálásában, vizualizálásában fontos szerepet tölt be az MDS is. Az általa szolgáltatott két dimenziós térképeken elhelyezett felhasználói pontok azok összehasonlíthatóságát, egymással való összevetését és értékelését teszik könnyebbé.

Eredményeinket – terjedelmi okokból vázlatosan, összefoglalva – [50] cikkünk tartalmazza.

5.4. Összesítés

Összességében elmondható, hogy az általunk vizsgált és módosított pseudo-véletlen egész vektorokat generáló algoritmus vizsgálatok új tesztek, illetve régi tesztek vektorokra való adaptációját sikeresen végrehajtottuk.

Az MDS széles körben használt eljárásához egy új területet sikerült találnunk, illetve annak általánosításaihoz egy újfajta irányt mutattunk.

Mindkét eljárás egy-egy fontos részét képezte egy gyakorlati alkalmazás megoldásának, így a két eljárás empirikus oldalról vizsgálva is jól teljesített. A felhasználók a számukra rendelkezésre álló, szűkösebbnek nevezhető informatikai háttér mellett is jól tudják alkalmazni az általunk fejlesztett rutinokat.

A vizsgálataink során minden esetben előre meghatároztuk azokat a kritériumokat, melyeknek szeretnénk volna megfelelni (vagy számunkra határozták meg a használok a különböző elvárásait), így a munkafolyamat során minden fejlesztés és javítás, módosítás ilyen irányból történt. A feladatok előre adottak voltak, amikhez alkalmazásokat kellett fejlesztenünk ([31] cikkünk után ezt talán méginkább fontos lehet kiemelni). Az empirikus teszteken a felhasználók által igénybe vett szolgáltatások megbízhatónak bizonyultak, az általuk szolgáltatott eredmények elfogadhatóak és a gyakorlatban megállják a helyüket.

5.5. Melléklet - NYILATKOZAT



Nyilatkozat

Igazolom, hogy Takács Szabolcs minden fejlesztési munkái, melyeket a Budapest Főváros Kormányhivatala Munkügyi Központ számára készített 2010. augusztus 1-je óta, a Központ tulajdonában vannak.

E fejlesztéseket részint a tervezési (közám- és/vagy keretgazdálkodás) folyamatokban kérésre illetve optimalizálási célokra, részint tanulmányok (munkaerő-piaci előrejelzések, hatástanulmányok, negyedéves-, félévi- és gyorstájékoztatók) megírása során alkalmaztuk, alkalmazzuk és a jövőben is terveink közt szerepel ezek alkalmazása, fejlesztése.

Budapest, 2012. 11. 06.



1082 Budapest, Kékvíz utca 11. – 1364 Bp. Pf. 244. – Telefon: +36 (1) 477-5700 – Fax: +36 (1) 477-5800
E-mail: fevarosah-mk@uh.hu – Honlap: <http://kormhivatal.munka.hu> / www.munka.hu

27. ábra. A Munkügyi Központ nyilatkozata

Hivatkozások

- [1] Ben-Tal, A.; Nemirovski, A.: *Robust optimization – methodology and applications*, Mathematical Programming, Vol. 92(3)., 453–480, 2002.
- [2] Besicovitch, A. S.: *Sets of fractional dimensions (IV): on rational approximation to real numbers*, Journal London Mathematics Society, 9, 126–131, 1934.
- [3] Ben-Tal, A.; El Ghaoui, L.; Nemirovski, A.: *Robust Optimization*, Princeton, Chapter 1, 2009.
- [4] Blake, F. B.; Schulze, S.; Hughes, M. J.: *Perceptual Mapping by Multidimensional Scaling: a Step by Step Primer*, Research Report in Consumer Behavior, Cleveland State University, 2003/09.
- [5] Chaitin, G. J.; Martin-Löf, P.: IEEE Transaction, IT-14, 662–664, 1968.
- [6] Church, A.: *On the concept of a random sequence*, American Mathematical Society Presentation, 1939.
- [7] Copeland, A. H.: *Admissible numbers in the theory of probability*, American Journal of Mathematics, Vol. 50, 535–552, 1928.
- [8] Efron, B.: *Bootstrap Methods: Another Look at the Jackknife*, Annals of Statistics, Vol. 7(1), 1–26, 1979.
- [9] Franklin, J. N.: *Deterministic Simulation of Random Processes*, Mathematics of Computation 17, 28–59, 1963.
- [10] Giguère, G.: *Collecting and analyzing data in multidimensional scaling experiments: A guide for psychologists using SPSS*, Tutorials on Quantitative Methods for Psychology, Vol. 2(1), 26–37, 2007.
- [11] Goh, J.; Sim, M.: *Distributionally Robust Optimization and Its tractable Approximations*, Operations Research, Vol. 58(4)., 902–917, 2010.
- [12] Griffiths, T. L.; Kalish, M. L.: *A multidimensional scaling approach to mental multiplication*, Memory and Cognition, Vol. 30(1)., 97–106, 2002.
- [13] Haruna, R.; Ishii, H.: *Simultaneous Approach of Clustering and Metric Multi-dimensional Scaling Methods with Similarity Data for Medical Diagnosis*, Asia Pacific Management Review, Vol. 11(6)., 341–345, 2006.
- [14] Herzog, T.; Owings, J. C., Jr.: *The inequivalence of two well-known notions of randomness for binary sequences*, Mathematical Reviews, Vol. (45-48)., 2134, , 1981.
- [15] Hu, X.; Cui, H.: *Generating Multi-dimensional discrete distribution random number*, Journal of Information and Computing Science, Vol. 6(1)., 23–26, 2011.
- [16] Janke, W.: *Pseudo Random Numbers: Generation and Quality Checks*, *Quantum Simulations of Complex Many-Body Systems: From Theory to Algorithms*, Lecture Notes, J. Grotendorst, D. Marx, A. Muramatsu (Eds.), John von Neumann Institute for Computing, Jülich, NIC Series, Vol. 10., 447–458, 2002.

- [17] Jaworska, N.; Chupetlovska-Anastasova, A.: *A Review of Multidimensional Scaling (MDS) and its Utility in Various Psychological Domains*, *Tutorials on Quantitative Methods for Psychology*, Vol. 5(1), 1–10, 2009.
- [18] Knuth, D.: *A programozás művészete, vol. 2.*, Műszaki Könyvkiadó, Budapest, 51–87, 1987.
- [19] Knuth, D.: *A programozás művészete, vol. 2.*, Műszaki Könyvkiadó, Budapest, 153–177, 1987.
- [20] Kolmogorov, A. N.: *Three approaches to the quantitative definition of information*. *Problemy Peredachi Informatii*, Vol. 1., 3–11, 1965.
- [21] Korobov, N. M.: *On functions with uniformly distributed fractional parts*. *Dokl. Akad. Nauk, SSSR* 62, 21–24, 1948.
- [22] Kuipers, L.; Niederreiter, H.: *Uniform Distribution of Sequences*, Wiley, New York, 1974.
- [23] Lessa, E. P.: *Multidimensional Analysis of Geographic Genetic Structure*, *Systematic Biology*, 242–252, 1990.
- [24] Loveland, D.: *A new interpretation of the von Mises' concept of random sequence*, *Z. Math. Logik Grundlagen Math.*, Vol. 12., 279–294, 1966.
- [25] Machado, J. T.; Duarte, G. M.; Duarte, F. B.: *Identify economic periods and crisis with the multidimensional scaling*. *Nonlinear Dyn* 2011-63, Springer, 611–622, 2011.
- [26] Martin-Löf, P.: *The definition of random sequences*, *Inf. Control*, Vol. 9., 602–619, 1966.
- [27] Marsaglia, G.: *Dichard Battery of Tests of Randomness*, <http://stat.fsu.edu/pub/dichard/>, 1995.(letöltés: 2012. 08. 30).
- [28] Mises, R., von: *Grundlagen der Wahrscheinlichkeitsrechnung*, *Mathematische Zeitschrift*, Vol. 5., 52–99, 1919.
- [29] Mogyoródi J.; Michaletzky Gy.: *Matematikai statisztika*, ELTE, TTK, Nemzeti Tankönyvkiadó, Budapest, 1995.
- [30] Móri T.; Székely G.: *Többváltozós statisztikai analízis*, Műszaki Tankönyvkiadó, Budapest, 1986.
- [31] Niroomand, S.; Takács Sz.; Vizvári B.: *To lay out or not to lay out?* *Annals of Operations Research*, Vol. 191, 183–192, 2011.
- [32] Niroomand, S.; Takács Sz.; Vizvári B.: *On the Generalization of the MDS Method*, *Rutcor Research Report*, 22, 2012.
- [33] Onn, S.; Weismann, I.: *Generating uniform random vectors over a simplex with implications to the volume of a certain polytope and to multivariate extremes*, *Annals of Operation Research*, Vol. 189., 331–342, 2011.
- [34] Popper, K.: *The Logic of Scientific Discovery*, London, 162–163, (1934) 1959.

- [35] Prékopa A.: *Stochastic Programming*, Kluwer, Dordrecht, 1995.
- [36] Prékopa A.: *Valószínűségelmélet műszaki alkalmazásokkal*, Műszaki Könyvkiadó, Budapest, 1962.
- [37] Ramírez Alfonsín, J. L.: *The Diophantine Frobenius Problem*, Oxford Lecture Series in Mathematics and its Applications, Vol. 30., Oxford University Press, 2005.
- [38] Rényi A.: *Valószínűségszámítás*, Tankönyvkiadó, Budapest, 1968.
- [39] Robinson, S. L.; Benett, R. J.: *A typology of deviant workplace behaviors: a multidimensional scaling study*, JSTOR, Academy of Management Journal, Vol. 38(2), 555–572, 1995.
- [40] Rounds, J. B.; Zevon, M. A.: *Cancer stereotypes: A Multidimensional Scaling Analysis*, Journal of Behavioral Medicine, Vol. 16(5), 1993.
- [41] Schied, A.: *Risk measures and robust optimization problems*, Lecture Notes, 2004.
- [42] Schnorr, C. P.: *Randomness and probability. An algorithmic foundation of probability theory*, Vol. 218 of Lecture Notes in Mathematics, Springer-Verlag, Berlin–Heidelberg–New York, 1971.
- [43] Shoben, E. J.: *Applications of multidimensional scaling in cognitive psychology*, University of Minnesota, 1983.
- [44] Sim, M.: *Robust Optimization*, Massachusetts Institute Of Technology, 2004.
- [45] Smallman-Raynor, M.; Cliff, A. D.: *Epidemiological spaces: the use of multidimensional scaling to identify cholera diffusion process in wake of the Philippines insurrection, 1899–1902*, Transactions of the Institute of British Geographers, JSTOR, 288–305, 2001.
- [46] Surányi Zs.; Babocsay Á.; Takács Sz.; Vargha A.: *Új klasszifikációs módszerek a személyiségpszichológiában*, Pszichológia, Vol. 31(4), 317–340, 2011.
- [47] Syrquin, M.: *The application of Multidimensional Scaling to a study of economic development*, The Quarterly Journal of Economics, Vol. 92(4), 1978.
- [48] Takács Sz.: *Egy nem hagyományos statisztikai eljárás bemutatása az OECD PISA adatbázison – esettanulmány*, Alkalmazott Matematikai Lapok, Vol. 27., 157–174, 2010.
- [49] Takács Sz.: *Eljárás jó minőségű véletlen egész vektorok generálására*, Alkalmazott Matematikai Lapok, Vol. 28., 41–58, 2011.
- [50] Takács Sz.: *A hátrányos helyzetűek foglalkoztatását segítő projekt tervezésének támogatása szimulációs modellel*, Munkaügyi Szemle, Stúdium, 2012. szeptember.
- [51] Vizvári B.: *Generation of Uniformly Distributed Random Vectors of Good Quality*, Rutcor Research Report, 1994.
- [52] Vizvári B.: *Generation of Uniformly Distributed Random Vectors of Good Quality*, Rutcor Research Report, 4–5, 1994.

- [53] Wald Á.: *Die Widerspruchsfreiheit des Kollektivbegriffes der Wahrscheinlichkeitsrechnung*,
Ergebnisse eines mathematischen Kolloquiums, Vol. 8., 38–72, 1937.
- [54] <http://www.afsz.hu/> - TÁMOP 1.1.4 pályázat (letöltés: 2012. 08. 30.)
- [55] <http://www.munka.hu/kozadat/> (letöltés: 2012. 08. 30.)

6. Összefoglaló

A dolgozat egy olyan munkafolyamat ismertetése, mely egy véletlen vektorokat generáló algoritmus megszületésétől egészen annak felhasználásáig tart – kisebb kitérőket is beleértve, hogy az alkalmazás felhasználói oldala is teljesebbé válhasson. A köztes lépéseket is részletesen tárgyaljuk a „véletlen” definiálásától egészen az alkalmazásban bemutatott indokolt használatig. Több helyen kisebb kitérőket teszünk alkalmazások, felhasznált rutinok irányába, melyek elméleti és gyakorlati szempontból lényegében tisztázottak és elfogadottak. A vizsgálatokat próbáltuk a lehető legegyszerűbb módokon megvalósítani, hogy a reprodukálhatóság lehetőségét mindinkább biztosítani tudjuk annak ellenére, hogy alapvetően a „véletlent” szándékoztuk vizsgálni.

A dolgozat három nagyobb blokkra tagolható. Az első blokk egy véletlen egész vektorokat generáló algoritmus köré szerveződik, annak vizsgálatáról szól. A vizsgált algoritmust Vizvári Béla kutatási riportjában [51] és Ramirez könyvében [37] is megtalálhatjuk. Az algoritmus viselkedésének vizsgálatát elvégeztük, eredményeit egyik cikkünk [49] mutatja be.

A második blokkban egy jól ismert statisztikai eljárást, a multidimenziós skálázást (MDS) vizsgáljuk. Az újdonság e blokk kapcsán az, hogy nem statisztikai, hanem optimalizálási feladatként írtuk fel, majd általánosítottuk az eljárást. Az általánosítást több módon is vizsgáltuk, eredményeink egy részét egy cikkben mutatjuk be [31], illetve további tapasztalatainkról szól egy másik kutatási riportunk [32].

A harmadik blokk a dolgozat oksági alapját adó alkalmazást ismerteti, összekapcsolva a két első fejezetet. Ennek keretében egy munkafolyamatot, egy közigazgatási probléma egy lehetséges megoldását mutatjuk be, melyben a fenti két módszer hasznos eszköznek bizonyult. Az alkalmazásban elért eredményeinket összefoglaló jelleggel [50] cikkünkben ismertetjük.

A dolgozat első két blokkjában az alábbi tagolásokat követjük.

- Az első rész a matematikai háttérrel adja az adott fejezetnek, amelyben (akár több különböző) forrás alapján általános, rövid ismertetéseket adunk a fejezet témájából – definíciókkal és eddig elért, a dolgozat szempontjából releváns eredményekkel.
- A második rész az adott fejezet tudományterületeken elért alkalmazásainak jegyzékébe ad betekintést: próbáltunk a lehetőségekhez képest széles körben keresni, felsorakoztatni tanulmányokat, könyveket vagy jegyzeteket – orvosi, biológiai, közgazdasági vagy akár humán tudományokból – annak érdekében, hogy látható legyen: a vizsgált módszerek általánosan elfogadottak, számos területen használtak.
- A harmadik rész minden fejezetben a vizsgált módszerekkel elért elméleti és/vagy gyakorlati eredményeinket, tapasztalatainkat ismerteti – részben a harmadik blokk alkalmazásához kapcsolódóan.

Az első két fejezet eredményei általánosan felhasználható eredmények, míg az eljárások, melyeket e két fejezetben ismertetünk, szervesen hozzájárulnak ahhoz, hogy az utolsóban ismertetendő alkalmazásunk eredményes legyen. A dolgozatban tehát lényegében egy munkafolyamat dokumentációját olvashatjuk: egy gyakorlati probléma kapcsán felmerült elméleti kérdésekre adott gyakorlati és elméleti válaszokat ismertetünk – mely elméleti eredményeket önállóan, a dolgozat témájától függetlenül is publikáltunk, a harmadik fejezetben pedig ezek közvetlenül egy alkalmazásban is manifesztálódnak.

7. Summary: Solving a problem of a hierarchical money-distribution system with pseudo-random integer vectors and multidimensional scaling

The thesis presents the solution process from problem statement through the development of an algorithm to the implementation of an application. The algorithm generates pseudo-random integer vectors and both research oriented and numerical results of its quality is conducted as well as an application performing simulation. The thesis describes all the important steps in this research from the definitions - and some problems of random numbers - to the final steps in the application.

Both the theoretical and the technical background are elaborated. The results are presented in detail with the aim of making them reproducible.

The thesis is divided into three major sections. The first section describes the algorithmic results concerning random integer vector generation that are published in a research report [51] and in a book [37]. The quality analysis of the algorithm and our results were published in [49].

The second section of the thesis shows results on the multidimensional scaling method (MDS). Generalizations on this method are described in several different ways, some of which have been published in [31] while rest in research report [32]. The novelty in these results is that optimization is applied - more precisely linear programming - to solve the problems on the MDS instead of the more traditional statistical approach.

The third section of the thesis presents the application and implementation details, in which the results of the first and second sections are applied. The results have been published in [50]. The practical background and the hierarchical structure of a civil service office are described, and the use of the implemented application is demonstrated on a specific project of the civil office.

All three main sections share the same structure:

- The first part of the section summarize the mathematical background.
- The second part collects technicalities and application related to the same theme of the section from articles from the field of medical, biological, humanities and economical fields.
- The third part presents the new results of the thesis.

The results of the first two sections are general results, which are then applied in the third section of the thesis.

